HAL
open science

Réseaux neuromimétiques, modularité et statistiques : estimation du mouvement pour l'asservissement visuel de robots

Patrice Wira

To cite this version:

Patrice Wira. Réseaux neuromimétiques, modularité et statistiques : estimation du mouvement pour l'asservissement visuel de robots. Traitement du signal et de l'image [eess.SP]. Université de Haute Alsace - Mulhouse, 2002. Français. NNT : . tel-00009211

HAL Id: tel-00009211

https://theses.hal.science/tel-00009211

Submitted on 9 May 2005

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

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

Année 2002 N° d'ordre : **01 MULH 0679**

UNIVERSITE DE HAUTE-ALSACE
U.F.R. DES SCIENCES ET TECHNIQUES

THÈSE

Présentée pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE DE HAUTE-ALSACE

Discipline : E.E.A.

par

Patrice WIRA

Réseaux neuromimétiques, modularité et statistiques : estimation du mouvement pour l'asservissement visuel de robots

(Arrêté Ministériel du 30 Mars 1992)

Soutenue publiquement le 7 janvier 2002 devant la commission d'examen :

Président :	M. Julien GRESSER	Professeur, Université de Haute Alsace (directeur de thèse)
Rapporteurs :	M. Gabriel ABBA	Professeur, Université de Metz
	M. Philippe MARTINET	Professeur, Université Blaise Pascal
	M. Didier WOLF	Professeur, Institut National Polytechnique de Lorraine
Examinateurs :	M. François CHAUMETTE	Directeur de Recherche, IRISA / INRIA
	M. Jean-Philippe URBAN	Professeur, Université de Haute Alsace (co-directeur de thèse)

Thèse préparée au sein du laboratoire MIPS – TROP
Ecole Doctorale « Sciences : de la molécule aux matériaux et aux systèmes »

A mes parents.

Une idée est fausse dès l'instant où l'on s'en contente.

Alain.

On se lasse de tout, excepté d'apprendre.

Virgile.

Réseaux neuromimétiques, modularité et statistiques : estimation du mouvement pour l'asservissement visuel de robots

Résumé :

Le travail de cette thèse s'inscrit dans le domaine de l'utilisation de la vision pour la réalisation de tâches robotiques et concerne plus particulièrement les aspects de poursuite de cible par asservissement visuel.

Nous nous sommes intéressé au cas d'objets mobiles, et plus particulièrement à l'estimation du mouvement de ces objets, nécessaire pour une exécution satisfaisante des tâches de poursuite. La qualité de poursuite est en effet grandement améliorée grâce à une estimation robuste du mouvement. Pour réaliser l'estimation, nous proposons une approche nouvelle, basée sur l'aspect adaptatif du filtrage de Kalman. Contrairement au filtre classique de Kalman, ce filtre n'utilise aucun modèle ni connaissance a priori. La représentation d'état est adaptée en permanence en fonction des observations courantes pour représenter au mieux la dynamique du système. Il s'agit d'une approche modulaire à modèle multiple. Plusieurs filtres sont utilisés et la probabilité pour chacun de calculer l'estimation optimale est déterminée par apprentissage. Un réseau de neurones, sur cette base statistique, supervise l'ensemble des filtres et permet de compenser le caractère non stationnaire des mouvements de l'objet mobile.

Les informations visuelles estimées servent à contrôler les déplacements du robot. Des extensions de la carte auto-organisatrice de Kohonen à sorties supervisées servent à approximer la transformation sensori-motrice du système robot-vision. La relation apprise se trouve dès lors à l'origine de la loi de commande du robot. L'apprentissage est entièrement réalisé en contexte, en exploitant les corrélations sensori-motrices durant les mouvements des robots.

Les méthodes présentées dans cette thèse ont été validées en simulation et par des expérimentations réalisées sur une plate-forme robotique réelle.

Mots-clés : robotique, contrôle adaptatif, asservissement visuel, prédiction, poursuite de cible, filtre de Kalman adaptatif, réseaux de neurones, cartes auto-organisatrices, mélange d'experts.

Neural networks, modularity and statistics : motion estimation for robotic visual servoing

Abstract:

This thesis investigates the use of vision to ensure robotic tasks, especially for mobile target tracking by visual servoing.

The aim of the thesis is to estimate the 2-D projection of the object of interest in the image sequence. Such an estimation is necessary to rewarding mobile target pursuit. Visual servoing is substantially improved with a robust and accurate motion estimation. We have developed an original method for motion estimation of simple objects which is based on an adaptive Kalman filter scheme. The proposed method does not require any prior information and does not depend on motion model and sensor model. This adaptive Kalman filter auto-adapts its state representation with the current observation to fit at best the system's dynamics. The method is extended to a multiple model approach. Different realizations of this filter, with different noise models, are used and an artificial neural network appraises the probability for each filter to compute the optimal estimation, given an input vector. This learning process supervises the filters and allows to compensate for the non stationary properties of the object's movements.

Visual servoing needs image data as input to realize robotics tasks. Thus, supervised extensions of Self Organizing Maps are used to learn the complex relationship between the sensor space and the robot's joint angle space. This neuro-controller is then used to combine the estimated image information to produces robotic control signals. The proposed method fulfills real-time constraints as well as online learning, based on the sensory-motor correlation during the robot movements.

The efficiency of this method is demonstrated through simulation results and real experiments. Visual servoing tasks are presented, which consists in tracking a mobile object with the end-effector of a three degree of freedom robot with respect to different objects.

Keywords: robotics, adaptive control, visual servoing, prediction, target tracking, adaptive Kalman filter, artificial neural networks, self-organizing maps, mixture of experts.

Remerciements

Cette thèse est l'aboutissement de trois années de recherche à la fois enrichissantes et passionnantes. Ces années resteront sans aucun doute inoubliables.

Outre Gabriel Abba, Philippe Martinet, et Didier Wolf, que je remercie d'avoir accepté d'être les rapporteurs de cette thèse, mes remerciements vont également vers François Chaumette pour sa participation à ce jury. Leurs remarques, mais aussi l'attention et l'intérêt qu'ils ont porté à mon travail m'ont permis de le finaliser au mieux. Je leur suis reconnaissant de s'être penchés sur ce travail avec rigueur et enthousiasme ; leurs remarques éclaireront mes travaux futurs.

Nul ne doit ignorer ce que je dois à Julien Gresser et à Jean-Philippe Urban. Qu'ils soient ici tous deux remerciés de la confiance qu'ils m'ont témoignée et de la liberté – sans prix – qu'ils m'ont accordée tout au long de mes investigations. Grâce à eux, cette thèse s'est déroulée dans d'excellentes conditions.

Que les autres membres du laboratoire MIPS-TROP soient également remerciés. Je pense à Jean-Luc Buessler pour nos nombreuses discussions scientifiques, mais également à Hubert Kihl et à Jean Mercklé.

Un grand merci à Gilles Hermann, à Nicolas Hrycenko, à Jean-Christophe Cexus, à Christophe Boeglin et à Stéphane Perello. Cette thèse m'a donné l'occasion d'encadrer leur stage d'ingénieur ou de DEA. Leurs contributions et leurs implications furent pour moi un échange fructueux.

Enfin, je tiens à adresser des remerciements plus décontractés, mais néanmoins chaleureux et sincères, aux autres doctorants et anciens doctorants du laboratoire. Je les veux à l'image de nos échanges et des agréables moments passés ensemble. Merci donc à Reda Kara, à William Jacquet, à Christophe Corne, et à Stéphane Breton. Leur bonne humeur fut un ingrédient de base dans la réalisation de cette thèse.

Remercier Laurent Vonau est pour moi un plaisir. Il a fait parti de ceux qui ont contribués au développement de la plate-forme robotique et qui m'ont aidés à faire fonctionner mes expérimentations.

Oublier ceux qui m'ont accompagné serait inqualifiable. Je tiens donc à associer à ce travail toutes les personnes qui m'ont aidées, de près ou de loin. Merci à Claudine d'avoir relu ce document.

Refermer ce chapitre des remerciements ne peut se faire sans accorder une pensée aux chercheurs qui m'ont inspiré ou à ceux qui m'ont servi de modèles. Connus ou reconnus, inlassablement, ils apportent des idées nouvelles ou reformulent avec une pertinence différente et novatrice des concepts déjà populaires. Entre les lignes de leurs écrits, j'admire leur enthousiasme jamais défaillant, et envie leur générosité.

Table des matières

TABLE DES MATIÈRES **1**

TABLE DES ILLUSTRATIONS **5**

LISTE DES TABLEAUX **6**

NOMENCLATURE : SYMBOLES ET NOTATIONS UTILISÉS **7**

SIGLES ET ABRÉVIATIONS USUELLES **8**

INTRODUCTION **11**

1 PRÉSENTATION 12
2 PROBLÉMATIQUE GÉNÉRALE 13
3 ORGANISATION DE LA THÈSE 14

CHAPITRE 1 ROBOTIQUE, COMMANDE PAR LA VISION **17**

1 INTRODUCTION ET PROBLÉMATIQUE 17
2 QUELQUES ASPECTS FONDAMENTAUX ET THÉORIQUES 18
2.1 Introduction *18*
2.2 Les coordonnées homogènes *19*
2.3 Position et orientation de l'effecteur d'un robot *20*
2.4 Le modèle géométrique *21*
2.5 Le modèle différentiel *22*
3 L'ASSERVISSEMENT VISUEL 23
3.1 La commande par la vision *23*
3.2 Les techniques d'asservissement visuel *23*
4 L'ASSERVISSEMENT VISUEL PAR APPRENTISSAGE 26
4.1 Une vue d'ensemble *26*
4.2 Principe *28*
5 LES CARTES AUTO-ORGANISATRICES 29
5.1 La carte auto-organisatrice de Kohonen *30*
5.2 La carte auto-organisatrice étendue *34*
5.3 La carte à sortie supervisée *35*
5.4 Considérations supplémentaires *37*
6 LE SUIVI DE CIBLE PAR ASSERVISSEMENT VISUEL 39
7 CONCLUSION 43

CHAPITRE 2 STATISTIQUES ET THÉORIES DE L'APPRENTISSAGE 45

1 INTRODUCTION 45
2 INCERTITUDES, PROBABILITÉS ET APPRENTISSAGE 46
2.1 Considérations générales 46
2.2 La nature statistique de l'apprentissage 49
2.3 L'approche Bayésienne 50
2.4 Les modèles de Markov 55
2.5 Bilan de l'apport des statistiques au connexionisme 56
3 LE FILTRAGE DE KALMAN 57
3.1 Généralités 58
3.2 Le filtre de Kalman 58
3.3 Le filtrage de Kalman adaptatif 68
3.4 Filtrage de Kalman et théorie de l'apprentissage 78
4 CONCLUSION 79

CHAPITRE 3 APPRENTISSAGE MODULAIRE 81

1 COMMENT INTRODUIRE PLUS DE CONNAISSANCES DANS LES RÉSEAUX DE NEURONES .. 81
2 DES NEURONES PLUS COMPLEXES 82
3 LA MODULARITÉ 83
3.1 Introduction et définition 83
3.2 La modularité à modèle multiple 85
3.3 La modularité vraie 87
3.4 Conclusion sur la modularité 92
4 LES MÉLANGES D'EXPERTS 93
4.1 Description 93
4.2 L'apprentissage 94
4.3 Discussion 96
4.4 Les extensions du modèle du Mélange d'Experts 96
4.5 Conclusion 100
5 CONCLUSION DU CHAPITRE 101

CHAPITRE 4 ESTIMATION PAR LE FILTRAGE DE KALMAN ADAPTATIF 103

1 INTRODUCTION 103
2 LE FILTRE DE KALMAN ADAPTATIF AU SENS DU MODÈLE D'ÉTAT 104
2.1 Adaptation du modèle d'état 104
2.2 Estimation des caractéristiques des bruits 106
2.3 Une première application simulée 107
2.4 Résultats de stabilité 109
2.5 Discussion 111
3 LE MÉLANGE DE FILTRES DE KALMAN ADAPTATIFS 111
3.1 Autres approches 111
3.2 Avec une estimation optimale des paramètres des bruits 113
3.3 Avec une estimation empirique des paramètres des bruits 114
3.4 Remarques 115
4 APPRENTISSAGE DE LA MATRICE DE TRANSITION 116
4.1 Principe 116
4.2 Simulations 117
4.3 Commentaires sur l'approche 119
5 CONCLUSION 120

CHAPITRE 5 APPLICATION ROBOTIQUE : LE SUIVI DE CIBLE 121

1 INTRODUCTION 121
2 PRÉSENTATION DE LA PLATE-FORME EXPÉRIMENTALE 121
3 SIMULATION D'UNE SCÈNE ROBOTIQUE 123
3.1 Autres environnements de simulations 123
3.2 Le logiciel Simulo 124
4 RÉALISATIONS DE CONTRÔLEURS PRÉDICTIFS 126
4.1 Application du mélange d'experts à la commande 126
4.2 Estimation du mouvement 135
4.3 Contrôle prédictif 141
5 REMARQUES DE CONCLUSION 148

CONCLUSION 149

1 CONTRIBUTIONS PRINCIPALES 149
2 PERSPECTIVES 150

ANNEXE UNE VUE GÉNÉRALE DU CONNEXIONISME 151

1 INTRODUCTION 151
2 MODÈLES CONNEXIONNISTES : PARADIGMES ET ARCHITECTURES 152
2.1 Apprentissage 152
2.2 Apprentissage et adaptation 159
2.3 Architecture 163
2.4 Terminologie 169
3 CONCLUSION 169

BIBLIOGRAPHIE 171

Table des illustrations

Figure 1-1 : Diagramme de l'asservissement visuel. 17
Figure 1-2 : Définition des repères pour un système mécanique articulé (série). 20
Figure 1-3 : Principe général de l'asservissement visuel direct d'un robot. 24
Figure 1-4 : Une carte de Kohonen bidimensionnelle. 30
Figure 1-5 : Evolution des paramètres d'une carte au cours de l'apprentissage. 31
Figure 1-6 : Evolution de la fonction de voisinage au cours de l'apprentissage. 32
Figure 1-7 : Représentation graphique d'une carte de Kohonen bidimensionnelle. 33
Figure 1-8 : Une carte auto-organisatrice étendue. 34
Figure 1-9 : Représentation d'un NeuroModule. 36
Figure 2-1 : Recouvrement de deux distributions Gausiennes mono dimensionnelles. 53
Figure 2-2 : Principe du filtre de Kalman. 60
Figure 2-3 : Transmission de fonctions de densité de probabilité Gaussiennes à travers un élément linéaire (à gauche) et un élément non linéaire (à droite). 62
Figure 2-4 : Le filtrage de Kalman décentralisé avec retour. 64
Figure 2-5 : Principe séquentiel de l'estimation de la vraisemblance par la simulation de particules (variables aléatoires). 76
Figure 3-1 : Le modèle de Miyamoto-Kawato-Setoyama-Suzuki. 83
Figure 3-2 : Exemple d'une architecture modulaire (modularité vraie). 88
Figure 3-3 : Exemple d'une architecture d'aide à l'apprentissage. 92
Figure 3-4 : Le principe du Mélange d'Experts. 93
Figure 3-5 : Un Mélange d'Experts hiérarchique. 97
Figure 4-1 : Comparatif sur une première application simulée. 108
Figure 4-2 : Etude du comportement de la matrice de transition adaptative. 110
Figure 4-3 : Principe du mélange de filtres adaptatifs. 113
Figure 4-4 : Principe du AKF avec apprentissage des matrices de transition et d'observation. 116
Figure 4-5 : Apprentissage et modélisation de la série temporelle de Hénon. 118
Figure 5-1 : La plate-forme robotique expérimentale. 122
Figure 5-2 : La scène simulée avec deux bras manipulateurs, un objet cible et un capteur stéréoscopique. 125
Figure 5-3 : Apprentissage de la Jacobienne de la relation entre deux espaces multidimensionnels.. 128
Figure 5-4 : Concept général du contrôle adaptatif inverse. 130
Figure 5-5 : Plusieurs vues d'une scène simulée. 131
Figure 5-6 : Principe de la vision active. 134
Figure 5-7 : Application de l'AKF en tant que prédicteur. 136
Figure 5-8 : Principe général de l'asservissement visuel direct d'un robot. 138
Figure 5-9 : Comportement du prédicteur SMAKF. 139
Figure 5-10 : Des exemples de trajectoires prédites. 140
Figure 5-11 : Principe de l'asservissement visuel avec prédiction. 141
Figure 5-12 : Distribution d'entrée et ordonnancement résultant des neurones de la carte de Kohonen. 142
Figure 5-13 : Evolution des régions d'intérêt en fonction de la trajectoire. 143
Figure 5-14 : Vue de la scène simulée avec une trajectoire ayant deux types de comportement. 144
Figure 5-15 : Illustration du comportement du mélange de SMAKF. 146

Figure 5-16 : Les trajectoires de la cible et celle suivie par l'effecteur avec et sans prédiction. 147
Figure 6-1 : Principe de l'apprentissage supervisé. 153
Figure 6-2 : Principe de l'apprentissage par assignation de crédit..... 154
Figure 6-3 : Principe de l'apprentissage par renforcement. 155
Figure 6-4 : Principe de l'apprentissage non supervisé. 155
Figure 6-5 : Block diagramme d'un filtre adaptatif. 159
Figure 6-6 : Compensation séquentielle de l'apprentissage..... 161
Figure 6-7 : Compensation de l'apprentissage basée sur le principe de la prédiction/comparaison.... 162
Figure 6-8 : Compensation de l'apprentissage par l'erreur. 162
Figure 6-9 : Une taxonomie des principales architectures neuronales. 163
Figure 6-10 : Continuum des modèles d'inférence. 164
Figure 6-11 : Un réseau multicouche à 2 entrées, 2 sorties, et une couche cachée. 165
Figure 6-12 : Architecture d'un réseau récurrent mono-couche à interconnexion totale..... 166
Figure 6-13 : Architecture du réseau RBF. 166

Liste des tableaux

Tableau 4.1 : Comparaison de l'erreur absolue de prédiction sur la série temporelle de Hénon..... 119
Tableau 5.1 : Erreurs d'approximation du ME avec différents réseaux superviseurs..... 127
Tableau 5.2 : Erreurs d'apprentissage des réseaux ME comparées à celles de NeuroModules..... 133
Tableau 5.3 : Comparaisons des erreurs de prédiction de différents prédicteurs. 137
Tableau 5.4 : Erreur absolue de prédiction par le mélange de SMAKF..... 144
Tableau 5.5 : Comparaison de l'erreur absolue de traînage dans l'espace articulaire (10^{-3} radians).... 147

Nomenclature : symboles et notations utilisés

Mathématiques

$\mathbf{M}$	Matrice
$\mathbf{v}$	Vecteur
s	Scalaire
$\mathbf{A}^{-1}$	Matrice inverse
$\mathbf{A}^{+}$	Matrice pseudo-inverse
$\| \, . \, \|$	Norme Euclidienne
$E[\ .\ ]$	Espérance mathématique
$\det(\ .\ )$	Déterminant d'une matrice
$\mathrm{p}(\ .\ )$	Probabilité

Réseaux de neurones

$\mathbf{x}_k$	Vecteur d'entrée à l'instant k
$\hat{\mathbf{y}}_k$	Réponse d'un réseau à l'entrée $\mathbf{x}_k$
$\mathbf{y}_k$	Réponse désirée ou valeur modèle pour l'apprentissage
$\mathbf{w}_k$	Matrice des poids synaptiques

Cartes auto-organisatrices

$\mathbf{i}$	Indice d'un neurone
$\mathbf{j}$	Indice et position du neurone vainqueur
η_k	Taux ou coefficient d'apprentissage
σ	Rayon de voisinage
$h_{\mathbf{j},\mathbf{i}(\mathbf{x}_k),k}$	Fonction de voisinage

Filtrage de Kalman

$\hat{\mathbf{x}}_k$	Vecteur d'état estimé
$\hat{\mathbf{x}}_{k+1\|k}$	Vecteur d'état prédit
$\mathbf{A}_k$	Matrice de transition
$\mathbf{C}_k$	Matrice d'observation
$\mathbf{Q}_k$	Matrice de covariance du bruit du processus
$\mathbf{R}_k$	Matrice de covariance du bruit de mesure
$\mathbf{P}_k$	Matrice de covariance de l'estimation
$\mathbf{P}_{k+1\|k}$	Matrice de covariance de l'estimation prédite

Sigles et abréviations usuelles

Adaline	ADApive LINear Element
AKF	Adaptive Kalman Filter
ANN	Artificial Neural Network
ANOVA	ANalysis Of VAriance
ART	Adaptive Resonance Theory
BP	Back Propagation
CART	Classification And Regression Trees
EM	Expectation Maximization algorithm
EKF	Extended Kalman Filter
ESOM	Extended Self-Organizing Map
GAL	Grow And Learn algorithm
GTM	Generative Topographic Map
HME	Hierarchical Mixture of Experts
HMM	Hidden Markov Model
KF	Kalman Filter
LMS	Least Mean Squares
LVQ	Learning Vector Quantization
MAP	Maximum A Posteriori
ME	Mixture of Experts
MLP	Multi-Layer Perceptron
MSE	Mean Square Error
NN	Neural Network
PCA	Principal Component Analysis
PDF	Probability Density Function
RCE	Reilly Cooper Erlbaum algorithm
RBF	Radial Basis Function
SOM	Self-Organizing Map
SSOM	Supervised Self-Organizing Map
UKF	Unscented Kalman Filter
VQ	Vector Quantization
WTA	Winner Takes All

Introduction

Si l'homme travaille autant, c'est parce qu'il est paresseux. Cette affirmation semble contradictoire, et pourtant…

L'homme n'a qu'un seul besoin réel, son confort, et c'est à cela qu'il travaille, sans répit. Cette motivation l'a poussé à construire des systèmes qui vont par la suite lui rendre de multiples services. Ces premiers systèmes ont d'abord été appelés des automates. Le nom « automate », de la racine grecque « automatos » signifie « qui se meut par soi-même » et s'applique plus complètement à tout appareil mû par un mécanisme intérieur caché.

Les automates étaient d'abord extrêmement simples. Ils n'étaient parfois même que des concepts, ou n'existaient que dans le monde de l'imaginaire. Les exemples, dans bon nombre de mythologies, sont aussi nombreux que célèbres. Dédale n'a-t-il point construit pour le roi Minos, un géant de métal dénommé Talus, qui gardait les rivages éclairés de la Crête ? Héphaïstos, le dieu du feu, n'a-t-il point inventé des trépieds mobiles qui pouvaient seuls aller et venir dans les palais de l'Olympe ?

Les avancées technologiques du siècle dernier, brutalement, ont fait entrer les automates dans le quotidien de nos civilisations. Aujourd'hui l'homme est doté d'artefacts de tous genres pour le suppléer jusque dans ses tâches les plus communes. Sa motivation reste pourtant intacte, il veut construire des systèmes de plus en plus complexes, il veut commander des machines qui le remplacent dans le travail, les souhaite de plus en plus autonomes, de plus en plus intelligentes, capables de tenir compte de plus en plus de paramètres et capables de traiter de plus en plus de données.

Une science y est consacrée, c'est l'automatique, qui donc s'adresse à la part la plus complexe de notre univers, aussi bien naturel qu'artificiel. Bâtie autour de l'information et de sa complexité dynamique, sa raison d'être est plus large que la simple transformation ou communication de cette information. Il s'agit de son interfaçage dynamique avec l'action, dans la plupart de nos artefacts ou des phénomènes naturels réactifs. C'est ce que prophétisait le titre explicatif de l'ouvrage célèbre de Norbert Wiener, "cybernetics: or control and communication in the animal and the machine" (Wiener 1961).

L'automatique est une discipline de plein droit. Si son langage est prioritairement celui des mathématiques, ce sont ses contraintes de faisabilité calculatoires et physiques qui font tout son intérêt méthodologique et technologique. L'automatique est indissociable des mathématiques, de l'informatique et du traitement du signal. Sa place se situe aux côtés de ses parents proches que sont la productique, la robotique, ou des domaines plus éloignés tels que la biologie et l'économie. Partout l'objectif reste le même, l'intégration des connaissances au service des technologies avancées, et le point commun est la notion de système.

Rendre les systèmes de plus en plus autonomes signifie les affranchir d'une « information initiale » rigide, pour les tourner vers « une information de travail », acquise en temps réel, messagère de l'environnement et source de rétroaction. La maîtrise dynamique de cette information suppose l'élaboration de modèles générateurs ou de représentations internes. Force est de reconnaître que seuls les connaissances sur les systèmes linéaires ou les machines à état fini ont été circonscrites de manière

générale, et que toutes les autres demandent à ce que d'énormes travaux de recherche soient encore consentis.

Les concepts finalisés des robots sont récents (Fukuda et al. 2001), et les plus « intelligents » d'entre eux font partie des systèmes automatiques les plus ambitieux. Ils font appel à un large spectre scientifique et technique et représentent un très vaste champ d'applications.

C'est selon cette même démarche que les robots ont été pourvus d'un processus de vision artificielle. D'aveugles, ils deviennent capables de se rendre compte de ce qu'ils font. Concrètement, cela se traduit d'une part par la prise en compte et l'interprétation d'un nouveau type d'information, des signaux bidimensionnels (des images numériques), et d'autre part par la traduction de ces informations en commandes (consignes) cohérentes. Il est alors possible d'envisager des comportements bien plus autonomes que ceux employés actuellement dans le monde industriel. D'aucuns y distinguent même un certain degré d'intelligence. La réalisation d'une telle commande ne va pas sans quelques difficultés. Ce sont ainsi des nouveaux défis qui sont lancés à l'automatique pour la perception, la représentation et l'action.

1 Présentation

L'introduction de la vision artificielle est tout à fait justifiée. Elle pose cependant un certain nombre de problèmes nouveaux, dus à l'implication d'un nouvel espace sensoriel, celui de la vision. Il faut d'abord traiter les images obtenues, extraire les informations relatives à la scène les plus pertinentes, puis il faut savoir impliquer ces résultats dans le calcul des consignes.

Les images numériques résultantes d'un processus de vision sont des signaux bidimensionnels forts d'une quantité d'informations innombrables, éparses et bruitées. Ces images doivent être traitées pour aboutir à une représentation simplifiée qui doit permettre l'extraction de quelques primitives simples. Ces primitives doivent être pertinentes et représentatives de la scène en évolution. Elles caractérisent les informations de la scène qui seront prises en compte dans la commande robotique. Leur extraction doit être effectuée dans un laps de temps aussi bref que possible pour ne pas dégrader les performance du contrôle.

Les caractéristiques visuelles constituent les informations de base d'une boucle de retour de plus haut niveau. L'asservissement se sert d'un calcul d'erreur pour corriger la position du système robotique dans son espace de commande. La boucle de retour d'informations visuelles est appelée conventionnellement *asservissement visuel* et tente de formaliser la relation entre l'espace des caractéristiques visuelles et l'espace des commandes robotiques. On appelle cette relation, la *transformation sensori-motrice*.

L'asservissement visuel repose donc sur le problème inverse d'un système complexe, multidimensionnel, fortement non linéaire et, en outre, dépendant de nombreux paramètres. Les approches modèles ont été les premières à tenter d'y répondre. Elles sont des plus contraignantes, aussi bien d'un point de vue théorique que pratique. Elles nécessitent des modèles idéaux et préétablis des éléments du système de vision et du système robotique ainsi que l'étalonnage de ces systèmes pour déterminer leurs paramètres. Les réseaux de neurones se posent alors comme une méthode alternative attrayante.

Les méthodes d'apprentissage proposent l'élaboration d'un modèle de connaissance à partir d'exemples et sans connaissance préalable. La transformation sensori-motrice peut être estimée à partir d'observations faites sur le comportement du système robotique. Un apprentissage permanent autorise d'ailleurs une mise à jour automatique de l'estimation permettant de faire face à tout changement dans la configuration ou dans la disposition de la scène robotique.

Le contrôle neuromimétique serait idéal s'il ne persistait un problème de dimension des réseaux de neurones requis. De surcroît, les réseaux de neurones ne sont naturellement pas prédisposés à intégrer les connaissances préalables disponibles. Connaissant leurs limitations, on cherche alors à

concevoir des approches modulaires, c'est-à-dire des architectures neuronales composées de plusieurs modules, plutôt que de développer un réseau unique pour résoudre la globalité d'un problème.

Dans (Slotine et Lohmiller 2001), Slotine et Lohmiller font remarquer que toute modélisation robotique est basée sur des modèles physiques et des formalismes mathématiques, alors que le concept fondamental des sciences biologiques repose sur la théorie de l'évolution. Or, l'évolution fonctionne grâce à l'accumulation et la combinaison d'états intermédiaires stables. La théorie de Darwin dit que survit celui qui s'adapte le mieux. En fait, survivre signifie pour chaque individu, de survivre à des états intermédiaires de stabilité. Le contrôle des mouvements chez les vertébrés est connu pour impliquer des combinaisons de primitives moteurs simples. Conceptuellement, ces accumulations constituent des thèmes récurrent dans l'histoire de l'intelligence artificielle (Braitenberg 1991; Wiener 1961). Slotine et Lohmiller pensent que l'évolution favorise une forme particulière de stabilité, qui garantie automatiquement la stabilité des modules entre eux.

L'objectif avoué de la modularité dans les réseaux de neurones est de préserver des structures neuronales de taille raisonnable lorsque la complexité de la tâche à apprendre se complexifie.

Lorsque des méthodes neuronales sont appliquées au problème de l'asservissement visuel, celui-ci est dit adaptatif ou par apprentissage.

2 Problématique générale

Les asservissements visuels peuvent s'intéresser, soit à des objets cibles statiques, soit à des objets cibles mobiles. Une tâche de positionnement s'occupe des cibles statiques. Pour les cibles en mouvement, il faut mettre en place une tâche de poursuite qui est plus délicate à effectuer que la tâche de positionnement. Elle nécessite d'estimer le mouvement de ces objets pour minimiser les erreurs de traînage. Le principe employé ici consiste à estimer le mouvement de l'objet et à le compenser par des lois de commande robustes.

Le travail présenté dans cette thèse s'intéresse à l'estimation du mouvement 2-D d'un objet appartenant à la scène. Nous entendons par mouvements 2-D, les mouvements observés dans l'espace sensoriel, c'est-à-dire dans l'espace des caractéristiques visuelles.

Plus précisément, l'estimation du mouvement 2-D est réalisée par un système de poursuite basé sur le principe statistique du filtrage de Kalman, capable de détecter et d'estimer les changements de direction et d'amplitude dans le mouvement de l'objet suivi. Nous proposons un estimateur à modèle multiple, utilisant plusieurs réalisations de filtres adaptatifs ne nécessitant aucun a priori, ni sur les objets, ni sur le système de vision. L'organisation de ces filtres suit le principe modulaire du Mélange d'Experts où un réseau de neurones estime la probabilité que chaque filtre calcule l'estimation idéale pour une entrée donnée. Cette forme Bayésienne de l'apprentissage est de ce fait un chemin indiqué pour la sélection de l'estimateur.

Nous avons développé une méthode permettant l'estimation du mouvement de l'objet à partir de la mesure de sa position, de sa vitesse et de son accélération dans l'image. Le mouvement estimé de l'objet dans l'image, permet de compenser dans la loi de commande son éventuel caractère non stationnaire. Il en résulte également une commande plus précise et plus robuste en présence d'incertitudes et d'erreurs de modélisation.

L'introduction de la vision dans le contrôle robotique ne peut s'accompagner que d'un traitement intelligent de l'information visuelle afin de concrétiser complètement l'autonomie du robot. L'intérêt de l'application des techniques de prédiction au domaine de la poursuite de cible par asservissement visuel est encore une fois démontré.

Nous proposons un schéma d'asservissement visuel adaptatif et prédictif, avec absence de toute intervention externe et basée sur une connaissance analytique minimale du système robot-vision. Ce schéma de commande, grâce à un contrôleur neuronal, est doté d'une capacité de réaction dynamique robuste et précise vis-à-vis de perturbations telles des modifications géométriques du

système. Il est également capable, grâce à un processus de prédiction, de compenser l'évolution de la scène et de l'environnement, comme celle qu'une cible qui bouge.

3 Organisation de la Thèse

Ce document est organisé de la façon suivante.

Le **Chapitre 1** est dédié à la description et l'analyse de quelques formalismes de modélisation des robots en tant que systèmes mécaniques articulés. Ces formalismes autorisent une meilleure compréhension des problèmes impliqués par l'introduction de la vision dans la commande des robots. Afin de bien poser la problématique, nous énumérons différentes techniques d'asservissement visuel et y détaillerons notamment l'asservissements 2-D sur lequel est basé notre approche. Cet asservissement permet de formaliser le problème de la commande par une régulation dans l'image. Nous détaillons par la même occasion les cartes auto-organisatrices de Kohonen ainsi que plusieurs de leurs variantes. Nous nous servons de ce type de réseaux de neurones pour la réalisation d'une approche de l'asservissement visuel par apprentissage. Les extensions des cartes auto-organisatrices permettent d'approximer la transformation sensori-motrice du système et permettent de ce fait de synthétiser un contrôleur neuronal. Ce chapitre propose également un survol des méthode de poursuite de cible par asservissement visuel.

Le **Chapitre 2** propose un point de vue statisticien des réseaux de neurones artificiels. Il décrit les interactions entre la théorie des statistiques et les méthodes par apprentissage, notamment comment les statistiques peuvent contribuer à l'obtention d'un apprentissage plus performant. Ce chapitre se concentre sur l'estimateur Bayésien qu'est le filtre de Kalman. Un grand nombre de techniques, toutes basées sur ce filtre, sont passées en revue. Il y est spécialement abordé le problème du filtrage de Kalman adaptatif.

Le **Chapitre 3** traite la question générale de l'introduction de connaissances dans les réseaux de neurones qui trouve des éléments de réponse dans les architectures modulaires. Nous développons une vue d'ensemble des architectures neuronales modulaires. Deux formes de modularité sont distinguées, la modularité à modèle multiple et la modularité vraie. Selon ces deux formes, les modules apprennent soit l'ensemble d'un problème, soit des sous-problèmes. Une forme particulière de modularité est présentée, l'approche des Mélanges d'Experts (ME).

Nous développons dans le **Chapitre 4** un estimateur pour prédire les mouvements d'un objet cible pour une tâche de poursuite par asservissement visuel. L'estimateur est basé sur un filtre de Kalman adaptatif au sens du modèle d'état (SMAKF). Ce filtre ne nécessite aucune connaissance a priori pour estimer l'état d'un système. Plusieurs filtres sont organisés au sein d'un schéma d'apprentissage modulaire (l'approche des ME) qui permet de compenser le caractère non stationnaire du système à estimer.

Subséquemment, le **Chapitre 5** inclut l'évaluation et la validation expérimentale de l'apport de l'estimation du mouvement dans des tâches de poursuite de cible par asservissement visuel ou par vision active. Les performances du mélange de filtres adaptatifs, pour estimer les mouvements d'un objet cible au comportement non connu, sont évaluées grâce au développement d'un logiciel de simulation et grâce à une plate-forme expérimentale. L'estimation du mouvement 2-D de l'objet est associée à une commande neuronale basée sur une extension des cartes auto-organisatrices de Kohonen. Nous décrivons également l'état de l'art de l'asservissement visuel par apprentissage.

Un regard critique est finalement posé sur l'ensemble de ces travaux en **Conclusion**. On évaluera la robustesse de l'estimateur modulaire proposé pour compenser les mouvements d'un objet mobile. On discutera de la validation du schéma de commande vis-à-vis du suivi des modifications du mouvement des objets suivis. Nous terminerons cet exposé par quelques perspectives envisageables dans un futur proche ou plus lointain.

Une **Annexe** contient une introduction aux réseaux de neurones artificiels. Nous présentons un survol des principales architectures neuronales et des paradigmes d'apprentissage pour mieux explorer la notion de réseaux neuromimétiques modulaires.

Chapitre 1

Robotique, commande par la vision

1 Introduction et problématique

Associer un capteur extéroceptif à un robot est essentiel. On peut y trouver plusieurs raisons, comme l'amélioration de performances et l'amélioration de ces possibilités dans la réalisation de différentes tâches. Le capteur de vision est un élément sensoriel particulièrement intéressant. Distant, il est capable de fournir des informations d'une très grande richesse sur l'environnement dans lequel un robot évolue.

Commander des systèmes robotiques à l'aide de capteurs de vision est appelé asservissement visuel. L'asservissement visuel est un domaine de recherche dont le but est d'intégrer en temps réel les informations visuelles dans la boucle d'asservissement des robots. La tâche de base, en asservissement visuel d'un bras robotique, est le contrôle de façon stable et robuste de l'attitude 6-D (position et orientation) de son effecteur. Ce principe est illustré de façon très simple sur la Figure 1-1.

D'autres tâches peuvent être envisagées, comme le contrôle uniquement en position de l'effecteur d'un bras, ou le contrôle seulement en orientation de caméras placées sur une tête robotique. Cette dernière tâche est appelée vision active. Elle permet de contrôler les paramètres des caméras, et elle sert à améliorer certaines tâches de perception comme la reconnaissance, la localisation ou la reconstruction 3-D d'objets.

L'asservissement visuel est une activité de recherche jeune. Depuis une quinzaine d'années, elle rassemble des savoir-faire tirés entre autres de la commande de robot, de la cinématique, de la dynamique, du traitement du signal et des images, du calcul temps réel, et de l'architecture des systèmes.

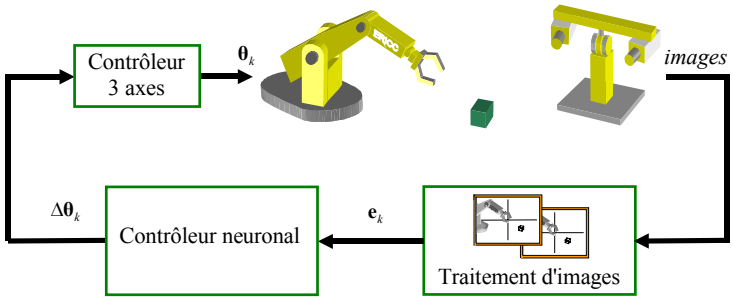


Figure 1-1 : Diagramme de l'asservissement visuel.

Les différents travaux en asservissement visuel s'intéressent soit à des objets statiques, soit à des objets mobiles. Les premiers demandent une tâche de positionnement. Les objets en mouvement réclament une tâche de poursuite. Cette dernière suppose d'estimer le mouvement des objets pour assurer un suivi correct, sans erreur de traînage.

Notre objectif vise à estimer le comportement d'une cible pour en tenir compte explicitement en l'intégrant dans la problématique du contrôle robotique dynamique par la vision. Cette tâche de poursuite de cible par asservissement visuel se veut sans connaissance a priori, elle n'inclut aucune information préalable sur la scène, sur le système robotique et sur l'objet à suivre.

Pour situer le contexte de nos travaux de recherche, nous allons introduire certaines notions fondamentales concernant à la fois la modélisation des systèmes robotiques et leur commande. Nous verrons également comment les informations visuelles sont prises en compte dans la réalisation d'une boucle de commande en récapitulant les différentes techniques proposées jusqu'à ce jour pour la réalisation de tâches d'asservissement visuel.

Nous introduisons l'asservissement visuel adaptatif ou asservissement visuel par apprentissage. Ces techniques d'asservissement visuelle utilisent une matrice Jacobienne adaptée ou estimée en ligne.

Ce chapitre introduit les cartes auto-organisatrices de Kohonen, ainsi qu'un certain nombre de leurs extensions, notamment une carte à sorties supervisées (SSOM) et une carte étendue (ESOM). Les extensions de la carte de Kohonen permettent d'appliquer le principe d'auto-organisation au problème de la commande robotique (Ritter et al. 1992) et de réaliser des schéma d'asservissement visuel adaptatif.

Enfin, ce chapitre se termine par une courte synthèse sur les travaux effectué en poursuite de cible par asservissement visuel.

2 Quelques aspects fondamentaux et théoriques

Un déplacement est une transformation spatiale qui résulte d'une rotation et d'une translation. Tout mouvement peut être décomposé en ces deux transformations élémentaires. Le formalisme analytique de la transformation sensori-motrice apporte une connaissance approfondie de la nature et de la complexité des problèmes de la commande, de la précision, de l'apprentissage, etc.

Le rappel de quelques aspects fondamentaux nous conduit, plus loin dans ce travail, à la modélisation des robots. Il permet surtout de bien comprendre dans quelle mesure il est possible de concevoir des contrôleurs pour la commande des robots, en particulier lorsque l'information de retour est fournie par un capteur de vision.

2.1 Introduction

Un robot est un système mécanique articulé avec plusieurs degrés de liberté. On devine aisément que faire suivre une trajectoire donnée à son organe terminal présuppose la maîtrise de la coordination de tous les degrés de liberté, en position, vitesse et accélération. L'approche classique, pour définir l'état du système robotique, ne prend en compte que la position et la vitesse. Ceci ne va pas sans faire une supposition, celle de se dire que la position et la vitesse sont les deux seuls paramètres pertinents ou caractéristiques de la fonctionnalité du système.

Avant d'aller plus loin, reprenons les sept éléments que distingue Coiffet pour définir un système robotique (Coiffet 1992) :

- la source d'énergie,
- les moteurs, également appelés actuateurs ou actionneurs,
- les transmissions des actuateurs aux articulations du robot,
- le système mécanique articulé qui n'est autre que le corps du robot,

- les capteurs internes ou proprioceptifs qui permettent de connaître l'état interne du robot (en simplifiant l'état peut être défini par la valeur et la vitesse de chaque articulation),
- les capteurs externes ou extéroceptifs qui fournissent des informations sur l'environnement dans lequel le robot évolue et interagit,
- le système de commande ou le contrôleur.

La modélisation du robot consiste à établir une représentation du système mécanique articulé. Un système mécanique articulé est un ensemble de solides reliés entre eux par des liaisons schématisées par la Figure 1-2. On définit le degré de liberté d'une liaison comme étant le nombre de mouvements indépendants possibles d'un solide par rapport à l'autre. Cette étape de modélisation permet de se forger une connaissance a priori d'un système robotique, et cette connaissance supplémentaire peut être utile pour l'élaboration d'un contrôleur.

Un robot doit être capable de se mouvoir, ou tout au moins de déplacer une partie de son corps seulement. Tout déplacement est une transformation spatiale qui résulte d'une rotation et d'une translation. Ce déplacement peut être vu comme une transformation entre deux repères.

Il existe plusieurs manières de décrire la position et l'orientation d'un corps rigide par rapport à un repère de référence. La méthode classique pour décrire la position d'un corps emploie le formalisme des matrices 3×3, les coordonnées homogènes celles des matrices 4×4. L'expression de l'orientation d'un corps rigide peut se faire par les cosinus directeurs, par les angles d'Euler, ou encore par les angles de Bryant.

Nous avons choisi le formalisme des matrices homogènes pour définir les transformations, et celui des vecteurs position (à trois composantes) associés aux angles d'Euler pour définir la position et l'orientation d'un solide dans l'espace tridimensionnel.

Nous allons d'abord voir comment exprimer un repère par rapport à un autre repère, puis comment exprimer les relations entre les coordonnées d'un point relativement à deux repères distincts. Ceci nous amènera à modéliser le système articulé qui constitue le corps d'un robot.

2.2 Les coordonnées homogènes

Soit R défini par $(\mathrm{O},\mathbf{i},\mathbf{j},\mathbf{k})$, un système de coordonnées de références orthonormées associé à l'espace tridimensionnel dans lequel un robot est susceptible d'évoluer. Si en plus, on considère un repère mobile R_m référencé dans R, défini par son origine O_m et par les trois vecteurs qui constituent sa base $\mathbf{i}_m$, $\mathbf{j}_m$ et $\mathbf{k}_m$ par rapport à R, alors on peut écrire :

$$\mathrm{OO}_m = p_x\mathbf{i} + p_y\mathbf{j} + p_z\mathbf{k}\,, \tag{1.1}$$

$$\mathbf{i}_m = n_x\mathbf{i} + n_y\mathbf{j} + n_z\mathbf{k}\,, \tag{1.2}$$

$$\mathbf{j}_m = o_x\mathbf{i} + o_y\mathbf{j} + o_z\mathbf{k}\,, \tag{1.3}$$

$$\mathbf{k}_m = a_x\mathbf{i} + a_y\mathbf{j} + a_z\mathbf{k}\,, \tag{1.4}$$

On peut référencer le repère R_m dans R en écrivant simplement que :

$$\mathrm{R}_m = \begin{bmatrix} \mathbf{i}_m & \mathbf{j}_m & \mathbf{k}_m & \mathrm{OO}_m \\ 0 & 0 & 0 & 1 \end{bmatrix}_\mathrm{R} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}_\mathrm{R}, \tag{1.5}$$

- où $\mathbf{A}$ est l'opérateur matriciel représentant la rotation (on l'appelle matrice de rotation généralisée ;
- et où $\mathbf{t}$ est l'opérateur représentant la transformation de translation (c'est un vecteur à trois éléments).

Nous allons montrer l'avantage du formalisme des transformations homogènes en étudiant les relations entres les coordonnées d'un même point exprimées dans deux repères distincts.

Pour cela, considérons deux repères, R_i et R_j d'origines respectivement O_i et O_j. Un point P peut être référencé dans les deux repères avec $\mathbf{x}^i$ le vecteur coordonnées dans R_i et avec $\mathbf{x}^j$ le vecteur coordonnées dans R_j. La relation entre les vecteurs de coordonnées $\mathbf{x}^i$ et $\mathbf{x}^j$ s'écrit :

$$\mathbf{x}^i = \mathbf{A}^{i,j}\mathbf{x}^j + \mathbf{t}^{i,j}, \tag{1.6}$$

avec $\mathbf{t}^{i,j}$ les coordonnées de O_j dans le repère R_i et avec $\mathbf{A}^{i,j}$ la matrice de rotation du repère R_j par rapport à R_i.

Avec $\mathbf{T}^{i,j}$ la matrice de transformation homogène,

$$\mathbf{T}^{i,j} = \begin{bmatrix} \mathbf{A}^{i,j} & \mathbf{t}^{i,j} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{1.7}$$

la relation (1.6) précédente s'écrit :

$$X^i = \mathbf{T}^{i,j} X^j, \tag{1.8}$$

où $X^i = \begin{bmatrix} \mathbf{x}^i & 1 \end{bmatrix}^T$ et $X^j = \begin{bmatrix} \mathbf{x}^j & 1 \end{bmatrix}^T$ sont les coordonnées homogènes du point P dans R_i et dans R_j respectivement.

En conséquence, on voit que le déterminant de $\mathbf{T}^{ij}$ est égal à 1, et que l'inverse $\mathbf{T}^{ji}$ de la matrice $\mathbf{T}^{ij}$ est définie par la relation suivante :

$$X^j = \mathbf{T}^{j,i} X^i. \tag{1.9}$$

Notons également que la matrice $\mathbf{T}^{ij}$ contient à la fois la rotation et la translation entre les deux repères.

2.3 Position et orientation de l'effecteur d'un robot

Chaque articulation du corps d'un robot est systématiquement associé à un changement de repère. Modéliser un robot revient donc à chercher à exprimer toutes les transformations de repère, pour passer du repère de l'effecteur à celui de la base du robot, voire même le référentiel de la scène.

C'est donc un problème d'évaluation de la position et de l'orientation d'un corps solide indéformable, en l'occurrence l'effecteur du robot, par rapport à un repère de référence. L'effecteur lui-même est défini en général par un repère connu qui lui est attaché, R_E. En fait, calculer les transformations successives dues à chaque degré de liberté revient à déterminer le modèle géométrique direct du robot.

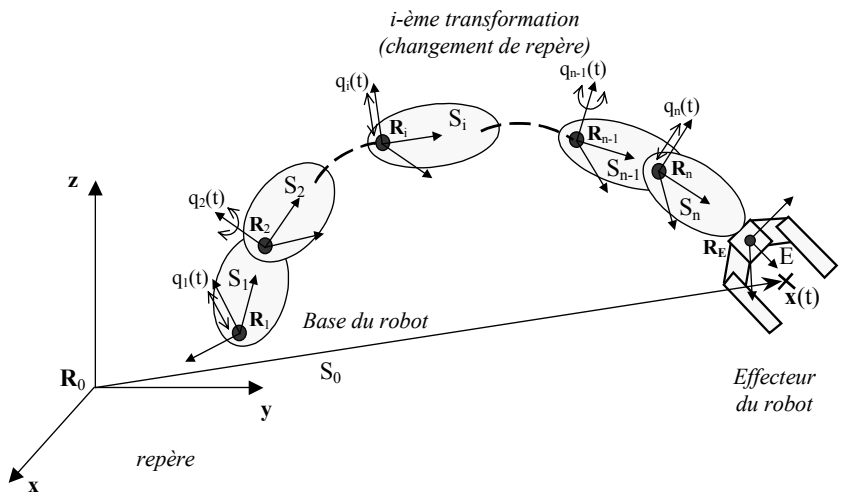


Figure 1-2 : Définition des repères pour un système mécanique articulé (série).

La modélisation du corps du robot se fait en exprimant les coordonnées d'un point dans R_n, puis dans R_{n-1}, jusque dans R_0. On peut se reporter à la Figure 1-2 pour avoir un aperçu des repères associés aux articulations.

La transformation de R_i à R_{i-1}, $\mathbf{T}^{i-1,i}$ dépend des variables articulaires θ_i du robot :

$$X^{i-1} = \mathbf{T}^{i-1,i} X^i = \mathbf{T}^{i-1,i}(\theta_i) X^i . \tag{1.10}$$

La relation récurrente (1.10) permet de déduire la position du point final du robot :

$$X^0 = \left(\prod_{i=1}^{n} \mathbf{T}^{i-1,i}(\theta_i) \right) X^n . \tag{1.11}$$

L'orientation de l'effecteur se déduit en effectuant le changement de base entre R_0 et R_E :

$$\mathbf{A}^{0E} = \left(\prod_{i=1}^{n} \mathbf{A}^{i-1,i}(\theta_i) \right) \mathbf{A}^{nE} . \tag{1.12}$$

De la matrice $\mathbf{A}^{0E}$, de dimensions 3×3, on peut déduire les angles d'Euler(Lallemand et Zeghloul 1994; Paul 1981, page 35).

Pour déterminer les matrices $\mathbf{T}^{i-1,i}$, c'est-à-dire les exprimer en fonction de θ_i, il existe des conventions, telles celles de Denavit-Hartenberg, ou de Khalil-Kleinfinger (aussi appelée convention de Denavit-Hartenberg modifiée), que nous n'allons pas détailler. Nous invitons le lecteur à se référer à un ouvrage incontournable (Paul 1981), ou à un ouvrage plus récent (Lallemand et Zeghloul 1994).

Ces conventions sont des méthodologies qui permettent de retrouver avec aisance les matrices de passage. Elles sont fonction du type d'articulation ; les plus usuelles étant le pivot et la liaison prismatique. Elles sont toutes listées dans (Lallemand et Zeghloul 1994, page 6).

2.4 Le modèle géométrique

2.4.1 Le modèle géométrique direct et inverse

Parmi tous les modèles de connaissance, celui qui satisfait le mieux l'inversion mathématique, l'écart modèle-réalité minimum, ainsi qu'un temps de calcul compatible avec la vitesse désirée du robot, est sans aucun doute le modèle géométrique. Ce modèle permet de lier l'espace des variables articulaires à l'espace des tâches.

En supposant que les segments d'un robot sont parfaitement rigides ou indéformables et ses articulations mécaniquement parfaites, c'est-à-dire sans élasticité et sans frottement, le modèle géométrique direct correspond à

$$\mathbf{x} = \mathbf{f}(\boldsymbol{\theta}) . \tag{1.13}$$

Dans la relation (1.13), $\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^T$ est un vecteur dont les *n* composantes sont les grandeurs à contrôler pour obtenir une exécution correcte de la tâche (variables opérationnelles). Ce vecteur est défini par rapport à un repère tridimensionnel orthonormé R_0 qui est le repère de la tâche, ou repère de référence. Le nombre *n* de composantes de $\mathbf{x}$ dépend de la tâche fixée. On utilise trois composantes pour contrôler la position de l'effecteur uniquement, les trois dimensions de l'environnement dans lequel le robot évolue, et six composantes pour contrôler à la fois la position et l'orientation de l'effecteur (3 positions et 3 variables angulaires).

Le vecteur $\boldsymbol{\theta}$ est un vecteur dont les *m* composantes sont les variables articulaires ou variables généralisées : $\boldsymbol{\theta} = [\theta_1 \quad \theta_2 \quad \cdots \quad \theta_m]^T$ où *m* est le nombre de degrés de liberté du robot.

Le problème du modèle géométrique direct revient à chercher la position et l'orientation de l'organe terminal, appelé effecteur, dans le repère R_0 en fonction des valeurs des variables articulaires. Notons que :

- on aura toujours $n \leq m$, ce qui traduit le fait qu'avec *m* degrés de liberté, on ne peut contrôler au plus que m grandeurs indépendantes ;

- on parlera de redondance des degrés de liberté, lorsque $n < m$, situation qui peut poser des problèmes au niveau de la commande par un modèle inversé.

Si la fonction $\mathbf{f}(.)$ exprime les relations entre $\mathbf{x}$ et $\boldsymbol{\theta}$, le modèle géométrique inverse est défini par la relation suivante :

$$\boldsymbol{\theta} = \mathbf{f}^{-1}(\mathbf{x}) . \tag{1.14}$$

2.4.2 Le modèle géométrique pour la commande

Si le modèle géométrique est utilisé pour la commande (son inverse plus exactement), alors on parle de commande géométrique ou de commande en position. Contrôler un robot par le modèle géométrique inverse revient à chercher le vecteur $\boldsymbol{\theta}$ des variables articulaires qui permet à l'effecteur du robot d'atteindre une position désirée $\mathbf{x}$ dans l'espace des tâches.

On parle de résolvabilité d'un robot lorsqu'on touche aux problèmes du modèle géométrique inverse (MGI). Utiliser le MGI pour trouver les variables articulaires abouti soit à un nombre infini de solutions, à un nombre limité de solutions, ou à une unique solution.

L'utilisation du modèle géométrique inverse est une méthode très peu performante à cause des non linéarité de la relation (1.13). Il faut généralement rajouter des contraintes géométriques supplémentaires, parfois très peu évidentes à mettre en œuvre et qui supposent une bonne connaissance de la tâche à effectuer pour distinguer une solution parmi les solutions possibles trouvées. Mais cette méthode est surtout très peu précise. Accroître les performances de la commande se fait en ayant recourt à des techniques itératives qui utilisent les relations dérivées entre $\mathbf{X}$ et $\boldsymbol{\theta}$, donc qui utilisent le modèle différentiel que nous allons sommairement introduire dans la section suivante.

2.5 Le modèle différentiel

Nous avons vu que le modèle géométrique pose deux problèmes quant à son utilisation dans la commande :

- la non linéarité de l'équation, qui entraîne des difficultés d'obtention de transformateur de coordonnées inverse,
- et la vitesse de déplacement du robot qui n'est pas contrôlée, alors qu'il est important de maîtriser la durée d'exécution d'une tâche.

L'utilisation du modèle cinématique ou différentiel permet dans une certaine mesure de contourner les deux inconvénients qui viennent d'être cités.

Précisons que le modèle cinématique découle de la description complète d'une modélisation par les formules de cinématique du solide (le mouvement est décomposé et caractérisé selon une vitesse de translation et une vitesse de rotation), alors que le modèle différencié se déduit du modèle géométrique direct par l'opérateur de dérivation matriciel (c'est donc une approximation, puisque seul le terme d'ordre 1 du développement de Taylor du modèle géométrique est conservé) :

$$\Delta\mathbf{X} = \mathbf{J}\Delta\boldsymbol{\theta} . \tag{1.15}$$

La matrice $\mathbf{J}$ est la matrice Jacobienne du modèle géométrique dérivé. La commande revient à rechercher les variations articulaires $\Delta\boldsymbol{\theta}$ pour des variations $\Delta\mathbf{x}$ imposées dans l'espace des tâches. Le modèle différentiel inverse est dans ce cas défini par la relation :

$$\Delta\boldsymbol{\theta} = \mathbf{J}^{-1}\Delta\mathbf{x} . \tag{1.16}$$

Déterminer le modèle différentiel inverse $\mathbf{J}^{-1}$ revient à un problème d'inversion de matrice. Lorsque la matrice $\mathbf{J}$ est carrée, c'est-à-dire lorsque le nombre de degré de liberté $n = 6$, le problème est de complexité limitée. Dans le cas contraire, l'algorithme proposé par Gréville est une méthode très robuste mais très peu compatibles avec des applications concrètes. En fait, toutes les méthodes d'inversion de matrice sont des contrôleurs potentiels de robot.

3 L'asservissement visuel

3.1 La commande par la vision

L'utilisation de capteurs proprioceptifs, qui mesurent les grandeurs internes du robot, n'autorisent que des applications d'un genre limité. L'objectif visé, lorsque la vision est introduite dans une boucle de régulation, est de rendre les robots plus autonomes, ou plus indépendants par rapport à la scène dans laquelle ils évoluent. En fait, il est facile de concevoir qu'un robot capable de tenir compte de son environnement voit son champ applicatif considérablement augmenter.

Si l'introduction de la vision – une perception à distance – paraît séduisante à première vue, elle n'en reste pas moins difficile à mettre en œuvre. En effet, introduire la vision dans une boucle de régulation amène des problèmes nouveaux. Le premier est lié à l'extraction des informations des données brutes que fournit le capteur, le second concerne la commande elle-même, puisqu'il s'agit d'être capable de tenir compte de ces informations pour les traduire en consigne pour le système robotique.

Un capteur de vision est complexe dans le sens où les images numériques qu'il fournit contiennent une quantité infinie d'informations. L'extraction d'informations visuelles pertinentes des images est un problème complexe. Les bases ont été jetées par Marr (Marr 1982) ; le traitement et l'interprétation des d'images restent encore un domaine de recherche. L'extraction des informations peut être favorisée en marquant les objets d'intérêt dans la scène. En plaçant des amers visuels sur les objets importants d'une scène, et en simplifiant à l'extrême le traitement des images par un seuillage binaire, l'information visuelle peut être extraite en temps réel, c'est-à-dire à la cadence vidéo.

La science qui s'intéresse à la commande des systèmes robotiques à l'aide de capteurs de vision est appelée asservissement visuel. Elle est intimement liée à l'informatique. Si l'on sait plus ou moins extraire les informations des images, encore faut-il les utiliser à bon escient dans une boucle de retour de haut niveau pour faire interagir le robot avec son environnement. Les boucles d'asservissement visuel forment deux familles définies selon l'espace dans lequel est déterminée l'erreur de positionnement de l'effecteur du robot par rapport à une position à atteindre, une cible. On distingue ainsi l'asservissement 3-D et l'asservissement 2-D.

3.2 Les techniques d'asservissement visuel

3.2.1 Asservissement 3-D et asservissement 2-D

Une première étude nous a préalablement permis de nous familiariser avec l'asservissement visuel en général et avec la vision active en particulier (Wira 1997). Les techniques d'asservissement visuel peuvent être séparées en deux catégories, selon l'espace dans lequel l'erreur de positionnement est définie : la première catégorie d'asservissement visuel utilise une erreur calculée dans un espace tridimensionnel estimé par une technique de reconstruction 3-D ; et la seconde catégorie se sert d'une erreur définie dans directement les images. Les techniques d'asservissment 2-D et 3-D ont fait l'objet travaux depuis quelques années (Espiau et al. 1992; Hashimoto 1993; Hutchinson et al. 1996; Weiss et al. 1987).

a Asservissement 3-D

Lorsque l'asservissement visuel utilise une erreur définie dans un espace tridimensionnel, on parle de "position-based visual servoing" ou d'asservissement 3-D (Sanderson et Weiss 1980; Wilson et al. 1996). Cette approche est composée de deux étapes successives, une étape de reconstruction 3-D et une étape de transformation de coordonnées de l'espace tridimensionnel vers l'espace articulaire du robot. La reconstruction consiste en l'interprétation des informations visuelles et en l'estimation de la

position de l'effecteur et de la position de la cible dans un repère Euclidien associé à l'espace tridimensionnel de la scène robotique. Plus précisément, si l'on considère une tâche de positionnement rigide contrôlant six degrés de liberté (3 pour la position, 3 pour l'orientation), la consigne est exprimée sous la forme d'un changement de repère décrivant la position et l'orientation d'un repère lié à l'objet d'intérêt par rapport au repère lié à la caméra. En théorie, une seule itération est suffisante et une commande du robot en position permet d'atteindre la position désirée. En pratique évidemment, les diverses erreurs de modèles et d'étalonnage impliquent un rebouclage et un asservissement en vitesse pour assurer la stabilité du système (Martinet et al. 1997; Wilson et al. 1996). La robustesse de cette approche dépend essentiellement de la précision de la reconstruction. La méthode de reconstruction la plus populaire est la contrainte épipolaire (Faugeras 1993; Faugeras et al. 1987).

b Asservissement 2-D

Les asservissements visuels qui utilisent en entrée de la boucle de commande des informations exprimées dans l'espace image, autrement dit une erreur définie dans l'espace des caractéristiques visuelles, sont appelés "image-based visual servoing" ou asservissement 2-D. Dans ce cas, l'erreur de positionnement mesurée dans les images entre l'effecteur du robot et la cible à atteindre est convertie directement en corrections angulaires. Cette approche est illustrée par la Figure 1-3.

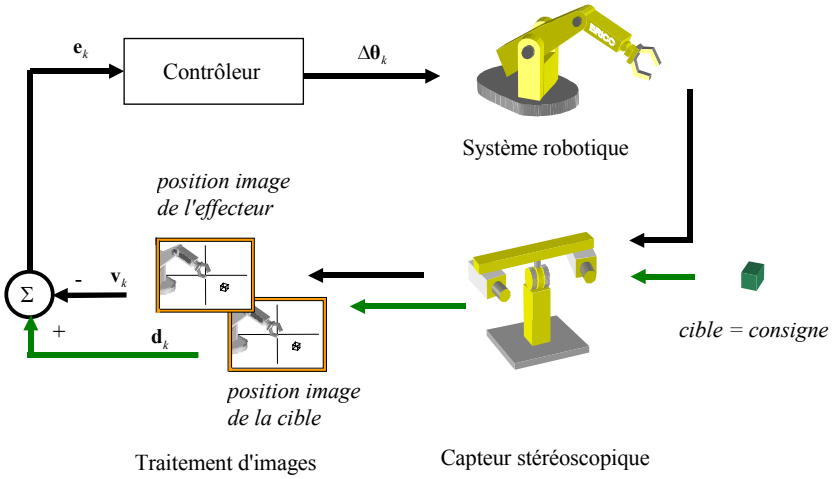


Figure 1-3 : Principe général de l'asservissement visuel direct d'un robot.
L'asservissement direct, ou asservissement 2-D, convertit directement une erreur évaluée dans l'espace sensoriel en une consigne dans l'espace des commandes, sans passer par une étape de reconstruction.

c Asservissements hybrides

Des asservissements visuels nouveaux ont été proposés dans des travaux très récents. Ils sont appelés asservissement 2-D½ (Malis et al. 1998), asservissement 2-½-D (Malis et al. 1999) ou asservissement d2-D/dt (Chaumette 1998) et sont généralement des approches où la caméra est fixée sur l'organe terminal du robot.

Concernant l'asservissement 2-D½ , les informations utilisées comme mesures et consignes sont, pour certaines d'entre elles, exprimées dans l'image, et pour les autres dans le repère de la

caméra. Cette approche est basée sur l'estimation de l'homographie qui relie l'image de plusieurs points. A partir de l'homographie estimée, il est possible de calculer le déplacement en rotation que la caméra doit effectuer pour atteindre une position désirée, ainsi que son déplacement en translation à un facteur d'échelle près. Un asservissement purement 3-D est impossible, par contre, la combinaison des informations 2-D et 3-D permet d'aboutir à une solution et l'approche peut être qualifiée d'hybride. Ainsi, l'erreur de rotation estimée est contrôlée directement avec les 3 degrés de liberté en rotation du robot et le mouvement le long de l'axe optique est contraint de manière à faire tendre le rapport des profondeurs entre la caméra et l'objet vers une valeur unitaire constante. Les deux degrés de liberté de translation parallèles au plan image servent à centrer la cible au milieu de l'image grâce aux coordonnées d'un point de l'objet. Deux boucles d'asservissement séparées se dessinent, une en rotation et l'autre en translation.

Les asservissements décrits jusque là reposaient sur l'utilisation de caractéristiques visuelles géométriques, les coordonnées d'un point, les paramètres d'une droite, etc. Les contraintes imposées sont que ces primitives géométriques doivent exister dans la scène et surtout qu'il soit possible de les extraire de l'image à une cadence soutenue afin de conserver la robustesse et la stabilité de la boucle de commande. Dans la pratique, l'introduction de marqueurs ou d'amers visuels disposés dans la scène et sur les objets permet d'extraire plus facilement les positions des projections perspectives dans l'image. L'asservissement d2-D/dt contourne ces contraintes fortes en employant des informations visuelles dynamiques. Les mouvements de la caméra sont alors contrôlés pour que le mouvement 2-D mesuré atteigne un champ de vitesse désiré. Les algorithmes d'analyse du mouvement 2-D sont génériques quant au contenu de la séquence d'images, et cet avantage leur confère d'appréhender simplement des tâches extrêmement difficiles à modéliser en utilisant les informations visuelles géométriques. L'approche a été validée par une tâche d'alignement du capteur dans la direction de déplacement de celui-ci, et par une tâche de positionnement telle que le plan image de la caméra soit parallèle à un plan quelconque. La suppression de connaissances géométriques fortes sur la scène grâce à des traitements de séquences d'images fondés sur l'analyse du mouvement devrait élargir le champ d'application de la commande référencée vision.

3.2.2 Asservissement visuel statique et dynamique

La boucle de retour des informations visuelles peut être d'une extrême simplicité : si l'erreur est évaluée lorsque la scène n'évolue plus, c'est-à-dire quand le robot est à l'arrêt, alors on se trouve dans le cas statique, "look and move" (Weiss 1984). En raison de sa très grande stabilité, cette approche trouve de nombreuses applications industrielles (Hutchinson et al. 1996).

La boucle de retour visuelle peut également être basée sur une constante évaluation de l'erreur de positionnement et sur une correction en temps réel de la position du robot. L'approche est logiquement qualifiée de "dynamic look and move" (Weiss et al. 1987). C'est cette boucle dynamique qui est généralement utilisée dans les tâches d'asservissement visuel. Les différentes approches sont résumées dans (Hashimoto 1993), dans (IEEE 1996), ou plus récemment dans (Martinet 1999). On distingue généralement celles qui utilisent des systèmes de vision embarqués sur le bras, de celles qui utilisent des caméras séparées et fixes par rapport au bras manipulateur. Les premières, "eye in hand", sont des cas particuliers des secondes, elles autorisent des simplifications dans l'élaboration des commandes mais sont en contrepartie plus sensibles aux erreurs d'étalonnage.

Nous avons décrit des approches où la vision était au service de la commande. L'inverse est également possible, parfois même nécessaire. La vision active regroupe les stratégies de perception visuelles permettant d'atteindre une connaissance beaucoup plus riche de l'environnement (Chaumette 1998). Dans cette configuration, le capteur est indépendant mais, surtout il est actif, c'est-à-dire capable de changer ses paramètres en fonction de l'évolution de la scène. Cela signifie que les caméras peuvent zoomer, ajuster la mise au point, s'orienter ou éventuellement se déplacer (Aloimonos 1993).

4 L'asservissement visuel par apprentissage

L'asservissement visuel est basé sur les transformations inverses à la fois du système robotique et du système de vision. L'emploi des modèles inverses est une pratique usitée. Une boucle dynamique de retour visuel est donc associée au modèle différentiel inverse du robot. A la position courante des variables articulaires s'ajoutent alors les corrections calculées par l'équation (1.16). L'espace de départ de cette fonction est un espace estimé, et l'étape de reconstruction qui permet d'approcher les positions tridimensionnelles représente la transformation inverse de la projection perspective de l'espace 3-D sur deux plans. L'asservissement visuel qui utilise les modèles implique nécessairement des méthodes laborieuses et contraignantes d'étalonnage, à la fois des systèmes robotiques, mais aussi du système de vision. L'utilisation des matrices Jacobiennes rend l'approche tributaire des modèles des robots, du modèle du capteur visuel, et des paramètres des différents éléments constituant la scène. Tout changement dans la configuration de la scène doit alors être explicitement pris en considération et inclus dans l'élaboration de la commande.

Pour élaborer une approche qui ne dépend plus des modèles, nous nous tournons naturellement vers les systèmes d'apprentissage. Ils permettent d'apprendre le système robot-vision, avec ses imprécisions et avec ses imperfections. L'approche neuronale permet en plus de tenir compte implicitement de tout changement de configuration de la scène robotique, ainsi que de tout événement imprévisible. Grâce à leurs propriétés d'apprentissage, notamment à leurs capacités à approximer des fonctions multidimensionnelles à partir d'un nombre réduit d'échantillons, les réseaux de neurones ont émergé dans le domaine du contrôle robotique.

4.1 Une vue d'ensemble

Les réseaux de neurones constituent aujourd'hui indéniablement une approche importante du traitement de l'information et de la science du contrôle. Deux caractéristiques des réseaux neuronaux les rendent particulièrement intéressants, même si la dernière n'est pas toujours favorablement exploitée : ils sont adaptatifs et massivement parallèles. Les modèles de neurones formels les plus efficaces sont des systèmes non linéaires. La non-linéarité, qui est à l'origine de la puissance de ces systèmes, constitue l'obstacle principal pour l'apprentissage et pour l'interprétation des résultats.

Une discussion générale sur les propriétés requises d'un apprentissage dans le domaine du contrôle peut être trouvée dans (Werbos 1992). De nombreuses architectures de réseaux ont été étudiées pour le contrôle des systèmes robot-vision, avec des approches diverses de la commande, du rôle des réseaux et des apprentissages. Parmi les ouvrages qui sont dédiés à l'approche connexionniste on peut citer des recueils d'articles, celui de Miller, Sutton et Werbos (Miller et al. 1990b) concernant le contrôle, celui de Omidvar et van der Smagt (Omidvar et van der Smagt 1997) concernant la robotique et celui de Ge, Lee, et Harris (Ge et al. 1998) concernant la commande des bras manipulateurs.

Plutôt que de proposer une classification des approches, nous proposons une liste des contributions les plus significatives dans le domaine. Parmi les contrôleurs neuromimétiques, on peut citer des modèles très biologiques pour contrôler des mouvements, tel :

- le contrôleur INFANT de Kuperstein (Kuperstein 1988) qui est constitué de cartes sensori-motrices qui agissent sur des couples antagonistes pour positionner un bras en boucle ouverte ;
- le contrôleur MURPHY de Mel (Mel 1990) qui calcule une trajectoire avec deux sous-systèmes en parallèles qui encodent chacun la position de la cible et la vitesse du robot ;
- le modèle DIRECT de Bullock et de Grossberg (Bullock et Grossberg 1987) ;

- l'approche de Jordan (Jordan 1990) qui utilise un réseau MLP et qui prend en compte un vecteur d'état et un vecteur de position désirée défini dans l'espace des tâches pour calculer la consigne ;
- l'approche hiérarchique de Kawato, Maeda, Uno, et Suzuki (Kawato et al. 1990) qui approxime simultanément les modèles cinématiques et dynamiques inverses avec des modules en cascade (Kawato et al. 1988) ;
- l'approche "feedback error learning" (Kawato et al. 1987) qui apprend le modèle cinématique inverse d'un robot 3 axes par un réseau de trois neurones (un par axes) dont les entrées sont constituées de 13 fonctions nonlinéaires des vitesses et accélérations. L'adaptation des 39 poids est faite par une correction d'erreur similaire à la règle LMS. Les paires d'entrées/sorties sont remplacées par la sortie du contrôleur inverse pour déterminer le signal d'erreur. Cette mesure d'erreur est moins précise de celle utilisée par Miller, mais possède l'avantage d'être directement disponible dans la boucle de contrôle. L'apprentissage est réalisé par une méthode de Newton.

D'autres auteurs ont proposés des asservissement visuels adaptatifs avec des soucis moins biologiques, ainsi :

- Kröse, van der Smagt, et Groen (Kröse et al. 1993) se basent sur une approximation de la transformation inverse avec un polynôme de Taylor du second ordre pour approcher des objets statiques lorsque la caméra est portée par le bras ;
- Ritter, Martinetz et Schulten (Ritter et al. 1992) utilisent une carte SOM 3-D avec une règle LMS pour apprendre le modèle cinématique inverse d'un robot 3 axes évoluant dans un espace 3-D ; leurs travaux les ont amenés à proposer des réseaux ESOM pour le positionnement d'un bras robotique en un point défini par les coordonnées image. La position de la cible (en coordonnées image) détermine le neurone qui va être activé et qui va déterminer les coefficients d'une matrice Jacobienne et les angles articulaires du robot. Après un positionnement du bras en boucle ouverte, la matrice Jacobienne permet de réduire itérativement l'erreur de positionnement entre l'effecteur et la cible.
- Miller, Hewes, Glanz et Kraft (Miller et al. 1990a) utilisent le principe d'une table associative avec une correction d'erreur similaire à celle de la règle LMS pour la commande cinématique d'un robot 5 axes ; la table associative est réalisée par un réseau de type CMAC ("Cerebellar Model Articulation Controller" proposé par (Albus 1975)) qui est dotée d'une méthode d'apprentissage rapide pour le contrôle en boucle fermée ; en fait, le principe du CMAC reprend l'idée du SOM étendu avec une règle LMS ; le CMAC est utilisé pour représenter l'espace d'état, alors que le SSOM représente l'espace 3-D de travail discrétisé du robot ; ces informations servent de sélection, leur espace étant discrétisé ;
- Gullapalli, Grupen et Barto (Gullapalli et al. 1992) reprennent le principe de la table associative avec une approche quelque peu différente, puisqu'elle utilise un apprentissage par renforcement ;
- (Hashimoto et al. 1992) utilisent un réseau MLP associé à la règle BP pour apprendre la matrice d'interaction (avec des mouvements aléatoires des caméras) et effectuer la commande d'un robot muni d'une caméra embarquée sur l'effecteur ;
- Jägersand (Jägersand 1996) effectue l'apprentissage de la matrice Jacobienne par une méthode des moindres carrés non-linéaires avec une méthode de quasi-Newton, la méthode d'estimation de Jacobienne de Broyden ; son contrôleur ne dispose pas de l'effet mémoire et il se trouve confronté à un problème de convergence de l'asservissement lorsque le point cible se trouve à une distance

assez importante ; pour contourner ce problème, il propose de disposer des points intermédiaires entre la position courante de l'effecteur et la cible à atteindre ;

- Kubota et Hashimoto (Kubota et Hashimoto 1996) utilisent deux réseaux MLP avec l'algorithme BP pour ajuster leurs poids : un réseau global apprend l'ensemble de l'espace de travail du robot manipulateur et un réseau local apprend le voisinage des points cibles ; c'est une approche "eye in hand", avec six degrés de liberté en simulation et cinq dans les expérimentations réelles ; l'apprentissage du réseau local est favorablement comparé à une méthode d'estimation linéaire, les moindres carrés.

Parmi toutes ces approches, qui réalisent à leur façon l'approximation du modèle inverse du système à contrôler, deux stratégies d'apprentissage coexistent. L'une réside dans un apprentissage global impliquant une structure de réseaux monolithiques (perceptrons multicouches, réseaux polynomiaux), la seconde suggère un apprentissage de la fonction par morceaux impliquant une structure de réseau composite. La première stratégie est confrontée au dilemme de la complexité structurelle et au dilemme de l'optimisation local/global. En privilégiant la simplicité et la multiplicité des modèles d'apprentissage locaux qui constitue la structure d'un réseau composite, la seconde stratégie apporte une solution à chacun de ces dilemmes. La seconde stratégie repose en quelque sorte sur une forme de modularité.

Dans (Massone 1995), l'auteur reprend les différents schémas d'apprentissage existants pour le contrôle adaptatif. Le schéma le plus robuste est sans doute le MRAC ("Model Reference Adaptive Control") de Narendra (Narendra 1990). Notons également que le principe de la modularité et la décomposition de l'apprentissage dédié au contrôle sont discutés dans (Wolpert et Kawato 1998).

Si la plupart des approches précédentes sont des approches directes (2-D, sans reconstruction), il existe des méthodes adaptatives d'asservissement visuel qui utilisent une reconstruction. L'étape de reconstruction utilise des apprentissages non supervisés, les méthodes d'analyse en composantes principales par exemple (Breton 1999). Une seconde étape se focalise alors sur l'estimation de la fonction inverse du système robotique (Jacquet 2001).

Le schéma d'asservissement visuel que nous allons présenter fait partie de la catégorie des asservissements visuels qui utilise une erreur définie dans l'espace des caractéristiques visuelles. La boucle de retour des informations visuelles pourra être considérée comme une boucle de haut niveau : elle servira à commander un contrôleur de bas niveau dont le rôle est de commander chaque axe du robot et d'en assurer la stabilité. C'est une approche par apprentissage basée sur les cartes auto-organisatrices.

4.2 Principe

Les méthodes d'asservissement visuel se fondent en général sur une boucle de retour dynamique et la principale difficulté réside alors dans le calcul de la Jacobienne inverse. En supposant que la cible et l'effecteur soient visibles dans les images, la commande est déterminée à partir de la différence de positionnement de l'effecteur vis-à-vis de la cible. Cette différence constitue une erreur dans l'espace des caractéristiques images, $\mathcal{V} \in \mathbb{R}^m$. Nous allons considérer, sans perte de généralité, que l'espace des caractéristiques images est défini par les coordonnées horizontales et verticales (extraites par un traitement d'images). Dans cet espace, l'effecteur est alors défini par le vecteur $\mathbf{v}_k \in \mathcal{V}$. La cible, c'est-àdire la position désirée, est représentée par le vecteur $\mathbf{d}_k \in \mathcal{V}$. Les deux grandeurs $\mathbf{v}_k$ et $\mathbf{d}_k$ permettent de déterminer, à l'itération courante, une erreur de positionnement $\mathbf{e}_k$ de la façon suivante :

$$\mathbf{e}_k = \mathbf{d}_k - \mathbf{v}_k . \tag{1.17}$$

Considérons la fonction vectorielle $\mathbf{f(.)}$, réalisant la transformation cinématique (ou différentielle) entre l'espace articulaire du robot de dimension q, $\mathcal{Q} \in \mathbb{R}^q$, et l'espace des caractéristiques images $\mathcal{V}$:

$$\mathbf{f} : \mathcal{Q} \rightarrow \mathcal{V}, \ \mathbf{v}_k = \mathbf{f}(\boldsymbol{\theta}_k), \tag{1.18}$$

où $\boldsymbol{\theta}_k \in \mathcal{Q}$ est le vecteur des variables articulaires du robot à l'instant k.

Notons que cette fonction dépend en effet à la fois des paramètres liés au robot, des paramètres de la caméra, et de la position relative caméra-robot. L'utilisation directe de son développement mathématique suppose la pleine connaissance de l'ensemble de ces paramètres avec une précision suffisante.

La matrice Jacobienne $\mathbf{J}$ de la fonction $\mathbf{f(.)}$ relie à un petit déplacement $\partial\boldsymbol{\theta}_k$ dans l'espace des angles, le déplacement $\partial\mathbf{v}_k$ correspondant dans l'espace des images :

$$\partial\mathbf{v}_k = \mathbf{J}\partial\boldsymbol{\theta}_k . \tag{1.19}$$

Selon les auteurs, la matrice Jacobienne $\mathbf{J}$ est appelée matrice des sensibilités des caractéristiques image (Weiss 1984), matrice d'interaction (Chaumette 1990), ou plus simplement matrice Jacobienne image (Hutchinson et al. 1996).

L'asservissement visuel s'appuie sur une approximation linéaire, dans le voisinage du point d'opération, en impliquant l'inverse $\mathbf{J}^{-1}$ de la matrice d'interaction. Plutôt que d'être calculée, la matrice inverse $\mathbf{J}^{-1}$ est estimée par apprentissage à l'aide des observations faites pendant les mouvements du robot. En effet, la matrice $\mathbf{J}$ permet également de relier les déplacements entre deux instants :

$$\Delta\mathbf{v}_k = \mathbf{J}\Delta\boldsymbol{\theta}_k . \tag{1.20}$$

Elle permet alors de déduire la correction angulaire $\Delta\boldsymbol{\theta}_k$ à appliquer au robot :

$$\Delta\boldsymbol{\theta}_k = \mathbf{J}^{-1}\mathbf{e}_k . \tag{1.21}$$

Dans les relations précédentes (1.20) et (1.21), $\Delta\mathbf{v}_k = \mathbf{v}_k - \mathbf{v}_{k-1}$ représente la vitesse de déplacement mesurée dans les caractéristiques image, et $\Delta\boldsymbol{\theta}_k = \boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}$ représente la vitesse de déplacement mesurée dans l'espace articulaire. Nous avons retenu ce schéma de commande dont le principe est reproduit sur la Figure 1-3.

Les approches adaptatives se proposent d'approcher l'estimation $\hat{\mathbf{J}}_k^{-1}$ de l'inverse de la matrice d'interaction avec des techniques itératives :

$$\hat{\mathbf{J}}_{k+1}^{-1} = \hat{\mathbf{J}}_k^{-1} + \Delta\mathbf{J}_k . \tag{1.22}$$

La règle généralement utilisée pour calculer la correction $\Delta\mathbf{J}_k$ de l'expression (1.22) est du type LMS. Elle est utilisée dans (Jägersand 1996), dans (Ritter et al. 1992), ou encore dans (Urban et al. 1998). Dans le premier cas, cette matrice est adaptée en fonction de la situation, sans être mémorisée. Dans les deux cas suivants, la notion de mémoire est adjointe à celle de l'adaptation pour constituer un véritable système d'apprentissage. Dans le dernier cas, l'apprentissage de $\mathbf{J}^{-1}$ est formalisé par des réseaux Adalines, et l'architecture du réseau neuronal permet de tenir compte du fait que l'estimation Jacobienne est locale et fonction de la position courante $\boldsymbol{\theta}_k$, autrement dit d'associer une matrice Jacobienne à chacun des Adalines. Le processus de sélection du réseau Adaline est réalisé par une représentation discrétisée de l'espace d'entrée à l'aide d'une carte auto-organisatrice.

Si l'approche est validée, ses perspectives d'application sont limitées par la dimensionnalité du problème. Un tel réseau, pour contrôler plus de 3 axes, n'est pas tangible avec une application temps réel. L'utilisation de cette technique continue à être explorer, notamment dans (Buessler 1999) où est introduite l'idée d'associer plusieurs réseaux.

5 Les cartes auto-organisatrices

Les principes génériques des réseaux de neurones, leurs types d'apprentissage, les paradigmes et les propriétés des apprentissages, les types de structures, fixes ou évolutives, sont présentés en

annexe. Nous allons dans cette partie nous intéresser à un réseau de neurones particulier, la carte auto-organisatrice SOM de Kohonen. Nous allons utiliser cette carte par la suite pour résoudre le problème de la commande robotique.

Après une présentation de l'algorithme développé et présenté par Kohonen, plusieurs extensions – les réseaux SSOM et ESOM proposés par (Ritter et al. 1992) – sont exposées. La première variante possède une sortie supervisée qui permet d'étendre ce réseau SOM à d'autres applications que celles qui lui étaient destinées au départ. La seconde variante, la carte étendue permet d'apprendre la relation entre deux espaces. La littérature anglophone parle de "mapping".

5.1 La carte auto-organisatrice de Kohonen

La carte auto-organisatrice, communément désignée carte SOM (pour "Self-Organizing Map"), et développée par Kohonen en 1982, est restée de nombreuses années ignorée malgré son grand intérêt. Ce réseau de neurones a depuis largement montré son efficacité. De nombreux raffinements et différentes variantes (notamment les algorithmes de quantification vectorielle par apprentissage, les LVQ) ont depuis été élaborés. Son application dépasse largement le cadre du contrôle robotique (Kohonen 1995).

L'intérêt des cartes de Kohonen est à la fois théorique et pratique, parce qu'elles introduisent notamment la notion d'interaction entre neurones et qu'elles utilisent des algorithmes relativement simples à mettre en œuvre.

Les neurones d'une carte auto-organisatrice sont disposés sur une grille régulière multidimensionnelle. La plupart des applications se limitent volontairement à des grilles ne dépassant pas trois dimensions. Au delà, les données deviennent difficilement interprétables.

La notion de grille de neurones est un artifice théorique qui permet d'introduire des interactions entre les neurones à travers un concept fondamental pour ce type de réseau neuronal : la notion de voisinage. Un neurone actif va ainsi pouvoir influencer ses neurones les plus proches dans la grille. C'est une qualité essentielle des cartes, puisque l'apprentissage sera ainsi à la fois plus rapide et plus stable.

La disposition des neurones dans la grille est immuable. Tous les neurones de la grille, considérés comme faisant partie d'une seule couche, sont reliés à une couches d'entrée. Toutes les valeurs (scalaires) présentées en entrée à un instant k, sont regroupées dans un vecteur d'entrée $\mathbf{x}_k$.

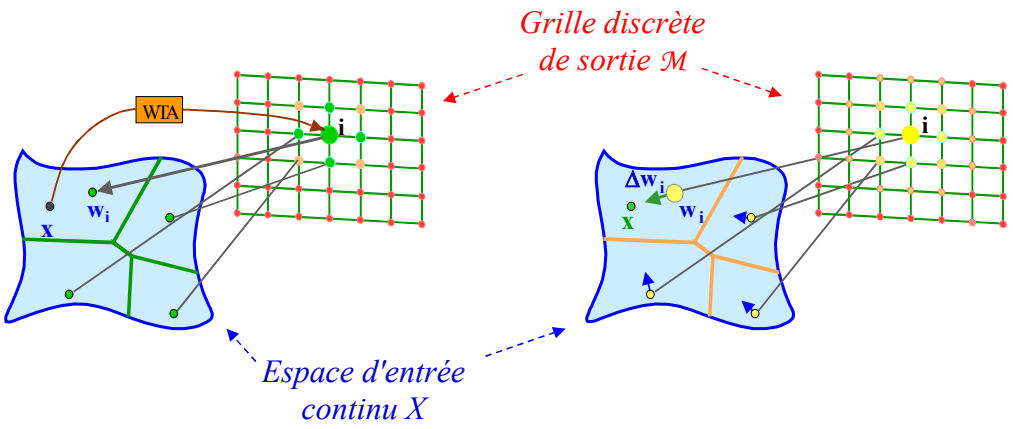


Figure 1-4 : Une carte de Kohonen bidimensionnelle.

L'étape de compétition est représentée à gauche et l'étape d'adaptation des poids du neurone vainqueur, ainsi que celle des poids de ses quatre plus proches voisins, est montrée à droite.

5.1.1 Principe

Le mécanisme d'apprentissage de la carte de Kohonen comprend deux phases pour chaque point de l'espace d'entrée présenté : une phase de compétition entre les neurones de la carte, suivie d'une phase d'adaptation des poids des neurones. Les principes des deux phases sont illustrés sur la Figure 1-4.

Si X est un espace de taille *m* dans lequel est répartie une distribution de points, chaque point est représenté par un vecteur $\mathbf{x} \in X$.

Soit $\mathcal{M}$ une grille de neurones de dimension M. Le vecteur poids synaptique $\mathbf{w}$ de chaque neurone possède une représentation dans l'espace d'entrée, il peut être vu comme un vecteur de référence. De cette façon, à un instant *k* où est présenté un vecteur prototype $\mathbf{x}_k$ tiré de la distribution de l'espace d'entrée, tous les neurones de la grille sont mis en compétition. Cette compétition revient à chercher un neurone *vainqueur*, c'est-à-dire celui qui se rapproche le plus du vecteur d'entrée. En d'autres termes, parmi tous les neurones de la carte, le neurone vainqueur d'indice $\mathbf{i}_k = \mathbf{i}(\mathbf{x}_k)$ est celui dont la distance entre son vecteur poids synaptiques et le vecteur d'entrée est la plus courte :

$$\mathbf{i}(\mathbf{x}) = \arg\min_{\mathbf{j}\in\mathcal{M}} \left\| \mathbf{x}_k - \mathbf{w}_{\mathbf{j},k} \right\|. \tag{1.23}$$

Le neurone vainqueur, pour un stimulus donné est également dénommé centre d'excitation de la carte. La distance généralement utilisée entre les vecteurs $\mathbf{x}$ et $\mathbf{w}$ est la distance Euclidienne mais tout autre calcul de distance peut être envisagé.

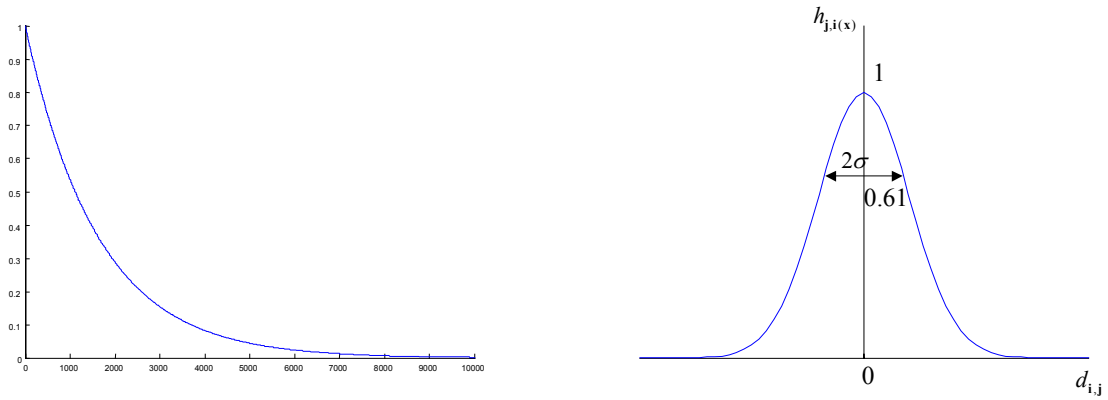
Le processus de compétition permet de déterminer la façon d'ajuster les poids des neurones de la carte. Ainsi, les vecteurs poids synaptiques $\mathbf{w}_\mathbf{j}$ des neurones d'indice $\mathbf{j}$ de la carte auto-organisatrice sont mis à jour par correction d'erreur (l'erreur est définie comme la distance entre le vecteur de prototype $\mathbf{x}$ et le vecteur de référence $\mathbf{w}_\mathbf{j}$ du neurone considéré) :

$$\begin{aligned} \mathbf{w}_{\mathbf{j},k+1} &= \mathbf{w}_{\mathbf{j},k} + \Delta\mathbf{w}_{\mathbf{j},k} \\ &= \mathbf{w}_{\mathbf{j},k} + \eta_k h_{\mathbf{j},\mathbf{i}(\mathbf{x}_k),k} \left(\mathbf{x}_k - \mathbf{w}_{\mathbf{j},k} \right) \end{aligned} \tag{1.24}$$

avec

$$\Delta\mathbf{w}_{\mathbf{j},k} = \eta_k h_{\mathbf{j},\mathbf{i}(\mathbf{x}_k),k} \left(\mathbf{x}_k - \mathbf{w}_{\mathbf{j},k} \right). \tag{1.25}$$

Dans les expressions (1.24) et (1.25) η_k représente le coefficient d'apprentissage et, $h_{\mathbf{j},\mathbf{i}(\mathbf{x}_k),k}$ est une fonction de voisinage. L'adaptation des poids de chaque neurone est donc fonction de la position d'un neurone dans la grille $\mathcal{M}$ par rapport au neurone élu.



a) L'évolution du coefficient d'apprentissage au cours de l'apprentissage

b) L'allure de la fonction de voisinage pour un rayon donné ($\sigma = 0.61$)

Figure 1-5 : Evolution des paramètres d'une carte au cours de l'apprentissage.

Le coefficient d'apprentissage (à gauche) décroît à partir d'une valeur initiale unitaire pour tendre vers une valeur finale proche de zéro (0.002) sur une période de 10^4 itérations. La fonction Gaussienne de voisinage $h_{\mathbf{j},\mathbf{i}(\mathbf{x})}$ représentée à droite dépend de la distance $d_{\mathbf{i},\mathbf{j}}$ du neurone considéré par rapport au neurone élu.

Le coefficient d'apprentissage, pour rendre l'apprentissage plus performant, est généralement fonction du temps et du type :

$$\eta_k = \eta_i \left(\frac{\eta_f}{\eta_i} \right)^{\frac{k}{k_{\max}}}, \tag{1.26}$$

où η_i représente la valeur initiale du coefficient, η_f sa valeur finale, et $k_{\max}$ détermine la durée de l'apprentissage. L'évolution temporelle du coefficient d'apprentissage est illustrée sur la Figure 1-5.

La fonction de voisinage généralement adoptée est la fonction Gaussienne (invariante selon la translation, c'est-à-dire qu'elle ne dépend pas du neurone vainqueur). Cette fonction est centrée sur le neurone déclaré vainqueur après la phase de compétition qui a suivi la présentation d'un vecteur d'entrée.

Il a été montré, dans (Erwin et al. 1992a), qu'une carte SOM utilisant une fonction de voisinage non convexe, aboutit à des états métastables. La fonction Gaussienne est cependant convexe, alors que la fonction rectangulaire (la carte originale proposée par Kohonen se servait d'une fonction rectangulaire) ne l'est pas. De plus, les mêmes auteurs ont remarqués qu'une carte auto-organisatrice basée sur une fonction Gaussienne converge plus rapidement que celle avec une fonction rectangulaire (Erwin et al. 1992b).

La fonction de voisinage, pour pondérer la correction des poids synaptiques $\Delta\mathbf{w}_{\mathbf{j},k}$ du neurone $\mathbf{j}$ à l'instant *k*, tient compte de sa distance par rapport à la position du neurone vainqueur. Soit $d_{\mathbf{j},\mathbf{i}}$ la distance entre le neurone vainqueur d'indice $\mathbf{i}$ et un neurone voisin d'indice $\mathbf{j}$. Cette distance ne se calcule pas dans l'espace des entrées mais dans l'espace topologique de la carte :

$$d_{\mathbf{j},\mathbf{i}}^2 = \left\| \mathbf{j} - \mathbf{i} \right\|^2 . \tag{1.27}$$

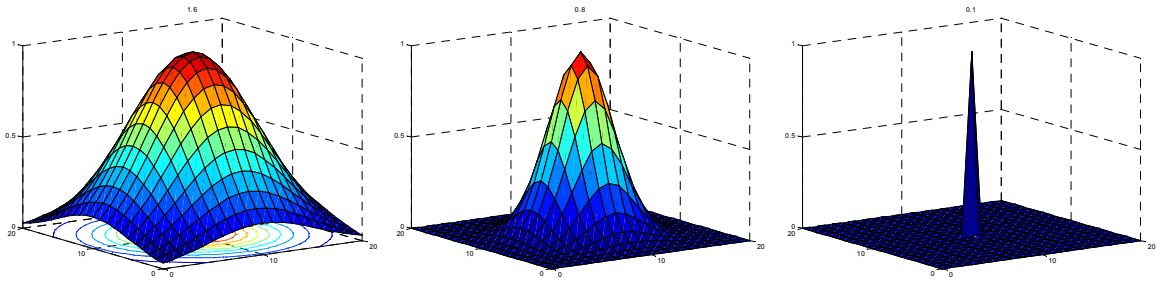
La fonction de voisinage $h_{\mathbf{j},\mathbf{i}(\mathbf{x}_k),k}$ s'écrit alors :

$$h_{\mathbf{j},\mathbf{i}(\mathbf{x}),k} = \exp\left(-\frac{d_{\mathbf{j},\mathbf{i}}^2}{2\sigma^2} \right), \tag{1.28}$$

où σ est un rayon de voisinage. Ce rayon σ peut être dépendant du temps selon l'expression suivante :

$$\sigma_k = \sigma_i \left(\frac{\sigma_f}{\sigma_i} \right)^{\frac{k}{k_{\max}}}, \tag{1.29}$$

avec σ_i sa valeur initiale et σ_f sa valeur finale. Il faut noter que lorsque $\mathbf{j} = \mathbf{i}$, alors le voisinage pour le neurone vainqueur vaut $h_{\mathbf{i},\mathbf{i},k} = 1$.



a) $\sigma = 1.6$ b) $\sigma = 0.8$ b) $\sigma = 0.1$

Figure 1-6 : Evolution de la fonction de voisinage au cours de l'apprentissage.

Le voisinage est volontairement grand au départ pour autoriser un apprentissage plus rapide. Il est progressivement réduit en diminuant le rayon σ afin de favoriser la mémoire à long terme du réseau de neurones. Sur cette figure la fonction de voisinage est représentée pour des valeurs de σ de 1.6, de 0.8 et de 0.1 (de la gauche vers la droite).

La Figure 1-6 illustre l'évolution de la fonction de voisinage d'une carte SOM bidirectionnelle comprenant 20 neurones dans chaque dimension. Le neurone vainqueur, d'indice $\mathbf{i} = [10 \quad 10]^T$, est le même pour les trois courbes.

5.1.2 Propriétés

L'originalité de la carte auto-organisatrice de Kohonen réside dans l'emploi d'une fonction de voisinage. Ce principe est tellement intéressant qu'il a été repris dans le développement de nouveaux réseaux. La carte SOM, comme presque toutes les cartes, se caractérise par un mode d'apprentissage dit non supervisé. Seules les données issues de l'espace des entrées sont nécessaires à l'apprentissage.

Les propriétés essentielles des cartes topologiques sont leurs interprétations géométriques. Les cartes SOM sont reconnues pour leur propension à traiter des grandeurs topologiques.

Le vecteur poids synaptique associé à chaque neurone représente la position du neurone dans un espace. Cet espace est celui dans lequel sont tirés les points de la distribution d'entrée à apprendre. D'un point de vue pratique, au départ l'initialisation des poids se fait aléatoirement dans un espace contenu dans l'espace à apprendre. Pendant l'apprentissage, les neurones vont, en s'adaptant virtuellement, se déplacer dans l'espace d'entrées en déformant et en étirant la grille. Lorsque suffisamment de prototypes de l'espace d'entrée ont été présentés, les neurones de la grille sont, après apprentissage, répartis de façon plus ou moins régulière dans l'espace d'entrée, réalisant ainsi de facto une discrétisation naturelle. Chaque neurone représente alors un centre d'attraction. Les sorties de l'architecture SOM définissent les régions d'influence comme étant l'ensemble des points les plus proches des $\mathbf{w}_j$ au sens de la distance utilisée. L'ensemble de ces régions réalise une partition de l'espace selon un pavage de Voronoï, qui dépend de la distance utilisée (Hérault et Jutten 1994). La carte s'adapte dynamiquement à l'espace des entrées et en détermine le meilleur pavage.

En plus de cette propriété de quantification ou de discrétisation, les cartes SOM permettent de préserver les propriétés topologiques de l'espace d'entrée. Autrement dit, deux neurones voisins dans la grille sont représentatifs de deux régions voisines dans l'espace d'entrée.

Une interprétation plus statistique conclue qu'après l'apprentissage correct d'une distribution de points, la distribution des vecteurs de référence reflète celle de l'espace d'entrée.

Tout au long de l'exposé qui va suivre, la carte de Kohonen sera représentée par le schéma simplifié de la Figure 1-7.

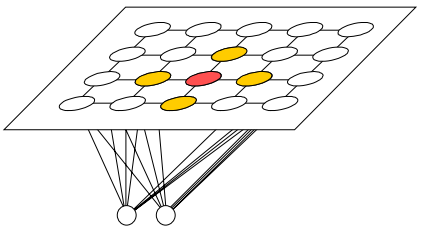


Figure 1-7 : Représentation graphique d'une carte de Kohonen bidimensionnelle.
Cette carte possède deux entrées. Le neurone vainqueur à un instant donné, ainsi que ses quatre plus proches voisins, sont représentés par des couleurs.

Il est à remarquer que dans le principe de fonctionnement d'une carte SOM, en plus d'une phase d'apprentissage, on retrouve deux grand principes récurrents dans le domaine des réseaux neuronaux, la compétition et la coopération.

1. La compétition pour le choix du neurone vainqueur, cf. équation (1.23) :
   il s'agit de distinguer parmi tous les neurones, le plus apte à répondre, traduit dans un sens géométrique par le plus proche du vecteur d'entrée proposé.
2. La coopération pour le partage des connaissances à l'aide de la fonction de voisinage, cf. équation (1.24) :
   lorsqu'un neurone a été choisi comme centre d'excitation de la carte, il partage ses connaissances avec ses voisins les plus proches en les forçant à s'adapter, c'est-à-dire en s'approchant au sens géométrique de la position du stimulus.

Ces deux propriétés, ainsi que l'interprétation géométrique des cartes auto-organisatrices, rendent ces réseaux neuronaux fort intéressant pour les applications de commande robotique.

5.2 La carte auto-organisatrice étendue

Pour utiliser les propriétés avantageuses des cartes auto-organisatrices, Ritter et Schulten (Ritter et Schulten 1986) ont été amené à modifier le modèle initialement proposé par Kohonen. En effet, la carte de Kohonen se contente de fournir un résultat suivant une représentation dans la grille des neurones. Pour effectuer une tâche de contrôle robotique, comme pour approximer une fonction, le problème se pose en terme d'apprentissage et d'estimation de la relation liant deux espaces, un espace d'entrée et un espace de sortie.

Pour cette problématique, une carte auto-organisatrice étendue, appelé ESOM pour "Extended SOM" a été proposée. Cette carte adjoint à chaque neurone d'une grille $\mathcal{M}$ un vecteur de poids synaptiques supplémentaire. Comme les poids initiaux représentent des vecteurs position dans l'espace d'entrée, ces poids supplémentaires deviennent des vecteurs position dans l'espace de sortie. Ces poids synaptiques permettent alors de relier la grille à l'espace de sortie, donc d'associer à une portion de l'espace d'entrée, un vecteur référence directement dans l'espace de sortie.

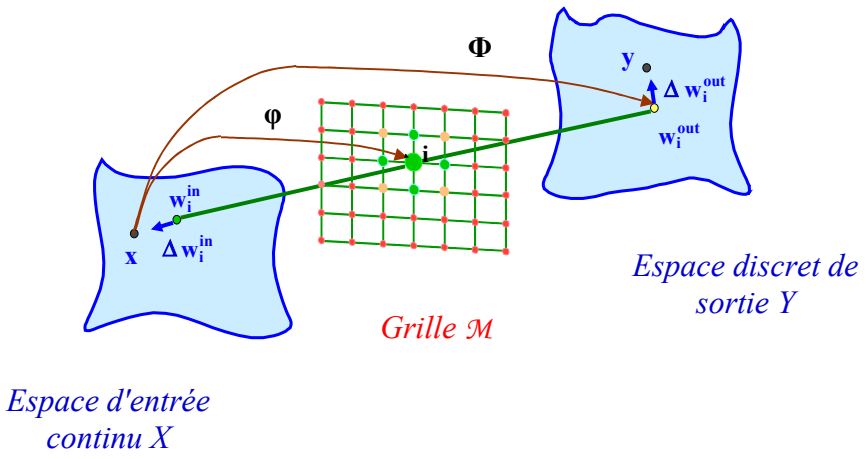


Figure 1-8 : Une carte auto-organisatrice étendue.

Dans cette approche, une distribution de points d'un espace d'entrée X est estimée par une grille $\mathcal{M}$. Ceci est effectué par les poids synaptiques $\mathbf{w}_{\mathbf{j},k}^{in}$ des neurones $\mathbf{j}$ de la carte. Les neurones sont liés à l'espace Y de sortie par des connexions synaptiques supplémentaires $\mathbf{w}_{\mathbf{j},k}^{out}$ comme le montre la Figure 1-8. Chaque neurone peut donc être vu comme ayant une position dans les deux espaces, X et Y ; ces positions sont données respectivement par les vecteurs $\mathbf{w}_{\mathbf{j},k}^{in}$ et $\mathbf{w}_{\mathbf{j},k}^{out}$.

Pour chaque vecteur $\mathbf{x}_k$ présenté en entrée, il faut procéder à l'élection du neurone vainqueur :

$$\mathbf{i}(\mathbf{x}) = \arg\min_{\mathbf{j}\in\mathcal{M}} \left\| \mathbf{x}_k - \mathbf{w}_{\mathbf{j}} \right\| . \tag{1.30}$$

Les vecteurs poids synaptiques $\mathbf{w}_{\mathbf{j}}^{in}$ et $\mathbf{w}_{\mathbf{j}}^{out}$ des neurones sont mis à jour selon les règles suivantes :

$$\mathbf{w}_{\mathbf{j},k+1}^{in} = \mathbf{w}_{\mathbf{j},k}^{in} + \eta_k h_{\mathbf{j},\mathbf{i}(\mathbf{x}_k),k} \left(\mathbf{x}_k - \mathbf{w}_{\mathbf{j},k}^{in} \right), \tag{1.31}$$

$$\mathbf{w}_{\mathbf{j},k+1}^{out} = \mathbf{w}_{\mathbf{j},k}^{out} + \eta_k^{'} h_{\mathbf{j},\mathbf{i}(\mathbf{x}_k),k}^{'} \left(\mathbf{y}_k - \mathbf{w}_{\mathbf{j},k}^{out} \right). \tag{1.32}$$

Dans la dernière expression (1.32), le vecteur $\mathbf{y}_k$ représente la sortie désirée. L'adaptation des deux types de poids demande des coefficients d'apprentissage η_k et $\eta_k^{'}$ et des fonctions de voisinage $h_{\mathbf{j},\mathbf{i}(\mathbf{x}_k),k}$ et $h_{\mathbf{j},\mathbf{i}(\mathbf{x}_k),k}^{'}$ distincts.

Le vecteur de sortie $\hat{\mathbf{y}}_k$ de ce réseau neuronal est directement fourni par un vecteur de référence dans l'espace de sortie, celui associé au vecteur de référence le plus proche du vecteur d'entrée :

$$\hat{\mathbf{y}}_k = \mathbf{w}_{\mathbf{i}(\mathbf{x}),k}^{out}. \tag{1.33}$$

La Figure 1-8 met en évidence deux fonctions, $\boldsymbol{\varphi}$ qui relie l'espace d'entrée à la grille et $\boldsymbol{\Phi}$ qui relie l'espace d'entrée à celui de sortie.

Mathématiquement, on a :

$$\boldsymbol{\varphi} : X \to \mathcal{M}, \ \mathbf{x}_k \mapsto \boldsymbol{\varphi}(\mathbf{x}_k) = \mathbf{i}(\mathbf{x}), \tag{1.34}$$

$$\boldsymbol{\Phi} : X \to Y, \ \mathbf{x}_k \mapsto \boldsymbol{\Phi}(\mathbf{x}_k) = w_{\mathbf{i}(\mathbf{x}),k}^{out} = \hat{\mathbf{y}}_k. \tag{1.35}$$

Si la fonction $\boldsymbol{\varphi}$ est moins intéressante pour les tâches proposées (l'expression (1.34) peut être vue comme une abstraction de (1.30)), à savoir l'approximation de fonctions ou le contrôle robotique, la seconde répond parfaitement au besoin. Le réseau peut même être considéré comme une table associative. La réponse obtenue est discrète. La carte ESOM réalise la discrétisation des deux espaces, d'entrée et de sortie, et sa précision est directement liée au nombre de neurones.

La carte ESOM permet de résoudre des problèmes très concrets sans les complexifier. Sa mise en œuvre est très simple, mais un inconvénient majeur surgit lorsqu' un des deux espaces est grand et requiert beaucoup de neurones. Il est cependant judicieux d'utiliser cette carte lorsque les distributions de points à apprendre ne sont pas les mêmes dans les deux espaces, celui d'entrée et celui de sortie. La carte ESOM compense le fait que la transformation d'un espace en un autre ne conserve pas la distribution des points par le positionnement des neurones dans les deux espaces.

5.3 La carte à sortie supervisée

Avec comme souci majeur l'amélioration des performances des cartes SOM, Ritter et Schulten ont proposé une seconde extension (Ritter et Schulten 1988b), une carte auto-organisatrice à sortie supervisée appelée SSOM ("Supervised Self-Organizing Map"). Cette carte combine le procédé non supervisé de discrétisation de l'espace d'entrée et le calcul d'une sortie linéarisée basée sur un apprentissage supervisé.

Chaque neurone de la carte de Kohonen est enrichi de poids synaptiques supplémentaires, ajustés itérativement à l'aide de la règle LMS. Un formalise plus récent permet de considérer ces poids comme constituant des éléments linéaires adaptatifs ou *Adaline*. Les réseaux Adalines ont été proposés par Widrow en 1960 et une synthèse complète sur ces réseaux se trouvent dans (Widrow et Walach 1995).

L'extension SSOM prend en compte deux entrées, $\mathbf{x}_k$ qui concerne la couche de Kohonen et $\mathbf{u}_k$ qui concerne la couche des Adalines. Comme pour l'algorithme de Kohonen, le calcul de la sortie passe par la phase de compétition des neurones donnée par l'expression (1.23) lorsqu'un vecteur d'entrée $\mathbf{x}_k$ est présenté. La couche des Adalines permet d'estimer la relation entre les entrées et les sorties par une relation linéaire. La sortie estimée par le réseau se calcule par le produit

$$\hat{\mathbf{y}}_k = \mathbf{A}_{\mathbf{i}(\mathbf{x}),k} \mathbf{u}_k, \tag{1.36}$$

où $\mathbf{A}_{\mathbf{j}}$ représente les poids synaptiques de l'Adaline associé au neurone j de la carte de Kohonen. En particulier, $\mathbf{A}_{\mathbf{i}(\mathbf{x}),k}$ est la matrice poids de l'Adaline associé au neurone vainqueur par rapport à l'entrée $\mathbf{x}_k$.

La phase d'adaptation se doit d'ajuster à la fois les poids des neurones de la carte de Kohonen et les poids des Adalines qui leur sont associés :

$$\mathbf{w}_{\mathbf{j},k+1} = \mathbf{w}_{\mathbf{j},k} + \eta_k h_{\mathbf{j},\mathbf{i}(\mathbf{x}_k),k} \left(\mathbf{x}_k - \mathbf{w}_{\mathbf{j},k} \right), \tag{1.37}$$

$$\mathbf{A}_{\mathbf{j},k+1} = \mathbf{A}_{\mathbf{j},k} + \eta_k^{'} h_{\mathbf{j},\mathbf{i}(\mathbf{x}_k),k}^{'} \left(\mathbf{A}^* - \mathbf{A}_{\mathbf{j},k} \right), \tag{1.38}$$

où le modèle de poids $\mathbf{A}^*$ est calculé selon la règle LMS normalisé :

$$\mathbf{A}^* = \mathbf{A}_{\mathbf{i}(\mathbf{x}),k} + \frac{\boldsymbol{\varepsilon}_k . \mathbf{u}_k^T}{\left\| \mathbf{u}_k \right\|^2}, \tag{1.39}$$

avec $\boldsymbol{\varepsilon}_k$ l'erreur entre la sortie désirée et la sortie estimée, $\boldsymbol{\varepsilon}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k$.

La convergence de l'algorithme LMS n'est plus à prouver (Widrow et Walach 1995). Une étude récente (Wang et al. 2000) montre tout de même qu'il faut parfois être prudent avec cette règle d'apprentissage. Il faut notamment ne pas laisser la norme du vecteur $\mathbf{u}_k$ s'annuler sous peine de perdre toute la connaissance déjà acquise par le réseau.

Contrairement à la carte ESOM, la carte SSOM ne fournit pas de sortie discrétisée mais continue (linéarisée). Il est intéressant de noter que d'un point de vue conceptuel, la structure de ce réseau est parfaitement bien adaptée à l'approximation de matrice Jacobienne. Cette notion clé a été reprise pour le développement d'une variante de la carte SSOM pour l'asservissement visuel d'un robot, le NeuroModule (Buessler et al. 1996). Ce réseau est représenté par la Figure 1-9. La carte de Kohonen peut se voir comme une couche de sélection. A un instant k donné, le vecteur $\mathbf{x}_k$ permet de sélectionner le neurone vainqueur. L'Adaline, représenté par une colonne sur cette figure, est alors stimulé par l'entrée $\mathbf{u}_k$ pour fournir une estimation. Ce réseau permet de réaliser une commande robotique basée sur une approche linéaire locale (Buessler 1999). La réponse est déterminée avec un unique Adaline, mais l'adaptation des poids des Adalines concerne tous les Adalines sélectionnés par la fonction de voisinage définie dans l'espace de la carte auto-organisatrice. C'est le principe du partage des connaissances. Il permet d'accélérer l'apprentissage du réseau.

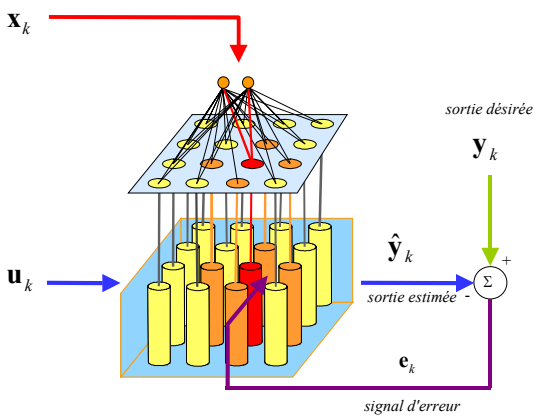


Figure 1-9 : Représentation d'un NeuroModule.

La carte de Kohonen sélectionne parmi les réseaux Adalines (représentés par des colonnes) associés aux neurones de la carte, celui qui est à même de répondre.

Les extensions ESOM et SSOM peuvent être fusionnées pour donner une nouvelle variante de carte auto-organisatrice supervisée. Ceci se fait en associant, dans le calcul de la sortie une estimation

des moments d'ordre 1 et 2, qui sont respectivement des estimations des moyennes et des vitesses des signaux de sortie :

$$\hat{\mathbf{y}}_k = \mathbf{w}_{\mathbf{j},k}^{out} + \mathbf{A}_{\mathbf{i}(\mathbf{x}),k}\left(\mathbf{x}_k - \mathbf{w}_{\mathbf{j},k}^{\text{in}}\right). \tag{1.40}$$

Le moment d'ordre 1, $\mathbf{w}_{\mathbf{j},k}^{out}$, est approximé par le principe de l'ESOM, alors que le moment d'ordre 2, $\mathbf{A}_{\mathbf{i}(\mathbf{x}),k}$, l'est par le principe SSOM. L'estimation $\hat{\mathbf{y}}_k$ fournie par le réseau après apprentissage revient à une expansion linéaire de la relation $\mathbf{y} = \mathbf{F}(\mathbf{x})$.

5.4 Considérations supplémentaires

Les cartes de Kohonen, dans toutes les applications recensées (Kaski et al. 1998), sont utilisées soit pour leurs propriétés de quantification, soit pour leurs propriétés d'ordonnancement. Ces deux propriétés apparaissent au cours de deux phases successives pendant l'apprentissage. A vrai dire, la carte SOM lorsqu'elle se déploie, présente une première phase d'auto-organisation suivie d'une seconde phase de convergence. Pendant la première phase, les coefficients d'apprentissage et de voisinage sont grands et la carte réalise une discrétisation très grossière de l'espace d'entrée. Pendant la seconde phase, les coefficients sont petits et la carte cherche à affiner sa représentation organisée de l'espace d'entrée et à mieux le quantifier (les déplacements des neurones sont alors très petits).

Malgré ses nombreuses applications et utilisations, la carte de Kohonen résiste à toute démonstration mathématique de sa convergence. Le seul cas abouti à ce jour est celui d'une carte monodimensionnelle : l'espace d'entrée comporte une dimension unique et la carte est une carte linéaire dont les neurones ne sont disposés que le long d'une seule dimension.

Des tentatives de démonstration ont été données par Kohonen lui même dans (Kohonen 1982a; Kohonen 1984). Une démonstration complète viendra plus tard (Cottrell et Fort 1987) mais se limite au cas d'une distribution d'entrée uniforme et à un voisinage carré. La démonstration a été généralisée pour d'autres distributions dans (Bouton et Pagès 1993) ou dans (Erwin et al. 1992a), puis avec d'autres fonctions de voisinage (Erwin et al. 1992b). Une démonstration complète se trouve dans (Fort et Pagès 1995). Des contributions théoriques sont cependant proposées dans différents contextes. Dans (Ritter et al. 1992) par exemple, les auteurs introduisent une fonction de potentiel lorsque les points d'entrée ne peuvent prendre que des valeurs discrètes, de façon à ce que l'algorithme de Kohonen réalise une descente de gradient stochastique, du moins pour chaque valeur constante de la fonction de voisinage. Beaucoup de travaux ne se contentent d'étudier la convergence que pendant la seconde phase de déploiement de la carte.

Pour des cartes multidimensionnelles, les résultats ne sont que partiels. La difficulté majeure réside en fait dans le manque d'outils mathématiques pour définir un ordonnancement correct dans des espaces de dimension supérieure à un.

Les principales limitations des cartes de Kohonen sont :

- une certaine difficulté pour apprendre les bords des espaces,
- un placement de neurones inutiles dans les régions vides des espaces non convexes,
- et des coefficients d'apprentissage difficiles à régler pour un apprentissage optimisé puisqu'ils dépendent des propriétés des données de l'application, donc des connaissances a priori qui sont disponibles (ou pas).

Une démarche empirique permet de régler les coefficients pour limiter les divergences (un repliement de la carte par exemple, n'est pas une solution finale optimale). La baisse du coefficient d'apprentissage et une diminution du rayon de voisinage comme celles des équations (1.26) et (1.29) suffisent en règle générale.

Dans (Haese et Goodhill 2001), une carte SOM, appelée Auto-SOM, est enrichie d'un algorithme qui estime automatiquement le coefficient d'apprentissage. L'algorithme consiste en un filtre de Kalman couplé avec une méthode d'estimation récursive de paramètres. Le filtre calcul l'adaptation des poids du réseau en estimant le coefficient d'apprentissage qui minimise la variance de

l'erreur d'estimation, alors que la méthode d'estimation calcule la largeur de la fonction de voisinage qui minimise la variance de l'erreur de prédiction du filtre de Kalman. A cela s'ajoute une "topographic function" qui permet de préserver le voisinage et prévient ainsi la non convergence de la carte.

Avec une fonction de voisinage mal choisie la carte peut, dans sa grille, ne plus refléter les propriétés topologiques de l'espace d'entrée qu'elle est originalement supposée reproduire.

Dans la mesure où le but recherché dans (Haese et Goodhill 2001) est la préservation des propriétés topologiques de l'espace d'entrée, l'auteur se voit dans l'obligation de mesurer la qualité de la topographie pendant la phase d'apprentissage. Le produit topographique est utilisé (Bauer et Pawelzik 1992). Dans (Haese 1998), c'est une fonction topologique proposée dans (Villmann et al. 1997) qui est utilisée. Cette fonction est une technique reconnue pour fournir une bonne qualité de la mesure de la topologie des cartes auto-organisatrices (Bauer et al. 1999; Bauer et Pawelzik 1992). D'autres études sont consacrées à la préservation des propriétés topologiques de l'espace d'entrée, (Herrmann 1995 ; Herrmann et al. 1997).

La carte Auto-SOM est mise à l'épreuve sur trois applications, l'approximation d'une distribution uniforme bidimensionnelle par une carte bidimensionnelle et la réduction de dimension d'une distribution non linéaire de dimension trois, avec et sans préservation des propriétés topologiques, grâce à une carte bidimensionnelle. Les performances de cette carte sont comparées avec succès à une carte GTM ("Generative Topographic Mapping"). Le même auteur, dans (Haese 1999), emploie deux filtres de Kalman : un pour régler le coefficient d'apprentissage optimal et un pour estimer le rayon de voisinage optimal de la carte en cours d'apprentissage.

Une approche différente qui permet de se passer du réglage de ces paramètres, consiste à employer le GTM de (Bishop et al. 1998).

Les cartes auto-organisatrices peuvent utiliser un autre type de voisinage. Celui du "neural gas" de (Martinetz et Schulten 1991) connaît un certain succès dans ses applications. Cette dernière carte est intéressante dans le sens où elle utilise une définition du voisinage différente du voisinage conventionnel proposé par Kohonen. Le résultat est une meilleure quantification des espaces convexes et la suppression de tous les neurones superflus.

Les cartes SOM peuvent être employées avec des règles d'apprentissage différentes. Ainsi, dans (de Bodt et Cottrell 2000), une carte SOM avec une méthode d'apprentissage basée sur l'algorithme de "Bootstrapping" permet d'augmenter la probabilité de sélectionner deux prototypes voisins de la carte avec deux vecteurs voisins dans l'espace d'entrée. Les auteurs en déduisent une analyse statistique de la conservation de la topologie.

Dans (Euliano et Principe 1998), les auteurs ajoutent une plasticité temporelle à l'auto-organisation des cartes, en particulier au "neural gas" de (Martinetz et Schulten 1991) qui est appliqué à la prédiction d'une série temporelle aléatoire (celle de Mackey-Glass). Cette carte permet de tenir compte des variations des propriétés statistiques de l'environnement. Leur approche est comparée à l'approche de base sur le même problème (Martinetz et al. 1993). L'architecture est basée sur l'algorithme "Self-Organizing Temporal Pattern Recognizer" (SOTPAR, décrit dans (Murray 1993)) qui utilise trois principes fondamentaux : un apprentissage compétitif avec fonction de voisinage, la diffusion de l'activité neuronale à travers l'espace de sortie et une activation décroissante dans le temps. Le concept clé, qui constitue l'originalité du réseau, est celui de la diffusion de l'activité dans l'espace de sortie.

L'étude de l'intégration d'un facteur temps dans les cartes SOM n'est pas nouvelle. La technique principale permettant d'intégrer le facteur temps consiste à introduire des informations temporelles brutes directement par des méthodes de moyenne ou des lignes à retard, comme l'a fait l'équipe finlandaise de Kohonen (Kangas 1990; Kangas 1991).

Une alternative consiste à introduire des couches ou à développer des cartes SOM hiérarchiques dans lesquelles une seconde couche vise comme objectif de capturer les propriétés de vitesse des entrées (Kangas 1990; Kangas 1991). Une autre voie consiste à intégrer dans la carte SOM

une mémoire uniquement dédiée à cette tâche (Chappell et Taylor 1993; Euliano et Principe 1996). Des cartes SOM récursives ont également été développées dans (Varsta et al. 1997) et dans (Koskela et al. 1998). Les premières approches souffrent du fait que des performances correctes ne sont réellement atteintes que pour des réseaux de neurones dont la taille n'est plus raisonnable. Les secondes approches utilisent quant à elles un algorithme d'apprentissage ou une structure du réseau trop complexe. Comme alternative, Horio et Yamakawa (Horio et Yamakawa 2001) proposent une carte SOM qui utilise une boucle de retour d'information pour prendre en compte l'information temporelle. Une sortie est associée à chaque neurone, celle du vainqueur est mise à 1, toutes les autres à 0, et est utilisée en tant qu'entrée supplémentaire pour tous les neurones de la carte. Chaque neurone s'adapte donc en fonction de l'entrée présentée, ainsi qu'en fonction de l'histoire des neurones vainqueurs.

Il existe d'autres cartes SOM hiérarchiques (Luttrell 1989), des cartes SOM organisées en arbre (Chiueh et al. 1994), ou des SOM à plusieurs niveaux (Hecht-Nielsen 1987). Kohonen leur trouve plusieurs avantages (Kohonen 1995) : une recherche plus rapide du centre d'excitation, une convergence plus rapide, des capacités de généralisation améliorées et de nouvelles aptitudes d'abstraction.

Des cartes auto-organisatrices avec un nombre de neurones variables ont été élaborées dans (Martinetz et Schulten 1994), dans (Choi et Park 1994), dans (Bauer et Villmann 1997) ou encore dans (Fritzke 1994). Cette idée poursuit assez naturellement l'idée d'auto-organisation. L'aboutissement à des architectures adéquates se fait souvent avec un prix élevé en temps de calcul et avec un nombre important de données d'apprentissage.

Parmi les variantes des cartes SOM intéressantes, Kohonen (Kohonen 1995) note les "Adaptive-Subspace SOM" (ASSOM). L'approche modulaire de ce réseau permet à l'aide d'un module par région, de détecter les points d'une région de l'espace d'entrée qui sont sujets aux mêmes transformations simples (Kohonen et al. 1997).

Une carte auto-organisatrice à architecture évolutive est introduite dans (Choi et Park 1994) pour palier la nécessaire prédétermination de la taille de la carte. Une carte PSOM ("Parametrized SOM") avec un espace supplémentaire des paramètres est présentée dans (Walter et Ritter 1995). Il est impossible de citer toutes les extensions des cartes SOM, puisqu'elles touchent les architectures à grilles fixes ou évolutives, les approches hiérarchiques simples jusqu'aux arbres (Campos et Carpenter 2001), pour arriver finalement aux algorithmes de quantification vectorielle (VQ). Rose, Gurewitz et Fox (Rose et al. 1992) distinguent d'ailleurs trois types de cartes auto-organisatrices, le SOM, le "neural gas" et le "minimal face energy VQ". Les propriétés topologiques sont aussi des caractéristiques importantes d'une carte. On trouvera une dissertation intéressante sur la topologie des cartes auto-organisatrices et sur le choix d'une topologie adéquate par rapport à une distribution à apprendre dans (Utsugi 1996).

6 Le suivi de cible par asservissement visuel

Il y a un demi siècle environ, Wiener était déjà confronté au problème délicat de la poursuite de cibles. A l'époque, il était formulé dans le contexte dramatique de la seconde guerre mondiale, à savoir l'élaboration d'un appareil de visé et de suivi de trajectoire de missiles en vue de leur destruction. Il a alors proposé de prédire la trajectoire (Wiener 1961).

Aussi surprenant que cela puisse paraître, certains travaux en asservissement visuel concernant la poursuite de cible considèrent le mouvement de l'objet comme une perturbation négligeable (Papanikolopoulos et al. 1993), ou suppose connue la trajectoire de l'objet (Allen et al. 1993).

Le concept de prédire avant d'asservir est relativement simple. La boucle de retour, au lieu d'utiliser directement les informations du capteur, se sert des grandeurs prédites. Nous voyons

plusieurs motivations à l'utilisation d'un contrôle prédictif dans une boucle d'asservissement visuel. Ces motivations sont les suivantes :

- traiter les phénomènes d'occlusions (partielles mais également complètes),
- faciliter le traitement des images, c'est-à-dire l'extractions des caractéristiques visuelles,
- permettre de résoudre le problème d'appariement,
- compenser le temps de retard introduit par l'acquisition et le traitement des images,
- améliorer les performances de l'asservissement en cherchant à réduire l'erreur de traînage,
- compenser la dynamique du système face à celle de la commande ou face à l'évolution de l'environnement,
- prendre en compte et compenser le fait que l'environnement soit non stationnaire.

Les éléments de cette liste peuvent être satisfaits en mettant au point un modèle de connaissance pour déterminer la position de la cible, surtout lorsque les observations faites dans les images ne sont que partiellement disponibles, ou ne sont pas disponibles. Ce modèle de connaissance est réalisé par le KF. La méthode la plus utilisée pour estimer, prédire ou débruiter les primitives visuelles est le classique filtre de Kalman qui nécessite la connaissance du mouvement de la cible (modèle dynamique).

Le suivi d'une trajectoire par un retour d'informations visuelles est un thème important parmi les applications robotiques. Pour des asservissements visuels temps réel, de nombreuses approches de coordination main-œil ont été développées (Feddema et Lee 1990; Hashimoto 1993). La plupart de ces approches imposent des environnements simples et sans occlusions.

Il y a une dizaine d'années, a été introduit le suivi de trajectoires basé sur les modèles ("model-based motion tracking") (Harris 1992; Lowe 1992). Ces méthodes permettent de donner des indications sur ce qui est attendu dans les images et donc de simplifier les traitements pour extraire avec plus de facilité la position courante de l'objet dans les images (des segments de droites, des coins, des contours, etc.)

Dans (Kosaka et Nakazawa 1995), l'application décrite consiste à estimer la position et l'orientation d'un objet de forme connue avec un filtre de Kalman dans l'espace 3-D. L'approche reprend l'idée clé développée dans (Harris 1992) ou dans (Lowe 1992), et se distingue par l'utilisation d'une prédiction de la position et de l'orientation, ainsi que par l'introduction d'une grandeur pour qualifier l'incertitude des mouvements de l'objet. L'estimation qui est faite est relative à un repère associé à la caméra sachant que celle-ci se déplace. Ces travaux proposent une méthode pour traiter le problème délicat et inévitable en vision que sont les phénomènes d'occlusions.

Dans (Allen et al. 1993; Allen et al. 1991), le calcul de la position tridimensionnelle de l'objet est fait par une méthode de flot optique. Cette position sert d'entrée à un filtre $\alpha-\beta-\gamma$ qui effectue une nouvelle estimation de la pose du bras robotique pour suivre l'objet cible. Le fait que les paramètres du filtre de Kalman soient figés (ils sont prédéterminés et constants) permet surtout de réduire le coût calculatoire. Le système de vision est constitué de deux caméras calibrées et, la position, la vitesse et l'accélération de l'objet sont tous estimés en temps réel. La boucle de contrôle visuelle est validée par le suivi d'un train électrique avec un robot PUMA 560.

Dans (Lange et al. 1999) et dans (Lange et al. 1998), un suivi de profilé est présenté. Une caméra unique montée sur l'effecteur du robot permet de simplifier le problème de la commande par des contraintes géométriques. Il s'agit d'un asservissement visuel à deux dimensions, où la courbe à suivre est entièrement contenue dans un plan parallèle au plan image de la caméra. Le traitement d'images est extrêmement simple et autorise une réactualisation rapide des consignes. L'originalité de l'approche réside dans une mise en œuvre d'une stratégie de contrôle prédictif. Au principe du "feedforward control" est ajouté un module de "path planning". Ce dernier nécessite la connaissance des modèles (et donc que les éléments soient calibrés), le modèle de la caméra et le modèle

cinématique pour du robot, et implique que les modèles des trajectoires soient disponibles (ce qui est uniquement le cas dans les applications industrielles courantes). Les observations servent alors à déterminer les erreurs de suivi par rapport à la trajectoire idéale pour la corriger. Le contrôleur est conçu pour minimiser cette erreur, avec une prédiction des paramètres du mouvement. La position, mais aussi la vitesse et l'accélération de la trajectoire désirée sont prédites à l'instant k+1 avec le KF et le IKF.

Une estimation du mouvement 2-D et une estimation du mouvement 3-D d'un objet de la scène robotique sont présentées dans (Bensalah 1996) et dans (Bensalah et Chaumette 1994). Pour l'estimation du mouvement 2-D, le système de poursuite est basé sur trois estimateurs en parallèle, le filtre de Kalman, l'algorithme du maximum de vraisemblance généralisé (GRL, "Generalized Likelihood Ratio") pour détecter, estimer et compenser les mouvements brusques de l'objet, c'est-à-dire les changements de direction et d'amplitude, et un algorithme de détection des sauts en vitesse ou en accélération. La littérature propose une gamme riche d'algorithmes de détection de rupture de modèle (Basseville 1988). Bensalah a privilégié le GLR car il permet de détecter, d'estimer et de compenser les sauts au niveau du vecteur d'état. Concernant l'estimation des mouvements 3-D, un algorithme est proposé pour l'estimation à partir des observations de position et de vitesse faites dans les images avec l'hypothèse que l'objet se déplace à vitesse constante en module et en direction. C'est une estimation des mouvements 3-D à partir de données 2-D, et l'originalité réside dans la vision dynamique active adoptée qui permet de résoudre le problème classique du facteur d'échelle.

La grande majorité des approches s'appuie sur la connaissance des modèles des éléments qui interviennent dans la boucle d'asservissement. La commande et la prédiction sont alors basées sur les connaissances a priori.

Parmi les approches de contrôle prédictif qui n'utilisent ni les modèles des robots, ni celui du système de vision, citons les travaux de Piepmeier, McMurray, et Lipkin (Piepmeier et al. 1998a; Piepmeier et al. 1998b). La commande est basée sur l'inverse de la matrice Jacobienne d'interaction qui est approchée par la méthode d'estimation de Broyden (comme le fait Jägersand (Jägersand 1996)), et une étape de prédiction cherche à compenser le temps de retard des informations visuelles disponibles dû à l'acquisition et au traitement des images. Dans leurs travaux, les auteurs comparent trois méthodes, un prédicteur du premier ordre, un filtre $\alpha-\beta$ et un filtre de Kalman. Le premier, extrêmement simple, n'intègre pas de modèle de bruit. Les deux autres estimateurs utilisent un modèle de caméra, et l'accélération de l'objet est supposée nulle. Pour le filtre de Kalman, un modèle de bruit de mesure qui permet de déterminer $\mathbf{R}_k$ analytiquement est proposé. La covariance du bruit du processus, $\mathbf{Q}_k$, est déterminée par une méthode de Monte Carlo. Les simulations proposées montrent que le KF permet d'augmenter les performances de la boucle de retour visuelle pour une opération de poursuite de cible mobile par asservissement visuel.

Si, dans le domaine de l'asservissement visuel, l'essentiel des méthodes de prédiction se fait par le KF, d'autres types de commande prédictive apparaissent, comme par exemple, la commande GPC ("Generalized predictive control") qui a été introduite en 1987 par Clarke, Mohtadi et Tuffs (Clarke et al. 1987). La commande GPC peut être définie comme une version simplifiée du correcteur LQG ("Linear Quadratic Gaussian"). Tous deux minimisent une fonction de coût dépendant de l'intégrale de l'erreur quadratique et de l'énergie de la commande. La commande GPC est présentée en détail dans (Astrom et Wittenmark 1995, page 174) où elle figure dans la classification des méthodes adaptatives de contrôle (page 168).

La commande GPC est utilisée pour le suivi de cible par asservissement visuel dans (Gangloff et al. 1999). Ce travail présente une tâche de suivi de profilé avec prédiction de la trajectoire. Le modèle profilé est tridimensionnel, contrairement à l'approche présentée dans (Lange et al. 1999) et dans (Lange et al. 1998). Lorsqu'au moins trois courbes parallèles distinctes sont visibles dans l'image, alors les paramètres opérationnels (3 coordonnées et 3 angles) du modèle profilé peuvent être estimés.

Slotine a également proposé une approche de suivi de cible par l'asservissement visuel en vue d'une opération de saisie au vol d'une balle avec un bras muni d'une pince. Le système de vision en couleur est associé à un traitement d'images relativement simple. Il est déporté et se compose de deux caméras avec deux degrés de liberté chacune. Encore une fois, les auteurs cherchent à compenser le temps de retard des informations visuelles dû à l'acquisition et au traitement des images. Les modèles sont nécessaires dans cette approche. On peut d'ailleurs remarquer que les caméras et le bras manipulateur possèdent des systèmes de coordonnées distincts. L'élaboration de la commande est basée sur l'estimation de la transformation pour passer de l'un à l'autre ; le système doit être calibré. Pour l'élaboration des commandes, quelques simplifications sont apportées par des contraintes géométriques dans une procédure de "cross calibration" (un algorithme du type LMS). Le contrôle robotique est adaptatif (Niemeyer et Slotine 1991) et la méthode de prédiction est basée sur une représentation d'état différente du EKF ; elle est appelée "sliding-mode method" et est décrite dans (Walcott et Zak 1987) et dans (Yaz et Azemi 1993). Cette méthode estime la vitesse et l'accélération de l'objet pour prédire sa future position et sa future vitesse. Dans le modèle de la trajectoire de l'objet, un polynôme d'ordre 3, l'accélération est supposée constante.

Dans (Marchand et al. 2001), Chaumette ne réalise pas de suivi 2-D avec une méthode de prédiction basée sur le filtrage de Kalman. Le modèle utilisé par le KF est trop contraignant et ses motivations sont les suivantes :

- la détermination du modèle constitue un inconvénient majeur,
- l'initialisation des matrices de covariance est une phase délicate et problématique,
- la modélisation par une représentation d'état contraint la vitesse à être constante (à cause d'une matrice de transition constante).

Pour le suivi 2-D, il préfère donc au filtre de Kalman un estimateur basé sur une transformation affine, et pour le suivi 3-D, il utilise un modèle CAD ("Computer Aided Design", un modèle 3-D projeté dans un espace 2-D). Il continue à associer estimation 2-D et estimation 3-D. Cette technique semble robuste du fait que chaque estimation peut servir l'autre, puisque c'est le même mouvement (la même trajectoire) qui est estimé dans deux espaces différents. La méthode est donc composée de deux étapes. La première se sert de deux images successives pour construire un modèle de mouvement affine 2-D. C'est une méthode statistique qui a été choisie à cause de sa robustesse. Elle est décrite dans (Bouthemy 1989). La seconde étape emploie un algorithme de calcul et d'estimation de pose en projetant le modèle 3-D sur les plans images. Ceci nécessite la pleine connaissance de la transformation entre l'espace tridimensionnel et l'espace de projection, autrement dit des caractéristiques visuelles. Pour cette dernière étape, un modèle polyédral est suffisant. L'inconvénient que les auteurs voient à leur approche est la nécessité de parfaitement connaître le modèle de l'objet suivi (un cube, une voiture, etc.) Cet algorithme de poursuite permet toutefois de réaliser une tâche d'asservissement visuel efficace et robuste avec des objets réels.

A partir d'images plus complexe, Deguchi propose une méthode d'interprétation directe des images (Deguchi 2000; Deguchi et Takahashi 1999b; Deguchi et Yonaiyama 2000) ou utilise la notion d'espace propre (Deguchi et Noguchi 1996; Deguchi et Takahashi 1999a) initialement mise au point par Nayar, Nene et Murase (Murase et Nayar 1994; Nayar et al. 1996). Dans ce dernier cas, l'idée consiste à utiliser la luminosité plutôt que d'extraire des caractéristiques visuelles des images. L'approche ne dépend plus de la forme des objets, de leur contours, ni même de leur propriétés géométriques. Les images sont projetées dans un sous-espace (appelé espace propre) qui est de taille moindre car seules les dimensions les plus pertinentes pour caractériser un objet sont conservées. Cette technique permet ainsi une approximation suffisante de la corrélation entre deux images. Elle est basée sur un apprentissage fondé sur le principe de l'analyse en composante principale. L'algorithme en lui-même est détaillé dans (Murakami et Kumar 1982). On peut lui reprocher un coût calculatoire important. L'approche est originale dans le sens où elle se base sur des images brutes, sans passer par

des caractéristiques visuelles. Elle requiert en contrepartie une maîtrise parfaite de la structure de l'éclaire de la scène.

Il existe également des méthodes extrêmement simples, voire même empiriques pour estimer les mouvements d'une cible. Dans (Feddema et Lee 1990), Feddema et Lee introduisent dans la boucle d'asservissement un générateur de trajectoires des primitives visuelles ("Feature-Based Trajectory generator"). Pour cela, une technique d'estimation de la vitesse maximale et de la vélocité minimale est employée.

Des approches neuronales de prédiction ont été développées avec le même objectif, à savoir la compensation des mouvements de l'objet cible. Miller (Miller 1989) fait de la prédiction avec un CMAC ("Cerebellar Model Articulation Controller ") et effectue l'apprentissage de la transformation sensori-motrice avec un autre CMAC. Il est obligé d'apprendre le modèle direct et le modèle inverse.

7 Conclusion

Dans le cadre de l'asservissement visuel, une loi de commande appliquée à un système robotique concrétise le positionnement et l'orientation de son effecteur dans l'environnement sous le contrôle de la perception visuelle binoculaire. Cette loi repose sur la formalisation de la coordination du système robotique et du système de vision, autrement dit sur la connaissance de la relation entre l'espace de perception du capteur et l'espace des commandes du système robotique.

La relation entre les caractéristiques visuelles et les consignes articulaires est appelée transformation sensori-motrice. Cette transformation, sur laquelle est fondée la loi de commande, nécessite la connaissance d'un certain nombre de modèles. Elle induit la modélisation paramétrique du système de vision et du système robotique, ainsi que celle des interactions spatiales liant ces deux systèmes. La détermination de ces modèles repose sur l'étalonnage des paramètres de modèles idéaux et préétablis des éléments du système de vision et du système robotique.

L'étalonnage est un problème complexe, et, le processus en lui même est contraignant par ses modalités opératoires. Une source de problèmes supplémentaires provient de l'idéalisation et de la simplification des modèles. La précision d'un modèle dépend de sa complexité. Une dernière limitation de l'approche modèle vient du fait que tout changement d'un système par un autre requiert une nouvelle modélisation et un nouvel étalonnage. Il en va de même pour tout déplacement d'un élément par rapport à l'autre ; il doit explicitement être référencé dans la transformation pour se voir pris en compte dans la commande. L'approche modèle n'est de ce fait pas générique, elle est fortement tributaire du système auquel elle s'applique.

Par opposition au caractère déterministe de l'approche modèle, nous préconisons une méthode adaptative ou par apprentissage. De manière générale, une approche neuronale réalise une approximation d'un système par un processus d'apprentissage. Appliquée à l'asservissement visuel, l'approche neuronale permet l'identification fonctionnelle de la transformation sensori-motrice en construisant un modèle de connaissance à partir d'exemples du comportement du système robotique.

Si l'apprentissage est permanent, toute évaluation du système engendre une mise à jour automatique de l'identification et par conséquent de l'adéquation dynamique entre le modèle de comportement estimé et le système robot-vision.

L'approche neuromimétique du contrôle robotique devient une alternative à l'approche modèle pour surpasser ses limitations et pour effectuer une commande sans a priori. On parle d'asservissement visuel adaptatif ou d'asservissement visuel par apprentissage.

Les cartes auto-organisatrices de Kohonen sont introduites constitue une solution efficace dans la réalisation de contrôleurs robotiques. Elles sont intéressantes pour ce genre d'application parce qu'elles conservent la topologie de l'espace d'entrée. Deux points proches dans la distribution d'entrée vont être associés à deux neurones (prototypes) proches dans la grille de la carte. Il est ainsi possible de tenir compte des propriétés géométriques d'un système dans l'élaboration des consignes en vue de le

commander. Ceci est d'autant plus intéressant que le système est non linéaire. Cette propriété rend les cartes auto-organisatrices d'ailleurs attrayantes pour bien d'autres applications.

Les cartes de Kohonen ne pouvant pas être appliquées sans quelques modifications au problème de la commande robotique, des extensions ont été mises au point, une carte étendue (ESOM) qui permet d'apprendre la transformation entre deux espace, et une carte à sortie supervisée (SSOM) dont l'architecture est parfaitement adaptée à l'approximation de matrices Jacobiennes.

Chapitre 2

Statistiques et théories de l'apprentissage

1 Introduction

Un réseau de neurones, lorsqu'il apprend, ajuste ses poids synaptiques afin d'aboutir à une description d'un système (d'une fonction, d'une tâche…) qu'il utilise pour relier ses entrées et ses sorties (on parle souvent de "mapping" dans la littérature anglophone). En cela, les ANN ont de fortes ressemblances avec les techniques statistiques.

Il est souvent fort intéressant de comparer les formalismes et les terminologies à travers les disciplines. Là où le traitement du signal cherche à minimiser des fonctions de corrélation, la physique et les mathématiques cherchent à minimiser des fonctions de coût ou d'énergie. Là où le traitement du signal analyse la convergence d'un système, là où les concepteurs de réseaux de neurones synthétisent la convergence d'un apprentissage, la physique et les mathématiques étudient le phénomène de relaxation. Dans tous les cas, les problématiques sont les mêmes. On peut donc être en droit de penser que les solutions, elles aussi sont les mêmes.

Depuis de nombreuses années, les mathématiques et la physique ont apporté toute une classe de problèmes liés à l'optimisation, ainsi que des méthodes d'approche de ces problèmes. C'est pourquoi nous allons étudier ce qui relie, mais aussi ce qui distingue, la théorie des statistiques de celle de l'apprentissage.

Dans de nombreuses applications, les réseaux de neurones se posent évidemment comme une alternative aux techniques statistiques. Voyons les choses autrement. Ne seraient-ils pas plutôt une manière différente et nouvelle de mettre en œuvre les statistiques ? Un réseau de neurones, et l'apprentissage qui lui est associé, peuvent être caractérisés à l'aide des statistiques. Une règle d'apprentissage peut être modélisée grâce aux outils statistiques. Un réseau qui a appris peut être compris et expliqué à travers des grandeurs statistiques.

Dans ce chapitre, une première partie permet d'introduire un certain nombre de généralités. Les différences et les points communs entre méthodes statistiques et méthodes par apprentissage, comment des phénomènes stochastiques se retrouvent dans les réseaux de neurones, et comment les statistiques peuvent aider à la réalisation d'un apprentissage optimal. Nous tenterons ensuite d'analyser la nature statistique de l'apprentissage. Partant de l'analyse du biais et de la variance d'un réseau, la théorie de l'apprentissage analyse la convergence d'un apprentissage en évaluant un risque empirique. Les théories les plus abouties sont les contributions de Vapnik (Vapnik 1995). Elles tentent de confronter la complexité structurelle d'un réseau à sa capacité d'apprentissage. L'approche Bayésienne sera ensuite brièvement présentée. Cette technique classique servira à mettre en avant les points communs et les différences des statistiques avec les ANN. Nous verrons que les processus de Markov

sont des outils statistiques qui permettent de modéliser un processus d'apprentissage avant de dresser un bilan des avantages de l'une comme de l'autre des deux approches pour constituer un modèle de connaissance à partir de données représentatives d'un problème.

La seconde partie de ce chapitre se concentre sur une méthode statistique particulière, le filtrage de Kalman. Cet estimateur Bayésien est un filtre linéaire à variance minimale ayant une formulation récursive. Comme toutes les méthodes Bayésiennes, il nécessite une connaissance a priori, en l'occurrence un modèle d'évolution. Le filtre de Kalman, de même que la théorie Bayésienne, utilise l'information contenue dans les mesures mais également une connaissance a priori de la valeur à estimer. Cette connaissance est modélisée à l'aide d'une densité de probabilité qui permet d'avoir une idée subjective de la valeur à estimer. C'est la traduction d'une connaissance a priori dans la théorie Bayésienne. Nous détaillerons les avantages et les inconvénients de ce filtre, avant d'exposer le filtrage de Kalman adaptatif. Nous établirons un état de l'art des différentes variantes adaptatives existantes. Nous présenterons enfin le filtre de Kalman en tant que nouvelle manière d'ajuster les poids synaptiques d'un ANN. Ces capacités à tenir compte à la fois de moments statistiques, et de grandeurs instantanées, lui confère une certaine robustesse dans la recherche d'un minimum global d'une surface d'erreur, et permet de ce fait de contourner les minima locaux.

2 Incertitudes, probabilités et apprentissage

2.1 Considérations générales

On rencontre en pratique des phénomènes qui, reproduits de façon apparemment identique un grand nombre de fois, donnent à chaque expérience des résultats qui semblent imprévisibles. On dit que de tel phénomènes sont aléatoires.

Ces phénomènes aléatoires, ou stochastiques, sont souvent omis à tort. Dans certaines situations, il se révèle indispensable de considérer le caractère aléatoire de certaines grandeurs. Le stochastique est par exemple au centre de la problématique de l'automaticien. Sans perturbation, la rétroaction n'aurait aucune raison d'être et l'essentiel de la discipline laisserait place à un univers conceptuel en boucle ouverte. Les acquis historiques sur la stabilité sont précieux, mais n'abordent pas la permanence des perturbations qui demande un cadre beaucoup plus large, tant en commande qu'en estimation. L'identification et l'adaptatif s'associent depuis longtemps au stochastique de façon naturelle à travers les principes Bayésiens et la vision académique générale du domaine s'est élargie et éclaircie au contact des mathématiciens.

Les mathématiques, et la théorie des statistiques en particulier, offrent un outil théorique, particulièrement bien adapté aux calculs portant sur les grandeurs stochastiques à travers les signaux aléatoires stationnaires du second ordre. Les probabilités conditionnelles et les densités de probabilité, en sont une forme plus manipulable. Des sciences plus proches de l'application vont donc naturellement les utiliser. On compte parmi elles, le traitement du signal ainsi que la science de l'apprentissage.

2.1.1 Objectifs identiques et moyens quasi-semblables

Les problèmes rencontrés dans la théorie de l'apprentissage sont les mêmes que ceux rencontrés en statistiques et les liens entre ces deux disciplines sont donc forts. Les points communs entre les ANN et les méthodes statistiques sont la taille des bases d'exemples, la représentativité des données, la signification, ainsi que la représentation des résultats et des estimations.

Les statistiques permettent de caractériser et de déterminer une distribution grâce à des paramètres ayant une représentation physique significative (les moments, les fonctions caractéristiques, les cumulants, etc.) alors que les réseaux de neurones cherchent à identifier une

distribution par rapport à leurs poids synaptiques, qui sont des grandeurs généralement hermétiques à toute interprétation.

Dans les statistiques, l'incertitude est modélisée par des distributions de probabilité et les inférences sont réalisées en construisant des probabilités conditionnelles a posteriori pour les variables non observables, à partir des variables accessibles. L'incertitude est rarement modélisée par les approches neuronales.

Une difficulté propre aux approches neuronales est la gestion de la complexité du modèle neuronal qui dépend du nombre d'exemples contenus dans le corpus d'apprentissage, de la quantité de bruit qui interfère, et de la complexité de la fonction à apprendre. Dans l'approche Bayésienne, ces questions sont traitées de manière inhérente et cohérente en définissant des connaissances a priori sur les hyperparamètres qui définissent le modèle. La complexité du modèle final tient compte de l'ensemble des données représentatives de la fonction à estimer. Il peut même supporter des complexités différentes, en différents endroits du modèle, en regroupant les paramètres d'importance identique et en les intervertissant.

Les performances des approches statistiques sont tributaires de la précision des informations a priori suggérées. Ces performances sont plus facilement interprétables avec les approches statistiques qu'avec les approches neuronales. La qualité d'une estimation se lit naturellement avec une densité de probabilité conditionnelle, ou avec une probabilité conditionnelle, sachant tous les exemples du corpus. Pour les ANN, les méthodes et les grandeurs représentatives d'un apprentissage correct se mettent seulement en place. Elles dépendent souvent de la structure et de la taille du réseau employé pour l'approximation.

Hecht-Nielsen (Hecht-Nielsen 1990) a déjà étudié la ressemblance entre identification par apprentissage et régression statistique. Selon lui, la façon qu'ont les réseaux de neurones d'approximer des fonctions peut être vue comme une généralisation de la régression statistique. Dans le cadre de la régression, la forme de la fonction à approximer doit dans un premier temps être choisie, puis elle est ajustée ("fitted") à l'aide d'un critère d'erreur. Comme exemple, citons les techniques d'estimation polynomiales, l'analyse de Fourier, ou le filtrage de Kalman. Les réseaux neuromimétiques vont au-delà de la régression. Leur principale avantage, par rapport aux techniques de régression statistique est leurs formes fonctionnelles plus générales que celles des techniques de régression statistique. Les réseaux neuromimétiques ne dépendent pas d'une superposition linéaire ou de propriété d'orthogonalité alors que les techniques statistiques de régression utilisent des outils mathématiques qui ne peuvent s'en passer. En clair, l'approche statistique requiert des informations a priori alors que les réseaux de neurones travaillent sans connaissance préalable. C'est en cela que les ANN vont plus loin que les statistiques et peuvent dans certaines situations être plus intéressants. L'approche Bayésienne est en cela très démonstrative (Lampinen et Vehtari 2001).

Les réseaux neuromimétiques comparés aux statistiques appliqués à des tâches d'approximation de fonction sont plus ouverts et plus souples, dans la mesure où ils sont tirés de sources d'inspirations multiples : le traitement du signal, le traitement des images, les sciences cognitives, l'automatique, la théorie du contrôle, mais également les neurosciences.

2.1.2 Les probabilités dans les réseaux de neurones

Il est difficile de modéliser l'incertitude perçue d'un système à l'aide des ANN. Cela peut paraître paradoxal, mais il existe cependant un caractère aléatoire inhérent à toute architecture neuronale. L'aléatoire peut se situer à plusieurs niveaux dans un réseau ; il peut s'agir des entrées, des connexions, ou encore de la dynamique des unités neuronales.

- Les entrées : il est souvent très commode de considérer les entrées d'un système (les stimulations, les formes à reconnaître, les objets à classifier, les prototypes à mémoriser) comme des objets aléatoires, composés d'une partie déterministe et

d'un terme aléatoire représentant tout ce qui n'est pas calculable, tout ce qu'il est difficile de prendre en compte de manière identifiable ; l'aléatoire correspond alors au bruit, aux erreurs de mesure, à l'aléas des stimulations sensorielles…

- Les connexions : on peut choisir des connexions purement aléatoires, ou bien des connexions calculées à partir d'entrées aléatoires au cours d'un apprentissage ;
- La dynamique : la règle de transformation de l'état de chaque unité du réseau peut être probabiliste, de manière à intégrer les fluctuations individuelles, les incertitudes, par analogie aux systèmes physiques où l'on considère le bruit ou la température.

Par conséquent, beaucoup de réseaux de neurones, définis au moyen de modèles déterministes, peuvent aussi l'être au moyen de modèles probabilistes. On peut même affirmer qu'aujourd'hui, les réseaux de neurones, et le processus d'apprentissage qui leur est associé, ne peuvent être dissociés de la théorie des statistiques.

2.1.3 Les statistiques pour améliorer l'apprentissage

Apprendre n'est pas une fin en soi. Un apprentissage est toujours réalisé avec un objectif, celui d'approximer, d'identifier, d'estimer de classer, etc. On cherche donc à obtenir le meilleur apprentissage possible, et pour cela, il faut pouvoir caractériser un processus d'apprentissage et analyser sa convergence. Il existe très peu d'outils suffisamment génériques pour surmonter tout ce qui distingue les différents réseaux neuromimétiques. Chaque réseau est différent, par les neurones, par l'architecture, par la règle d'apprentissage, ou encore par sa source d'inspiration. Notons que les réseaux de neurones sont nouveaux et que très peu trouvent des justifications mathématiques quant à la convergence de leur apprentissage.

Les statistiques sont génériques et constituent alors un outil puissant pour analyser et comprendre un apprentissage. Les méthodes statistiques qui permettent d'analyser une machine d'apprentissage sont regroupées sous la désignation de théorie de l'apprentissage.

A ce jour, la théorie des statistiques permet :

- d'analyser les performances d'un apprentissage,
- de modéliser un processus ou une règle d'apprentissage,
- d'optimiser l'apprentissage d'un réseau.

Les capacités de généralisation peuvent être interprétées à partir d'un point de vue statistique, grâce au biais et à la variance introduits précédemment par les équations (6.15) et (6.16). Le processus d'apprentissage peut lui-même être associé à un modèle statistique, les chaînes de Markov en l'occurrence. Nous verrons brièvement dans quelle mesure cela est possible dans la suite de cette discussion. Modéliser une règle d'apprentissage permet de mieux la comprendre, mais surtout, et c'est là son intérêt principal, permet de démontrer sa convergence dans le respect de certaines conditions.

Concernant la règle d'apprentissage, l'optimisation des poids d'un réseau s'est élargie aux méthodes statistiques. Ces dernières présentent des apparences fort séduisantes pour guider l'apprentissage de plusieurs types de réseaux puisqu'elles permettent d'éviter de tomber dans les minima locaux disposés entre le points de départ et le minimum global d'une surface d'erreur. L'emploi de l'algorithme EM ("Expectation Maximization"') ou du KF (filtre de Kalman) semble avoir montré une certaine efficacité (De Nicolao et Ferrari-Trecate 2001).

La théorie des groupes que manipulent les statistiques permet de ne pas tenir compte seulement d'une information ponctuelle (la dernière information dans l'espace temporel), mais aussi de grandeurs passées reflétant un comportement plus général. Toutes les méthodes basées sur le calcul d'un gradient instantané sont des méthodes sans mémoire, capables de ne tenir compte que de l'information courante. Les méthodes statistiques, si elles convergent souvent de façon robuste vers la

solution idéale, possèdent en revanche des contre-parties. Elles sont notamment plus lentes à converger.

Les méthodes statistiques peuvent également être utilisées pour régler les paramètres structurels des réseaux, dans les approches à modèle multiple pour la recherche de la combinaison optimale (au sens statistiques) des réponses de plusieurs réseaux, pour choisir un réseau parmi plusieurs, pour guider le choix du nombre de modules, etc. Des références bibliographiques sont données et discutées dans le Chapitre 3, dans la partie traitant des approches modulaires des réseaux de neurones.

2.2 La nature statistique de l'apprentissage

2.2.1 Apprentissage supervisé

Un réseau neuronal peut être analysé du point de vue statistique en termes de convergence et de performance de son apprentissage. Les développements qui vont suivre sont fait dans le cadre de l'apprentissage supervisé.

Le problème de l'apprentissage supervisé est celui de la sélection d'une fonction particulière $F(\mathbf{x},\mathbf{w})$ qui approxime dans un sens optimal la sortie désirée d. L'optimalité est ici définie au sens statistique et par rapport aux exemples de la base d'apprentissage $\mathcal{T}$, auparavant définie par l'expression (6.1).

On parle d'apprentissage stochastique dans la mesure où les exemples, c'est-à-dire les paires $(\mathbf{x}_i, d_i)$, sont tirées aléatoirement de $\mathcal{T}$ pour être présentées au réseau de neurones. A chaque paire est associée une fonction cumulative de probabilité conjointe $F_{\mathbf{X},D}(\mathbf{x},d)$.

L'apprentissage est guidé par une mesure de perte ou de divergence $L\big(d, F(\mathbf{x},\mathbf{w})\big)$. Cette grandeur scalaire permet de mesurer le coût de confondre la réponse désirée d correspondant à une entrée $\mathbf{x}$, avec la réponse estimée $F(\mathbf{x},\mathbf{w})$. Elle est généralement une mesure de la distance Euclidienne quadratique entre $d = f(\mathbf{x})$ et $F(\mathbf{x},\mathbf{w})$:

$$L\big(d, F(\mathbf{x},\mathbf{w})\big) = \big(d - F(\mathbf{x},\mathbf{w})\big)^2 . \tag{2.1}$$

On définit alors une fonction d'erreur moyenne également appelée risque fonctionnel ou coût moyen par :

$$R(\mathbf{w}) = \int L\big(d, F(\mathbf{x},\mathbf{w})\big)\, dF_{\mathbf{X},D}(\mathbf{x},d) . \tag{2.2}$$

L'objectif de l'apprentissage supervisé revient alors à minimiser le risque donné par (2.2). Ce risque permet de choisir parmi l'ensemble des fonctions $\{F(\mathbf{x},\mathbf{w}), \mathbf{w} \in \mathcal{W}\}$, la fonction solution $F(\mathbf{x},\mathbf{w}^*)$ pour laquelle le risque est minimum. L'optimisation de l'apprentissage se formule donc par :

$$F(\mathbf{x},\mathbf{w}^*) = \underset{F(\mathbf{x},\mathbf{w})}{\arg\min}\, R(\mathbf{w}) . \tag{2.3}$$

Déterminer $F(\mathbf{x},\mathbf{w}^*)$ est difficile car cela nécessite la pleine connaissance de la fonction cumulative de probabilité conjointe $F_{\mathbf{X},D}(\mathbf{x},d)$ qui est une fonction très souvent inconnue. Les seules informations disponibles le sont sous la forme des exemples que constitue $\mathcal{T}$. L'introduction d'un risque empirique permet de parer à ce problème (Vapnik 1995). Ce risque empirique, qui se calcule plus simplement que le risque moyen $R(\mathbf{w})$, le remplace dans l'optimisation (2.3) de l'apprentissage.

Le risque fonctionnel empirique est ainsi défini :

$$R_{\text{emp}}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} L\big(d_i, F(\mathbf{x}_i,\mathbf{w})\big) . \tag{2.4}$$

Il existe deux avantages à utiliser le risque empirique $R_{\text{emp}}(\mathbf{w})$ plutôt que le risque moyen $R(\mathbf{w})$:

- le risque empirique ne dépend pas de la fonction cumulative de probabilité conjointe $F_{\mathbf{X},D}(\mathbf{x},d)$,

- et il peut en théorie être minimisé par rapport au vecteur $\mathbf{w}$ des poids synaptiques.

Le critère donné par (2.4) fait uniquement intervenir les connaissances contenues dans la base d'apprentissage $\mathcal{T}$. Lorsque la taille de la base d'apprentissage est infiniment grande, et sous la condition d'une convergence uniforme du risque empirique $R_{\text{emp}}(\mathbf{w})$ vers le risque moyen $R(\mathbf{w})$, alors la solution empirique converge en probabilités vers la valeur théorique $F(\mathbf{x},\mathbf{w}^*)$.

L'étude de la convergence du risque empirique $R_{\text{emp}}(\mathbf{w})$ vers le risque moyen $R(\mathbf{w})$ est construite sur un paramètre important de la théorie statistique de l'apprentissage : la dimension de Vapnik-Chervonenkis. Ce paramètre, plus simplement appelé dimension VC, est une mesure de la capacité d'une classe de fonctions réalisées par le réseau de neurones. La dimension VC est un concept combinatoire qui permet l'étude de la stabilité d'un réseau.

La complexité est en effet déterminante dans les résultats que l'on peut espérer d'un système d'apprentissage. Si la classe des fonctions à l'intérieur de laquelle on choisit le système d'apprentissage est très riche (dimension VC élevée), on pourra généralement choisir une fonction qui s'adapte parfaitement aux exemples mais n'offre pas de garantie en généralisation. En revanche, si une fonction appartenant à une classe de fonctions simples s'adapte convenablement aux exemples d'apprentissage, on peut considérer que ce modèle, qui a mis en évidence une structure interne de ces exemples, est généralisable.

Nous n'allons pas détailler plus la théorie statistique de l'apprentissage. Des synthèses plus ou moins détaillées se trouvent dans (Haykin 1994; Jodouin 1994b; Vapnik 1995). En fait, la théorie de Vapnik permet de définir dans certains cas la taille d'un réseau part rapport à la complexité d'un problème (pour éviter des structures neuronales aux tailles trop élevées, on parle de la minimisation du risque structurel) ; elle permet d'éviter les phénomènes de sur-apprentissage en considérant la quantité d'exemples que constitue la base d'apprentissage, et permet surtout une mesure de la stabilité (donc de la convergence) de l'apprentissage. On trouvera une application convaincante dans (Lengellé et Schaltenbrand 1997). La détermination de l'architecture de réseaux multicouches est appliquée à un problème de discrimination, l'analyse automatique du sommeil. Les auteurs utilisent l'algorithme de "cascade-correlation".

La mise en oeuvre des idées de Vapnik est également illustrée dans (Bottou 1997). Pour une palette plus large d'applications, on se référera au numéro spécial (IEEE 1999). Précisons pour terminer que la convergence uniforme en probabilités des réseaux multicouche est synthétisée dans (Gascuel 1997).

2.2.2 Apprentissage non supervisé

Lorsqu'aucune sortie désirée n'est disponible, la vocation de l'apprentissage non supervisé est d'élaborer une représentation interne de la connaissance du système grâce aux paramètres internes du réseau. Cette élaboration n'est guidée que par les stimuli présentés au réseau. La qualité de cette représentation peut être caractérisée par une mesure de divergence $L\big(F(\mathbf{x},\mathbf{w})\big)$ qui ne dépend que des réponses du réseau.

Le risque fonctionnel empirique se définit alors par :

$$R_{\text{emp}}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} L\big(F(\mathbf{x}_i,\mathbf{w})\big). \tag{2.5}$$

Ce formalisme permet de trouver une solution en minimisant le critère du risque empirique.

2.3 L'approche Bayésienne

Dans l'approche Bayésienne, on considère la fonction de représentation comme étant une variable aléatoire $\mathbf{X}$ tiré d'un espace $\mathbb{R}^n$. Chaque classe C_i est représentée par une densité de probabilité conditionnelle, une fonction $f_{\mathbf{x}}(\mathbf{x}|C_i)$. La décision de prendre une valeur $\mathbf{x}$ de $\mathbf{X}$ consiste

à lui assigner la classe qui maximise la probabilité $p(C_i|\mathbf{x})$, autrement dit, en lui appliquant la formule de Bayes, la classe qui maximise la quantité $f_{\mathbf{x}}(\mathbf{x}|C_i)p(C_i) = f_{\mathbf{x}}(\mathbf{x}|C_i)p_i$.

On peut aisément se rendre compte que l'approche Bayésienne, de part sa nature, est propice aux tâches de classification.

2.3.1 La classification Bayésienne

Mettre au point un classificateur Bayésien revient à minimiser un risque moyen qui prend en compte le coût des différentes erreurs de classification. Nous allons développer l'approche Bayésienne pour un problème de classification avec deux classes C_1 et C_2. Le risque moyen $\mathcal{R}$ est alors défini par :

$$\begin{aligned}\mathcal{R} = c_{11}p_1 \int_{S_1} f_{\mathbf{x}}\left(\mathbf{x}|C_1\right)d\mathbf{x} + c_{12}p_2 \int_{S_1} f_{\mathbf{x}}\left(\mathbf{x}|C_2\right)d\mathbf{x} \\ + c_{21}p_1 \int_{S_2} f_{\mathbf{x}}\left(\mathbf{x}|C_1\right)d\mathbf{x} + c_{22}p_2 \int_{S_2} f_{\mathbf{x}}\left(\mathbf{x}|C_2\right)d\mathbf{x}\end{aligned} \tag{2.6}$$

où :

- p_i représente la probabilité a priori que le vecteur d'observation $\mathbf{x}$, représentant une réalisation de la variable aléatoire vectorielle $\mathbf{X}$, soit tiré du sous-espace S_i, avec $i = 1,2$ et $p_1 + p_2 = 1$;
- c_{ij} désigne un coefficient de coût de décision en faveur de la classe C_i représentée par le sous-espace S_i, avec $(i,j) = 1,2$;
- $f_{\mathbf{x}}(\mathbf{x}|C_i)$ est la fonction de densité de probabilité du vecteur aléatoire $\mathbf{X}$, pour un vecteur d'observation $\mathbf{x}$ tiré du sous-espace S_i (ou densité conditionnelle de l'observation $\mathbf{x}$ sachant que la classe C_i est vraie).

Les termes dans l'équation (2.6) pondérés par c_{ii} représentent des décisions correctes de classification. Dans cette même équation, les termes pondérés par les coefficients c_{ij}, avec $i \neq j$, représentent des cas de mauvaises classifications. Chaque décision est en fait pondérée par un produit de deux termes : le coefficient impliqué dans la prise de décision et la probabilité a priori.

Si les valeurs p_i sont en général connues et fixées à l'avance, il en va différemment pour la densité conditionnelle assignée à chaque classe C_i.

a Cas général

Soit S l'espace total des observations : $S = S_1 + S_2$. L'ensemble des sous-espaces S_i forme une partition de l'espace total d'observation S. Le vecteur d'observation doit être classé sur l'ensemble de l'espace d'observation, ainsi le risque $\mathcal{R}$ peut aussi se réécrire en faisant apparaître dans les domaines de décision un terme constant :

$$\begin{aligned}\mathcal{R} = c_{11}p_1 \int_{S_1} f_{\mathbf{x}}\left(\mathbf{x}|C_1\right)d\mathbf{x} + c_{12}p_2 \int_{S_1} f_{\mathbf{x}}\left(\mathbf{x}|C_2\right)d\mathbf{x} \\ + c_{21}p_1 \int_{S-S_1} f_{\mathbf{x}}\left(\mathbf{x}|C_1\right)d\mathbf{x} + c_{22}p_2 \int_{S-S_1} f_{\mathbf{x}}\left(\mathbf{x}|C_2\right)d\mathbf{x}\end{aligned} \tag{2.7}$$

Sachant que $c_{11} < c_{21}$, $c_{22} < c_{12}$ et que

$$\int_S f_{\mathbf{x}}\left(\mathbf{x}|C_1\right)d\mathbf{x} = \int_S f_{\mathbf{x}}\left(\mathbf{x}|C_2\right)d\mathbf{x} = 1, \tag{2.8}$$

l'expression du risque peut se simplifier en :

$$\mathcal{R} = c_{21}p_1 + c_{22}p_2 + \int_{S_1}\left((c_{12} - c_{22})p_2 f_{\mathbf{x}}\left(\mathbf{x}|C_2\right) - (c_{21} - c_{11})p_1 f_{\mathbf{x}}\left(\mathbf{x}|C_1\right)\right)d\mathbf{x}. \tag{2.9}$$

Dans cette nouvelle expression du risque, les deux premiers termes désignent un coût fixe. La minimisation de $\mathcal{R}$ porte donc uniquement sur les deux derniers termes compris dans l'intégrale.

La classification de Bayes peut donc se formuler de la façon suivante :

Si la condition $(c_{21}-c_{11})p_1 f_{\mathbf{x}}\left(\mathbf{x}|C_1\right) > (c_{12}-c_{22})p_2 f_{\mathbf{x}}\left(\mathbf{x}|C_2\right)$ *est vraie, alors le vecteur d'observation* $\mathbf{x}$ *est assigné à la classe* C_1. *Dans le cas contraire,* $\mathbf{x}$ *appartient à la classe* C_2.

Pour simplifier les choses, les deux grandeurs suivantes peuvent être définies :

$$\Lambda(\mathbf{x}) = \frac{f_{\mathbf{x}}\left(\mathbf{x}|C_1\right)}{f_{\mathbf{x}}\left(\mathbf{x}|C_2\right)}, \tag{2.10}$$

$$\xi = \frac{(c_{12}-c_{22})p_2}{(c_{21}-c_{11})p_1}. \tag{2.11}$$

La première grandeur, $\Lambda(\mathbf{x})$, représente un rapport entre deux fonctions de densité de probabilité conditionnelle et est de ce fait appelée rapport de vraisemblance. ξ est appelé seuil de test. Ces deux grandeurs, qui sont toujours positives, ont été introduites pour formuler la classification Bayésienne de façon plus simple. Ainsi :

Si, pour une observation $\mathbf{x}$ *donnée, le rapport de vraisemblance* $\Lambda(\mathbf{x})$ *est supérieur au seuil* ξ, *alors* $\mathbf{x}$ *est assigné à la classe* C_1. *Dans le cas contraire,* $\mathbf{x}$ *appartient à la classe* C_2.

La classification Bayésienne, en terme de calcul, revient essentiellement à déterminer le rapport $\Lambda(\mathbf{x})$. Les probabilités a priori et les coefficients n'affectent pas le rapport $\Lambda(\mathbf{x})$ et ne complexifient pas les calculs, mais affectent uniquement le seuil de test ξ. D'un point de vue pratique, il est plus aisé d'utiliser le logarithme du rapport de vraisemblance plutôt que le rapport lui-même. Ceci se comprend par le fait que le logarithme est une fonction monotone et que $\Lambda(\mathbf{x})$ et ξ sont des grandeurs positives.

b Avec des distributions Gaussiennes

Lorsque les distributions sont Gaussiennes, le vecteur aléatoire $\mathbf{X}$ possède une valeur moyenne qui dépend de son appartenance à la classe C_1 ou à la classe C_2. La matrice de covariance $\mathbf{C}$ de $\mathbf{X}$ par contre est la même pour les deux classes. Les tirages aléatoires issus des classes C_1 et C_2 sont corrélés. La conséquence directe se retrouve dans le fait que la matrice de covariance $\mathbf{C}$ n'est alors pas une matrice diagonale. On supposera toutefois que cette matrice est non singulière, de sorte que son inverse $\mathbf{C}^{-1}$ existe.

Les deux classes C_1 et C_2 sont définies par :

- C_1 : $E[\mathbf{X}] = \boldsymbol{\mu}_1$ et $E[(\mathbf{X}-\boldsymbol{\mu}_1)(\mathbf{X}-\boldsymbol{\mu}_1)^T] = \mathbf{C}$;
- C_2 : $E[\mathbf{X}] = \boldsymbol{\mu}_2$ et $E[(\mathbf{X}-\boldsymbol{\mu}_2)(\mathbf{X}-\boldsymbol{\mu}_2)^T] = \mathbf{C}$.

On peut exprimer la fonction de densité de probabilité conditionnelle de $\mathbf{X}$ à valeurs dans $\mathbb{R}^n$, par :

$$f_{\mathbf{x}}\left(\mathbf{x}|C_i\right) = \frac{1}{(2\pi)^{n/2}\left(\det(\mathbf{C})\right)^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\mathbf{C}^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right),\ i = 1,2. \tag{2.12}$$

On suppose de plus que :

- les deux classes C_1 et C_2 sont équiprobables : $p_1 = p_2 = 1/2$;
- les erreurs de classifications portent le même coût : $c_{12} = c_{21}$;
- les classifications correctes ne représentent aucun coût, c'est-à-dire $c_{11} = c_{22} = 0$.

L'expression (2.10) du rapport de vraisemblance s'écrit alors, en prenant le logarithme naturel et après simplifications :

$$\begin{aligned}\log\Lambda(\mathbf{x}) &= -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\mathbf{C}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T\mathbf{C}^{-1}(\mathbf{x}-\boldsymbol{\mu}_2) \\ &= (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T\mathbf{C}^{-1}\mathbf{x} + \frac{1}{2}(\boldsymbol{\mu}_2^T\mathbf{C}^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^T\mathbf{C}^{-1}\boldsymbol{\mu}_1)\end{aligned} \tag{2.13}$$

En prenant le logarithme naturel et après simplifications, l'expression (2.11) du seuil s'écrit :

$$\log \xi = 0 . \tag{2.14}$$

Les deux dernières expressions, (2.13) et (2.14) signifient que le classificateur Bayésien, lorsque l'environnement est Gaussien, est un classificateur linéaire du type :

$$y = \mathbf{w}^T \mathbf{x} + b , \tag{2.15}$$

avec les paramètres

- $y = \log \Lambda(\mathbf{x})$,
- $\mathbf{w} = \mathbf{C}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$, et
- $b = \frac{1}{2}(\boldsymbol{\mu}_2^T \mathbf{C}^{-1} \boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^T \mathbf{C}^{-1} \boldsymbol{\mu}_1)$.

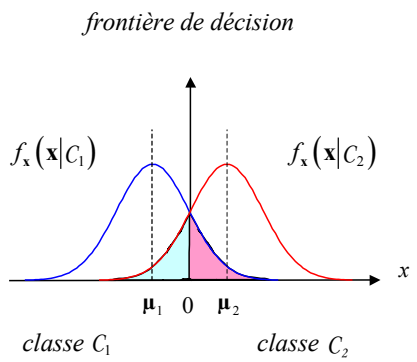


Figure 2-1 : Recouvrement de deux distributions Gausiennes mono dimensionnelles.

Même si le classificateur Bayésien est linéaire, les distributions Gaussiennes peuvent se chevaucher et être alors non séparables. La superficie du recouvrement est dépendante des valeurs moyennes $\boldsymbol{\mu}_1$ et $\boldsymbol{\mu}_2$ ainsi que de la covariance $\mathbf{C}$. Ce cas de recouvrement est représenté par la Figure 2-1 dans le cas d'une variable scalaire (n=1). Le classificateur Bayésien minimise la probabilité de l'erreur de classification, et cette minimisation est indépendante de la superficie de recouvrement de deux classes. Le classificateur Bayésien effectue la séparation entre les deux classes exactement au point d'intersection des deux distributions Gaussiennes de C_1 et de C_2.

Le classificateur Bayésien est paramétrique dans le sens où il dépend de la forme des distributions.

2.3.2 Approche Bayésienne et apprentissage

Il est possible de reproduire le principe de Bayes grâce à un réseau de neurones. Dans ce cas les différentes unités de sorties représentent les classes d'appartenance. L'activation désirée de l'unité à laquelle appartient le patron d'entrée est alors unitaire, toutes les autres activations des unités de sortie ayant une activation nulle. Dans ce cas nous savons qu'après apprentissage, les activations des unités de sortie approchent la probabilité conditionnelle d'appartenir à la classe C_i étant donnée l'observation du patron d'entrée. Le patron est affecté à la classe ayant la probabilité d'appartenance la plus élevée. Ceci est exactement la règle de décision Bayésienne, qui est une règle statistique optimale puisqu'elle minimise le risque d'erreurs de classification. Cependant, il s'agit d'une approximation forcément incomplète. L'estimation des probabilités conditionnelles se base sur un nombre fini d'exemples d'apprentissage.

On suppose que les patrons d'entrées sont représentés par un certain nombre de prototypes associés à des vecteurs de sortie, représentant leur classe d'appartenance. Un même vecteur prototype

se trouve plusieurs fois dans l'ensemble d'apprentissage, avec éventuellement des classifications différentes. Dans ce cas, l'ensemble d'apprentissage est bruité. C'est la situation la plus courante.

Une fois l'apprentissage terminé, les unités de sorties fournissent un degré d'appartenance aux différentes classes, qui peut être utilisé pour prédire la classe d'appartenance de la nouvelle donnée. Il faut rappeler que le seul intérêt d'utiliser des techniques d'estimation telles que les réseaux neuronaux pour la classification (en réalité les probabilités conditionnelles pourraient s'obtenir simplement en comptant) est leur capacité d'extrapolation dans le cas de données d'apprentissage insuffisantes. Dans ce cas les échantillons ne sont plus représentatifs et les probabilités ne peuvent plus être estimées de manière fiable par comptage. Si la distribution est continue, une courbe d'interpolation fournit une meilleure estimation. Ceci peut facilement s'illustrer en imaginant que la distribution d'une variable peut être représentée par un histogramme. Avec un nombre insuffisant d'exemples d'apprentissage, la distribution représentée par un histogramme est mal définie. Certaines valeurs peuvent pratiquement ne pas être représentées. Dans ce cas, une courbe lissée fournit une meilleure estimation de la distribution sous-jacente.

On remarquera que ce problème possède de fortes similitudes avec celui de l'interpolation de fonction à partir d'un ensemble de points.

2.3.3 Exemples d'apprentissage Bayésien

On peut dès lors comprendre la force de l'approche Bayésienne dans les tâches de partitionnement et d'identification. Le classificateur Bayésien en lui-même reste un outil peu souple. Une façon de rendre cet outil mathématique plus flexible consiste à lui adjoindre une forme de flexibilité à travers la théorie de l'apprentissage, à travers l'adaptation par l'exemple. C'est en cela que certains réseaux neuromimétiques sont la suite logique de la théorie de Bayes.

L'intérêt d'utiliser des réseaux de neurones est dû au fait que les principes généraux et les techniques fondamentales de l'inférence statistique s'appliquent aux réseaux de neurones formels, car ce sont des modèles statistiques d'une très grande flexibilité. Cette flexibilité du modèle est liée au choix de l'architecture. L'adéquation du modèle aux données est réalisée par les connexions et des poids entre les cellules de l'architecture. Toutefois, malgré leur flexibilité et leur universalité, les réseaux de neurones ne peuvent se soustraire aux limites intrinsèques de toute modélisation statistique, et plus particulièrement aux limites de l'estimation non-paramétrique, notamment au dilemme biais/variance. Ce dilemme se résume à un compromis entre l'augmentation de la complexité de l'architecture du réseau et le fait que cette augmentation entraîne une dégradation du taux de classement en généralisation, bien qu'il y ait une meilleure adaptation du réseau à l'ensemble d'apprentissage.

Les algorithmes neuronaux de classification, tels que les réseaux MLP, RCE ou GAL, définissent des classes ou des frontières de classes de façon heuristique, soit par des modifications infinitésimales des poids, soit par placement (Hérault et Jutten 1994). Depuis quelques années, notamment avec les réseaux de neurones probabilistes (Anouar et al. 1998), ou les travaux de (Mackay 1992b), de (Geman et al. 1992) ou encore de (Jutten et Comon 1993), l'intérêt d'une approche neuronale Bayésienne se dessine de plus en plus.

Les réseaux neuronaux Bayésiens sont intéressants car ils permettent de tenir compte des probabilités a priori des différentes classes, et peuvent affecter des coûts différents aux différentes erreurs. L'inconvénient majeur réside dans le fait de nécessiter la connaissance parfaite des densités de probabilité conditionnelle $f_{\mathbf{x}}(\mathbf{x}|C_i)$ pour pouvoir prendre la décision optimale.

Les recherches dans ce domaine ont aboutit à des réseaux qui sont de parfaits classificateurs Bayésiens. Un exemple typique de classificateur Bayésien est le réseau RBF, avec autant de noyaux que de classes, suivi d'une couche WTA. Un réseau RBF seul, réalise un estimateur de densité de probabilité.

La grande majorité des réseaux neuronaux peuvent aujourd'hui, soit trouver des fondements dans l'approche Bayésienne, comme les cartes auto-organisatrices SOM (Luttrell 1994), soit trouver une comparaison directe avec ces dernières comme le montre Haykin (Haykin 1994) avec un réseau multicouche. Cette comparaison est issue d'une synthèse des travaux de Lippmann (Lippmann 1987).

2.4 Les modèles de Markov

2.4.1 Introduction

L'évaluation des performances signifie souvent l'étude de l'évolution de la dynamique d'un système. Cette analyse suppose l'existence d'un modèle représentant la réalité du système étudié et la mise en équation de ce modèle. La manière la plus naturelle et la plus efficace pour représenter ce modèle est de définir un ensemble ou un espace d'état dans lequel évoluera le système. On est ainsi amené à étudier deux types de problèmes :

- L'évolution temporelle du système : à partir d'un état initial X_0, le système évolue pour atteindre un état j à l'instant *k* avec une certaine probabilité $p(X_k = \mathrm{j})$. Un cas particulier de ce type d'analyse est l'évaluation des temps de premier passage dans un certain état. En règle générale, l'étude systématique de la dynamique d'un système est extrêmement difficile, pour ne pas dire impossible.
- L'étude des probabilités stationnaires : on s'intéresse alors à la distribution des probabilités d'état quand le système a atteint un état d'équilibre, c'est-à-dire au bout d'un temps infini. Intuitivement, cela signifie que l'on a « oublié » l'état du système à l'instant initial et que les probabilités d'état deviennent constantes. Bien entendu, cela ne signifie pas que le système n'évolue plus (il continue à changer d'état), mais les probabilités de se trouver dans tel ou tel état restent constantes.

L'outil de base pour étudier ces deux types de problèmes est la théorie des processus stochastiques, et son cas particulier appelé la théorie des processus Markoviens. Ces notions ne seront pas étudiées en détail mais seulement présentées dans cette section.

2.4.2 Définition

a Définition d'une chaîne de Markov

Un processus de Markov est un processus tel que, étant donné la valeur de X_k, la valeur de X_s pour $s > k$ ne dépend pas des valeurs prises avant *k*, soit $\{X_u, u < k\}$. Ceci s'écrit :

$$p(X_k \in]a,b] \big| X_{k_1} = x_1, X_{k_2} = x_2, \ldots, X_{k_n} = x_n) = p(X_k \in]a,b] \big| X_{k_n} = x_n), \tag{2.16}$$

pour $k_1 < k_2 < \cdots < k_n < k$.

Un processus de Markov ayant un espace d'état fini ou dénombrable est appelé chaîne de Markov.

Dans l'ensemble des processus stochastiques, les chaînes de Markov caractérisent des processus dits sans mémoire, c'est-à-dire dont l'évolution ne dépend que de l'état courant et pas de tout le passé du processus (c'est-à-dire de la suite des états qui a amené à l'état courant). En d'autres termes, une séquence de variables aléatoires $X_{k_1}, X_{k_2}, \cdots, X_{k_n}, X_{k_{n+1}}$ forme une chaîne de Markov, si la probabilité que le système se trouve à l'état x_{n+1} à l'instant k_{n+1} ne dépend que de la probabilité qu'il se trouve à l'état x_n à l'instant k_n. Une chaîne de Markov peut ainsi être vue comme un certain nombre d'états, liés entre eux deux à deux par de possibles transitions.

Cette caractérisation, qui implique bien entendu un certain nombre d'hypothèses sur les lois des processus élémentaires gouvernant le système, facilite énormément l'analyse et permet le plus souvent d'obtenir des résultats mathématiques relativement simplement.

b Définition d'une chaîne de Markov homogène

Une chaîne de Markov à temps discret est homogène si le processus $\{X_k\}$ qui la caractérise possède les deux propriétés suivantes :

1 - Propriété de Markov : l'évolution du système entre les instants discrets *k* et *k*+1 ne dépend que de son état courant X_k et pas de la suite des états antérieurs, $X_s \quad s < k$. Ceci se traduit par : pour tout *k* et pour tout ensemble d'états de E $(\mathrm{j}_0, \mathrm{j}_1, \ldots, \mathrm{j}_k, \mathrm{j}_{k+1})$, on a la propriété :

$$p(X_{k+1} = \mathrm{j}_{k+1} | X_k = \mathrm{j}_k, \ldots, X_0 = \mathrm{j}_0) = p(X_{k+1} = \mathrm{j}_{k+1} | X_k = \mathrm{j}_k) . \tag{2.17}$$

2 - Propriété d'homogénéité : l'évolution du processus entre deux instants *k* et *k*+1 ne dépend que des états à ces instants et pas de l'instant lui-même. Donc, pour tout couple i, j d'états de E :

$$p(X_{k+1} = \mathrm{j} | X_k = \mathrm{i}) = P(\mathrm{i}, \mathrm{j}) . \tag{2.18}$$

Ces deux définitions caractérisent donc un processus dont l'évolution ne dépend pas du passé (propriété Markovienne) et ne dépend pas de l'instant considéré mais uniquement de l'état présent (propriété d'homogénéité). Ce processus est donc complètement caractérisé par deux informations : l'état initial X_0 et l'ensemble des probabilités de transition P(i, j). Ces dernières sont représentées par la matrice de transition $\mathbf{P}$ dont les coordonnées sont les P(i, j).

2.4.3 Chaîne de Markov et apprentissage

L'état d'un réseau de neurones dépend en général du temps *k*. Modéliser l'état de ce réseau au temps *k* peut se faire à l'aide des suites aléatoires X_k indexées dans le temps. Le cas le plus simple de telles suites est celui où les variables aléatoires sont indépendantes. Très souvent, cela n'est pas vérifié. Une situation intermédiaire, entre l'indépendance complète et la dépendance de toutes les variables aléatoires, peut être trouvée en considérant que l'état courant ne dépend que de l'instant précédent et non pas du passé.

La plupart des règles d'apprentissage, celle du Perceptron, celle des mémoires associatives, l'état d'un réseau de neurones stochastiques ou encore l'auto-organisation de Kohonen sont des exemples courants de chaînes de Markov.

Modéliser un réseau de neurones par une chaîne de Markov se fait principalement avec comme objectif la démonstration de sa convergence. Les propriétés de convergence des cartes SOM ont été étudiées dans (Ritter et al. 1992) et dans (Ritter et Schulten 1988a), avec un modèle Markovien. Une approche similaire a été adoptée pour analyser la convergence de cartes mono-dimensionnelle dans (Kohonen 1984) et dans (Cottrell et Fort 1986) ou (Cottrell et al. 1994).

Dans (Neal 1996), l'auteur utilise des réseaux multicouches pour réaliser un apprentissage Bayésien. L'étude et la convergence de son apprentissage se base sur la théorie des chaînes de Markov. En effet, Neal cherche à contrôler l'apprentissage et tente d'ajuster la complexité structurelle du réseau grâce aux chaînes de Markov. Sa démarche aboutit à une méthode hybride de Monte Carlo qu'il applique à un robot à deux degrés de liberté dans un plan : il apprend la transformation liant l'espace plan à l'espace articulaire. Ses travaux sont la suite de ceux de (Mackay 1992a).

2.5 Bilan de l'apport des statistiques au connexionisme

En théorie de l'apprentissage, les modèles statistiques sont intéressants pour plusieurs raisons. Tout d'abord, les données observées contiennent très souvent du bruit et la valeur mesurée ne représente pas tout à fait la valeur réelle. Le bruit peut dépendre soit partiellement de la valeur de la variable (une grande valeur se verra ajouter plus de bruit qu'une petite), soit d'autres variables non accessibles. Pour cette dernière raison, le bruit est souvent modélisé comme une variable aléatoire.

La variable mesurée peut, elle-même, dépendre d'autres variables que des variables d'entrées. Un modèle statistique peut dans cette situation compenser le manque d'information. Notons qu'il est

souvent impossible d'avoir toutes les informations sur un phénomène et donc de les fournir en entrée à un modèle (au sens large du terme : algorithme, apprentissage, modèle paramétrique…).

Enfin, il faut savoir que tout modèle a ses limites. Si tous les mécanismes d'un phénomène ne sont pas connus (et c'est presque toujours le cas lorsqu'on emploie des méthodes d'apprentissage), une représentation statistique peut compenser les imperfections et le manque de précision d'un modèle en lui ajoutant un phénomène aléatoire, plutôt que de générer de manière déterministe une mauvaise réponse. C'est ce que fait typiquement le filtre de Kalman (KF) par rapport aux méthodes d'estimation itératives par les moindres carrés.

Le connexionisme ne peut aujourd'hui que profiter de ces nouvelles considérations, d'autant plus qu'il existe un certain nombre de points communs entre les approches statistiques et les approches neuronales. Elles possèdent des objectifs communs à travers les tâches de classification et d'estimation de densité de probabilité (Fiori et Bucciarelli 2001). Ce qui relie principalement ces deux approches, c'est qu'elles utilisent toutes les deux un grand nombre d'observations pour établir un modèle de connaissance. Ce dernier point entraîne une comparaison logique entre elles.

Ne nous arrêtons pas à cette comparaison. Nous avons vu dans les paragraphes précédents que la théorie des statistiques offre des outils pour aider et compléter le connexionisme. En effet, elle propose des méthodes qui permettent d'analyser les performances d'un apprentissage. Le biais et la variance d'un réseau permettent, respectivement, de caractériser les capacités de généralisation et la sensibilité d'un réseau. A fortiori, aucun calcul formel de l'espérance de l'estimateur neuronal n'est possible et aucun intervalle de confiance ne peut être fourni pour la réponse du réseau. L'obtention d'un intervalle de confiance pour le taux de généralisation théorique nécessite le recours à des résultats plus généraux issus de la théorie de l'apprentissage. Cette dernière permet d'étudier la convergence d'un apprentissage grâce à l'évaluation d'un risque empirique. L'aboutissement de ces outils est la dimension de Vapnik-Chervonenkis qui cherche à optimiser le rapport entre la complexité structurelle et la capacité d'apprentissage d'un réseau. D'autres outils, comme les chaînes de Markov tentent de modéliser les processus d'apprentissage, toujours en vue d'expliquer la convergence ou la non convergence d'un réseau par rapport à la tâche à apprendre.

Nous allons voir, dans le Chapitre 3, que la théorie des statistiques apportent des contributions aux architectures modulaires de réseaux de neurones, en particulier dans les approches d'apprentissage à modèle multiple. Elle ouvre également des horizons nouveaux dans l'apprentissage lui-même. Dans la section qui suit, nous introduisons une approche statistique Bayésienne, le filtre de Kalman. Cet estimateur, qui peut être modélisé par une chaîne de Markov, constitue en quelque sorte une nouvelle façon d'adapter les poids d'un certains nombre de réseaux neuronaux. Cette nouvelle règle d'apprentissage possède de sérieux atouts par rapport aux méthodes traditionnelles utilisant les gradients instantanés d'une surface d'erreur.

3 Le filtrage de Kalman

Le traitement de données incluant des variables aléatoires date des années 1800 à partir des apports théoriques de Gauss, qui a inventé la méthode des moindres carrés déterministes. La théorie de l'estimation progresse un siècle plus tard avec Fisher (Fisher 1912) qui, à travers la notion de densité de probabilité, introduit l'approche du maximum de vraisemblance. Wiener (Wiener 1948) met au point, dans les années quarante, un filtre optimal en utilisant la théorie des processus aléatoire. Kolmogoroff (Kolmogoroff 1941) dans la même période traite le problème numérique que Wiener avait résolu dans le cas continu. Les travaux de Wiener ont par la suite été étendus pour des systèmes non linéaires, et Kalman (Kalman 1960) a développé un filtre optimal récursif basé sur la représentation d'état et parfaitement adapté à une implémentation sur calculateurs.

Ce filtre, qui est un observateur au sens de l'automatique, est parfaitement intégré dans la théorie de l'estimation.

3.1 Généralités

Le problème du filtrage non linéaire est de calculer à chaque pas de temps la loi conditionnelle π_k de l'état $\mathbf{x}_k$, sachant une réalisation des observations $\mathbf{z}_{1:k} = (\mathbf{z}_1, \dots, \mathbf{z}_k)$ jusqu'à l'instant courant *k*, i.e.,

$$\pi_k = P[\mathbf{x}_k \in dx | \mathbf{z}_{1:k}]. \tag{2.19}$$

La suite de probabilités conditionnelles $(\pi_k)_{k\geq 0}$, avec la convention $\pi_0 = p_0$, est appelée filtre optimal. Pour satisfaire des contraintes temps réel, il est souhaitable d'obtenir un algorithme récursif de calcul de π_k, c'est-à-dire un calcul de π_k qui ne soit fonction que de la dernière observation $\mathbf{z}_k$ et de la loi conditionnelle précédente π_{k-1}.

Tandis que le calcul du filtre optimal est un problème difficile, l'évolution temporelle du filtre est plus facile à décrire. Le passage de π_{k-1} à π_k fait intervenir la loi conditionnelle $\pi_{k|k-1}$ de l'état $\mathbf{x}_k$ sachant une réalisation des observations $\mathbf{z}_{1:k-1}$ jusqu'à l'instant précédent :

$$\pi_{k|k-1}(dx) = P[\mathbf{x}_k \in dx | \mathbf{z}_{1:k-1}]. \tag{2.20}$$

La suite de probabilités $(\pi_{k|k-1})_{k\geq 0}$, où $\pi_{0|-1} = p_0$, est appelée filtre de prédiction. Une approximation du filtre optimal comme du filtre de prédiction peut être réalisée entre autre, par le filtre de Kalman.

L'objet de cette section est simplement de rappeler la forme récursive de l'estimateur de Kalman (Kalman 1960).

3.2 Le filtre de Kalman

Le problème du filtrage consiste à estimer l'état d'un système dynamique aux vues d'observations bruitées des états passés du système. Ce problème est suffisamment général pour trouver de nombreuses applications variées telles que le pistage de cible, la commande optimale, ou l'évaluation de risque en finance. C'est ce contexte applicatif riche qui motive pour l'essentiel la recherche dans ce domaine. Dans ce travail de thèse, nous avons choisi de considérer le cas particulier du filtrage en temps discret. Ce choix est naturellement justifié pas le caractère nécessairement échantillonné des observations délivrées par l'instrument de mesure dans notre application. Précisément, nous cherchons à estimer à chaque instant d'observation, la valeur d'un processus aléatoire Markovien, $(\mathbf{x}_k)_{k\geq 1}$, à valeurs dans $\mathbb{R}^n$ et de transitions connues. Ce processus état est indirectement lié à un processus observation $(\mathbf{z}_k)_{k\geq 1}$ dont on connaît une réalisation. Le but est alors de calculer (de manière approchée) à chaque instant d'observation *k*, la distribution de probabilité conditionnelle π_k de l'état à l'instant courant connaissant les observations passées,

$$\mathbf{x}_0 = \pi_0, \text{ et pour } k \geq 1,\ \pi_k(dx) = P[\mathbf{x}_k \in dx | \mathbf{z}_{1:k}].$$

Le filtre de Kalman permet en effet le calcul exact et rapide du filtre optimal, lorsque les modèles d'état et d'observation ne font intervenir que des fonctions linéaires et des bruits additifs Gaussiens. On montre que dans ce cas, la loi conditionnelle recherchée est Gaussienne. Il suffit alors de propager sa moyenne et sa variance : le problème est de dimension finie. Mais, en toute généralité, le problème du filtrage non linéaire n'admet pas de solution de dimension finie. En pratique, on continue à utiliser des versions dérivées du filtre de Kalman, comme le filtre de Kalman étendu (EKF), qui linéarise le modèle autour de l'estimation courante, et qui est depuis une trentaine d'années le filtre non linéaire le plus utilisé. Il n'a été que très récemment justifié, en particulier grâce aux travaux de Picard (Picard 1989) qui montrent son efficacité dans certains cas d'observations très précis. Hormis ces cas, l'erreur commise par le filtre de Kalman étendu n'est en général pas connue. On observe d'ailleurs des mauvais résultats lorsque le système est très non linéaire, ou lorsque le filtre est mal initialisé. Une autre approche plus rigoureuse, propose de calculer la solution des équations du filtrage par des techniques de maillage de l'espace d'état. Cependant, la complexité de ces méthodes numériques, qui augmente fortement avec la dimension de l'espace, les rend inutilisables en temps réel

à partir de la dimension 3. Plus récemment, les méthodes de Monte Carlo ont été proposées comme une approche attrayante. Connues pour leur quasi-insensibilité à la dimension de l'espace et aux non linéarités du système, leur utilisation semble particulièrement adapté au filtrage non linéaire. Les méthodes particulaires proposent d'approcher la loi conditionnelle recherchée par une mesure empirique, chargeant les points d'un système de particules aléatoires évoluant avec le signal et les observations. Cependant, les algorithmes particulaires développés jusqu'à présent montrent un comportement instable en dehors de certaines hypothèses. On observe notamment la divergence de ces méthodes dans le cas de systèmes faiblement bruités.

Les points forts du filtre de Kalman restent l'utilisation d'une représentation théorique maîtrisée, le modèle d'état et son implémentation adaptée aux calculateurs grâce à une formulation récursive.

3.2.1 Modèle d'état et formulation récursive du filtre

Cette forme d'algorithme, qui utilise le passage d'un espace à un autre (avec une étape de prédiction) pour prendre en compte une observation (à l'aide d'une étape de recalage), sera utilisé pour le développement du filtre de Kalman adaptatif au sens de la représentation d'état présenté au Chapitre 4.

a L'espace d'état

En considérant le filtre de Kalman, on suppose d'une part que les observations sont probabilistes, et d'autre part que la connaissance de l'évolution entre l'observation k et l'observation k+1 est celle d'un modèle dynamique régissant l'évolution d'un vecteur d'état entre ces deux instants. L'étude la plus complète sur ce type de filtre se trouve dans (Gelb 1974). Des investigations moins détaillées mais néanmoins complètes sont disponibles dans (Stengel 1986), dans (Haykin 1996) ou encore dans (Brown et Hwang 1996).

Un modèle linéaire Gaussien est un modèle où le processus suit une évolution linéaire avec un bruit additif Gaussien, et où l'observation est une fonction linéaire de l'état, entachée d'un bruit additif Gaussien. Ce modèle est mathématiquement décrit par le système suivant :

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{D}\mathbf{w}_k \,, \tag{2.21}$$

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \,. \tag{2.22}$$

Dans les expressions (2.21) et (2.22), $\mathbf{x}_k \in \mathbb{R}^n$ représente l'état du processus, $\mathbf{z}_k \in \mathbb{R}^p$ est l'observation disponible à l'instant k, $\mathbf{u}_k$ est une entrée de forçage, et $\mathbf{w}_k \in \mathbb{R}^s$ et $\mathbf{v}_k \in \mathbb{R}^p$ représentent respectivement un bruit de modélisation et un bruit de mesure. Les bruits traduisent l'incertitude que l'on a sur le modèle et le bruit de mesure du capteur. Ces bruits sont blancs Gaussiens de covariance respective $\mathbf{Q} \in \mathbb{R}^{s\times s}$ et $\mathbf{R} \in \mathbb{R}^{p\times p}$. Ils sont mutuellement indépendants et sont indépendants de la condition initiale $\mathbf{x}_0$. Cette condition initiale de l'état du processus, est de moyenne $E[\mathbf{x}_0] = \overline{\mathbf{x}}_0$ et de covariance $\mathbf{Q}_0$.

Les matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times r}$, $\mathbf{D} \in \mathbb{R}^{n\times s}$ et $\mathbf{C} \in \mathbb{R}^{p\times n}$ sont déterministes. La matrice $\mathbf{A}$ est la matrice de transition de l'état de l'instant k à k+1, $\mathbf{B}$ est la matrice de commande, $\mathbf{C}$ est la matrice d'observation et la matrice $\mathbf{D}$ permet de lier le bruit de modélisation $\mathbf{w}_k$ au vecteur d'état.

Il a été montré que si les conditions précédentes sont respectées, alors le processus $(\mathbf{x}_k, \mathbf{z}_k)_{k\geq 1}$ est Gaussien et que, par conséquent, les lois conditionnelles π_k sont elles aussi Gaussiennes. Le problème est alors de dimension finie et il suffit de calculer l'espérance $\hat{\mathbf{x}}_k$ et la matrice de covariance $\mathbf{P}_k$ de π_k. Ces deux grandeurs sont calculées récursivement par le filtre de Kalman.

b La formulation récursive

Lorsque les conditions initiales du filtre sont données par $\hat{\mathbf{x}}_0 = \overline{\mathbf{x}}_0$ et par $\mathbf{P}_0 = \mathbf{Q}_0$, les grandeurs $\hat{\mathbf{x}}_k$ et $\mathbf{P}_k$ sont estimées par les étapes de prédiction et de correction suivant les équations :

- équations de prédiction

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\mathbf{u}_k \,, \tag{2.23}$$

$$\mathbf{P}_{k+1|k} = \mathbf{A}\mathbf{P}_k\mathbf{A}^T + \mathbf{D}\mathbf{Q}_k\mathbf{D}^T \,, \tag{2.24}$$

- équations de correction

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathbf{C}^T(\mathbf{C}\mathbf{P}_{k+1|k}\mathbf{C}^T + \mathbf{R}_k)^{-1} \,, \tag{2.25}$$

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}(\mathbf{z}_{k+1} - \mathbf{C}\hat{\mathbf{x}}_{k+1|k}) \,, \tag{2.26}$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C})\mathbf{P}_{k+1|k} \,. \tag{2.27}$$

La première étape d'évolution du filtre optimal utilise la connaissance a priori du système. En effet, elle permet la prédiction de l'état du système $\hat{\mathbf{x}}_{k+1|k}$ et de sa précision $\mathbf{P}_{k+1|k}$ (matrice de covariance de l'erreur a priori), à partir de l'état $\hat{\mathbf{x}}_k$ et du modèle d'évolution du système. $\mathbf{P}_{k+1|k}$ est la matrice de covariance associée à l'état $\hat{\mathbf{x}}_{k+1|k}$.

La deuxième étape utilise la connaissance a posteriori fournie par l'observation. La mesure $\mathbf{z}_{k+1}$ permet de calculer une erreur de prédiction de l'observation $\hat{\mathbf{z}}_{k+1|k}$. Cette erreur est un terme d'innovation :

$$\hat{\mathbf{s}}_{k+1|k} = \mathbf{z}_{k+1} - \mathbf{C}\hat{\mathbf{x}}_{k+1|k} = \mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k} \,. \tag{2.28}$$

Elle sert à pondérer le gain $\mathbf{K}_k$ du filtre. Ce gain est généralement appelé gain de Kalman, et sert à déduire l'estimation de l'état $\hat{\mathbf{x}}_{k+1}$ et sa matrice de covariance associée $\mathbf{P}_{k+1}$. En fait, la matrice $\mathbf{K}_k$ minimise la matrice de covariance de l'erreur a posteriori $\mathbf{P}_k = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T]$.

Si la covariance de l'erreur de mesure $\mathbf{R}$ s'approche de zéro, le gain fait peser le résidu plus fort et

$$\lim_{\mathbf{R}_k \to 0} \mathbf{K}_k = \mathbf{C}^{-1} \,, \tag{2.29}$$

c'est-à-dire que plus l'erreur de mesure diminue, plus le filtre fait confiance à la véritable mesure $\mathbf{z}_k$ au détriment de $\mathbf{C}\hat{\mathbf{x}}_{k|k-1}$. Inversement, lorsque la covariance de l'erreur a priori tend vers zéro, le gain $\mathbf{K}_k$ diminue également et

$$\lim_{\mathbf{P}_{k|k-1} \to 0} \mathbf{K}_k = 0 \,. \tag{2.30}$$

On remarque que le filtre maintient les deux premiers moments de la distribution sur les états, et que la variance de cette distribution est bien :

$$E[\mathbf{x}_k] = \hat{\mathbf{x}}_k \,, \tag{2.31}$$

$$E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] = \mathbf{P}_k \,, \tag{2.32}$$

$$p(\mathbf{x}_k | \mathbf{z}_k) \sim N(\hat{\mathbf{x}}_k, \mathbf{P}_k) \,. \tag{2.33}$$

Le principe de fonctionnement du filtre est entièrement représenté sur la Figure 2-2.

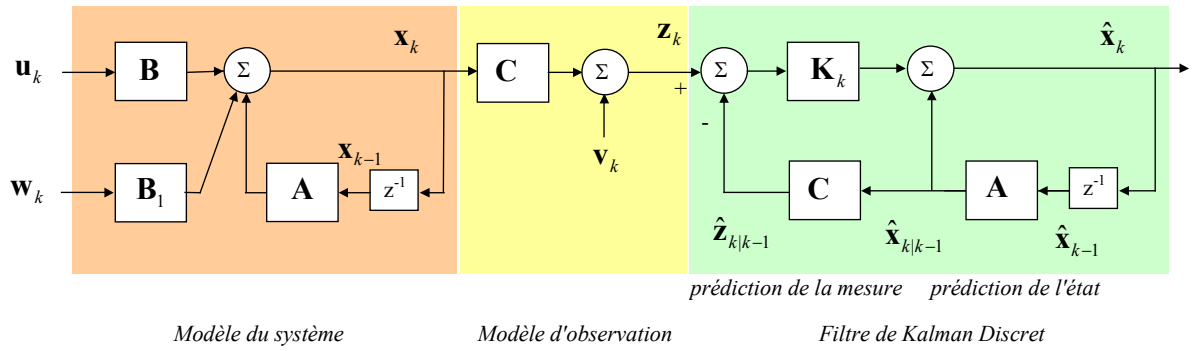


Figure 2-2 : Principe du filtre de Kalman.

On note que les suites des matrices de covariances $(\mathbf{P}_k)_{k\geq 0}$ et $(\mathbf{P}_{k+1|k})_{k\geq 0}$, ainsi que la suite $(\mathbf{K}_k)_{k\geq 0}$ des gains, ne dépendent pas des observations $(\mathbf{z}_k)_{k\geq 0}$. Les deux premières sont les solutions d'une équation de Ricatti. Cette démonstration est reprise dans (Gelb 1974, page 136).

Il existe au moins trois approches différentes qui aboutissent aux équations du filtre de Kalman :

- une première approche, Bayésienne, consiste à calculer explicitement les deux étapes ;
- une seconde approche, dite de projection et développée dans (Anderson et Moore 1979), consiste à montrer que les innovations $\hat{\mathbf{s}}_{k+1|k}$ représentent une suite de variables aléatoires indépendantes et à projeter le processus état sur la base orthogonale associée ;
- une dernière revient à associer le problème du filtrage linéaire Gaussien à celui des moindres carrés récursifs, et à le résoudre en optimisant un critère correspondant aux vraisemblances des observations sous une contrainte correspondant à la connaissance a priori de l'état (Franklin et al. 1998).

Le filtre de Kalman est une méthode optimale. L'estimation qu'il fournit est celle qui minimise les erreurs quadratiques moyennes. Même si les bruits $\mathbf{w}_k$ et $\mathbf{v}_k$ ne respectent pas l'hypothèse de départ, à savoir être Gaussiens, en restant dans le cadre d'un système dynamique linéaire, le filtre de Kalman fournit toujours l'estimée linéaire à variance minimale de l'état du système, et c'est pour cette raison qu'il est si répandu (Oudjane 2000).

3.2.2 Les extensions du filtre de Kalman

a L'estimateur $\alpha-\beta$

L'estimateur $\alpha-\beta$ a été mis au point dans le souci de simplifier les équations permettant d'obtenir l'estimation de l'état d'un système. Ce filtre est identique au filtre de Kalman conventionnel mais utilise un gain $\mathbf{K}$ constant. Lorsque le vecteur d'état et sa vitesse sont estimés on parle de filtre $\alpha-\beta$, et si à ces estimations s'ajoutent celles des accélérations, de filtre $\alpha-\beta-\gamma$. Il ne possède évidemment pas les mêmes performances que celles d'un filtre optimal ; néanmoins, il offre une alternative aux filtres statistiques récursifs et possède l'avantage de fournir une solution simple à implémenter et à utiliser. Il est clair que le fait de ne pouvoir disposer de la mesure de la position porte préjudice à la qualité et à la stabilité de ce filtre.

b Le filtre étendu et le filtre linéarisé

La formulation récursive du filtre de Kalman permet à tout instant d'estimer l'état d'un système dynamique linéaire. Dans un contexte plus général, et dès lors qu'apparaissent des non linéarités, la représentation d'état d'un système dynamique peut s'écrire à l'aide des équations suivantes :

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, k) + \mathbf{B}\mathbf{u}_k + \mathbf{D}\mathbf{w}_k , \tag{2.34}$$

$$\mathbf{z}_k = \mathbf{H}(\mathbf{x}_k, k) + \mathbf{v}_k . \tag{2.35}$$

Les hypothèses faites sur les grandeurs des lois linéaires d'évolution (2.21) et d'observation (2.22) sont toujours considérées comme valables pour les expressions (2.34) et (2.35). Seules les matrices constantes $\mathbf{A}$ et $\mathbf{C}$ sont, dans le cadre de la non linéarité, remplacées par des fonctions dépendant essentiellement du temps et du vecteur d'état.

Connaissant les moments d'ordre 1 et 2 du vecteur d'état, $\hat{\mathbf{x}}_k \sim N(\overline{\mathbf{x}}_k, \mathbf{P}_k)$, la formulation récursive du filtre de Kalman étendu est donnée par les équations suivantes :

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F}(\hat{\mathbf{x}}_k, k) + \mathbf{B}\mathbf{u}_k , \tag{2.36}$$

$$\mathbf{P}_{k+1|k} = \mathcal{F}(\hat{\mathbf{x}}_{k+1|k}, k)\mathbf{P}_k \mathcal{F}^T(\hat{\mathbf{x}}_{k+1|k}, k) + \mathbf{D}\mathbf{Q}_k\mathbf{D}^T , \tag{2.37}$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathcal{H}^T(\hat{\mathbf{x}}_{k+1|k},k)\left(\mathcal{H}^T(\hat{\mathbf{x}}_{k+1|k},k)\mathbf{P}_{k+1|k}\mathcal{H}^T(\hat{\mathbf{x}}_{k+1|k},k)+\mathbf{R}_k\right)^{-1}, \tag{2.38}$$

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\left(\mathbf{z}_{k+1} - \mathbf{H}^T(\hat{\mathbf{x}}_{k+1|k},k)\right), \tag{2.39}$$

$$\mathbf{P}_{k+1} = \left(\mathbf{I} - \mathbf{K}_{k+1}\mathcal{H}(\hat{\mathbf{x}}_{k+1|k},k)\right)\mathbf{P}_{k+1|k}, \tag{2.40}$$

avec les deux définitions suivantes :

$$\mathcal{F}^T(\hat{\mathbf{x}}_{k+1|k},k) = \left.\frac{\partial \mathbf{F}(\hat{\mathbf{x}}_{k+1|k},k)}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k=\hat{\mathbf{x}}_{k+1|k}}, \tag{2.41}$$

$$\mathcal{H}^T(\hat{\mathbf{x}}_{k+1|k},k) = \left.\frac{\partial \mathbf{H}(\hat{\mathbf{x}}_{k+1|k},k)}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k=\hat{\mathbf{x}}_{k+1|k}}. \tag{2.42}$$

Les termes $\mathcal{F}^T(\hat{\mathbf{x}}_{k+1|k},k)$ et $\mathcal{H}^T(\hat{\mathbf{x}}_{k+1|k},k)$ des expressions (2.41) et (2.42) sont respectivement les matrices Jacobiennes de la fonction du système et de la fonction de mesure.

Le filtre de Kalman étendu utilise une linéarisation des équations de transition et de mesure autour de $\hat{\mathbf{x}}_{k+1|k}$ et $\hat{\mathbf{x}}_k$. Cette linéarisation est obtenue par un développement en série de Taylor du premier ordre des deux fonction matricielles $\mathbf{F}(\mathbf{x}_k,k)$ et $\mathbf{H}(\mathbf{x}_k,k)$. Il est important de remarquer qu'aucun élément ne permet d'assurer sa convergence. En effet, si le passage d'une estimation à l'autre tend à sortir de la zone de linéarité, alors le modèle linéarisé n'est plus valable et il est possible que le filtre diverge. Une étude du comportement du filtre étendu est synthétisée à travers plusieurs articles réunis dans (Chen 1993).

L'effet produit par les non linéarités est illustré sur la Figure 2-3. Cette figure montre clairement que la transformation d'une fonction de densité de probabilité Gaussienne à travers un élément linéaire donne une fonction de densité de probabilité Gaussienne, alors que la transformation à travers un élément non linéaire retourne une fonction de densité de probabilité non Gaussienne. Les deux premiers moments, qui suffisent à décrire complètement une fonction de densité de probabilité Gaussienne, ne permettent plus de décrire la fonction de densité de probabilité Gaussienne transformée par un élément non linéaire. De plus, on voit que deux entrées centrées sur la même valeur moyenne mais, l'une à variance faible et l'autre à variance plus élevée, sont déformées différemment. Celle avec un moment d'ordre deux élevé est bien plus déformée que celle avec un moment d'ordre deux faible. Ces considérations ne sont jamais prises en compte lorsqu'une fonction linéaire est approximée par un modèle linéarisé.

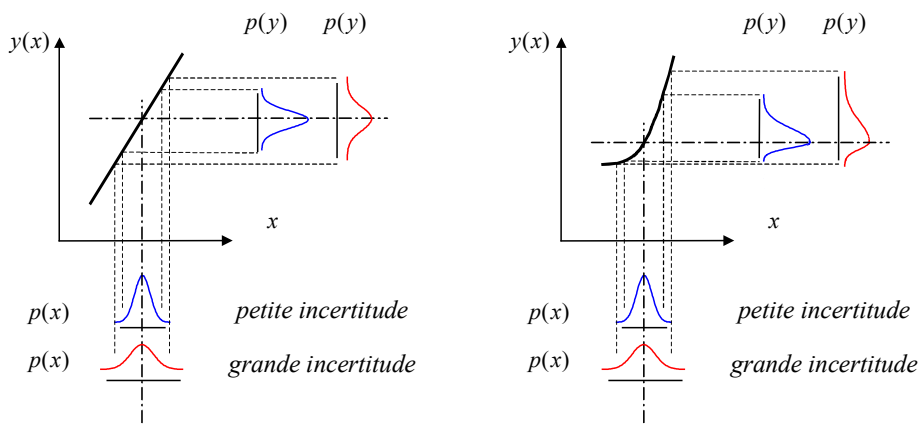


Figure 2-3 : Transmission de fonctions de densité de probabilité Gaussiennes à travers un élément linéaire (à gauche) et un élément non linéaire (à droite).

Linéariser les fonctions matricielles $\mathbf{F}(\mathbf{x}_k, k)$ et $\mathbf{H}(\mathbf{x}_k, k)$ autour d'un vecteur $\overline{\mathbf{x}}_k$ représentatif de données du système préalablement acquises, conduit au filtre de Kalman linéarisé. La matrice $\mathbf{K}_k$, le gain de Kalman, peut alors être calculée et définitivement fixée avant le traitement et l'acquisition des observations. Il en va de même pour les deux matrices Jacobiennes.

Procéder de la sorte conduit logiquement aux équations :

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F}(\overline{\mathbf{x}}_k, k) + \mathbf{B}\mathbf{u}_k , \tag{2.43}$$

$$\mathbf{P}_{k+1|k} = \mathcal{F}(\overline{\mathbf{x}}_k, k)\mathbf{P}_k \mathcal{F}^T(\overline{\mathbf{x}}_k, k) + \mathbf{D}\mathbf{Q}_k\mathbf{D}^T , \tag{2.44}$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathcal{H}^T(\overline{\mathbf{x}}_k, k)\left(\mathcal{H}^T(\overline{\mathbf{x}}_k, k)\mathbf{P}_{k+1|k}\mathcal{H}^T(\overline{\mathbf{x}}_k, k) + \mathbf{R}_k\right)^{-1} , \tag{2.45}$$

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\left(\mathbf{z}_{k+1} - \mathbf{H}^T(\overline{\mathbf{x}}_k, k)\right), \tag{2.46}$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathcal{H}(\overline{\mathbf{x}}_k, k))\mathbf{P}_{k+1|k} . \tag{2.47}$$

Le filtre linéaire est généralement moins performant que le filtre étendu parce que $\overline{\mathbf{x}}_k$ n'est pas aussi proche que $\hat{\mathbf{x}}_k$ de la valeur réelle de l'état. Il est cependant bien moins coûteux en ressources de calcul.

3.2.3 Autres variantes

Les filtres étendus et linéarisés sont parfois des représentations trop simplistes d'un problème. Leurs performances se détériorent lorsque les distributions (ou les densités de probabilité dans le cas continu) des variables aléatoires sont déformées par les transformations non linéaires et ne suivent plus la loi normale. Le EKF n'est alors plus qu'un estimateur ad hoc qui réalise une approximation (linéaire) optimale au sens de la loi de Bayes. Dans ses travaux (Julier et Uhlmann 1997; Julier et al. 1995), Julier tente de développer un EKF qui préserve la normalité des distributions des variables aléatoires malgré les non linéarités.

a Les filtres d'ordre supérieur

Le filtre de Kalman étendu est appelé filtre d'ordre un, parce qu'il n'utilise qu'une expansion à l'ordre un des lois non linéaires. Des filtres non linéaires d'ordre supérieur peuvent être construits sur la base de développements en série de Taylor d'ordre supérieur à un. On peut ainsi trouver dans la littérature, l'utilisation de filtre d'ordre deux, ou encore des "iterated EKF" (Gelb 1974). Ce dernier filtre, en plus d'utiliser un terme du deuxième ordre dans le développement du polynôme de Taylor – pour estimer $\hat{\mathbf{x}}_{k+1|k}$ dans l'expression (2.40) avec une précision meilleure – utilise une procédure itérative pour mettre plusieurs fois à jour $\hat{\mathbf{x}}_{k+1|k}$, $\mathbf{P}_{k+1|k}$ et $\mathbf{K}_k$ en linéarisant les Jacobiennes à chaque fois autour de la dernière estimation. Il est évident que ce filtre n'est pas conçu pour un environnement temps réel. Plus récemment, dans (Ormoneit 1999) est présentée une méthode qui consiste à mettre à jour les grandeurs a priori et le gain de Kalman plusieurs fois de suite à partir d'elles mêmes (sans utiliser de mesure), avec une linéarisation effectuée toujours autour de l'estimation la plus récente. Elle est utilisée pour l'ajustement des poids d'un réseau neuromimétique.

b Le filtre quasi-linéaire

Dans (Gelb 1974) (pages 204-209 et table 6.2-1), un estimateur non linéaire par linéarisation statistique, appelé filtre quasi-linéaire dans (Stengel 1986) – en anglais "statistically linearized filter" ou "quasi-linear filter" – est présenté. Cette méthode est très intéressante, car elle possède l'avantage de ne pas nécessiter le calcul de dérivées à l'instar des méthodes basées sur le développement en série de Taylor. De cette façon, un grand nombre de non linéarités (phénomènes de saturation, de seuil, palier, etc.) peut être pris en compte, sans remplacer les discontinuités et les déviations de la fonction non linéaire par des fonctions "smooth" ou plus simples, ou encore par des approximations locales. Un

inconvénient persiste : la fonction de densité de probabilité de la fonction doit être connue. D'un point de vue statistique, cette méthode est plus précise et plus robuste.

c Le filtre décentralisé et le filtre fédéré

Un autre filtre, qu'il est utile de mentionner est le filtre décentralisé, présenté dans (Brown et Hwang 1996, pages 371-377) et dans (Hashemipour et al. 1988). Les filtres qui ont été vus jusque là, peuvent être considérés comme centralisés. Le concept de filtre décentralisé utilise plusieurs filtres, chacun étant chargé d'intégrer une ou plusieurs séries d'observations. Cette structure se justifie essentiellement par le fait que toutes les données d'observations ne sont pas toujours disponibles à chaque instant discret. Des filtres locaux sont alors chargés de prendre en compte les observations à leur propre rythme, sans les partager avec un autre filtre. Ces filtres sont autonomes et leurs sorties sont utilisées comme les entrées d'un filtre maître appelé filtre de fusion. Les inconvénients de cette structure en cascade se limitent à deux points :

- si en plus des estimations, la précision des erreurs d'estimation faites par les filtres locaux – idéalement la matrice $\mathbf{P}_k$ – n'est pas transmise au filtre principal, celui-ci risque de diverger ;
- chaque filtre local doit utiliser un vecteur d'état complet, c'est-à-dire celui du filtre principal, sous peine de voir l'ensemble de la structure rester très peu efficace.

Des structures plus robustes, comme celle de la Figure 2-4, incluent une boucle de retour du filtre principal aux filtres locaux. Pour entièrement respecter le principe de fonctionnement du filtre de Kalman, qui met à jour dans le temps les moments d'ordre 1 et 2 pour calculer son estimation, le filtre principal retourne les moments d'ordre 1 et d'ordre 2 de son estimation aux filtres locaux. Chaque filtre local peut ainsi se mettre à jour avec les estimations globales qui sont plus précises, puisque issues d'une fusion des données de tous les filtres. Le filtre principal maintient l'optimalité de l'estimation globale, même si les filtres locaux utilisent un vecteur d'état complet.

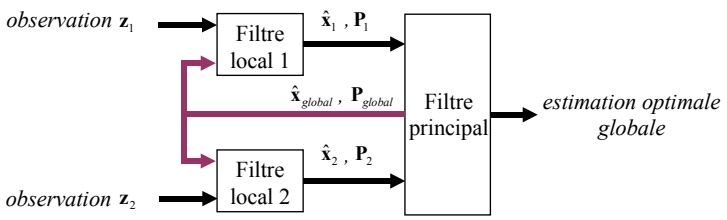


Figure 2-4 : Le filtrage de Kalman décentralisé avec retour.

Pour les deux architectures décentralisées, avec et sans retour, les estimations locales atteignent un certain degré d'optimalité puisque chaque filtre local met à jour une estimation complète du vecteur d'état à partir de ses propres observations. Ne pas utiliser un vecteur d'état complet pour les filtres locaux conduit à un comportement sous-optimal.

Récemment, des structures plus complexes apparaissent sous le terme de filtre fédéré. L'architecture des filtres fédérés est la même que celle d'un filtre décentralisé, à la différence près que chaque filtre local ne reçoit qu'une partie de l'information contenue dans la boucle de retour, et que certaines des informations sont communes à tous les filtres locaux. Dans un filtre fédéré, les observations arrivent directement au filtre principal. La théorie concernant les filtres fédérés est très peu évidente dans le sens où on ne peut pas en analyser la stabilité aussi clairement que pour un filtre centralisé.

d Autres filtres

Les structures en cascade de plusieurs filtres sont également une approche qui semble intéressante. Par exemple, dans (Xie et al. 1998), deux filtres sont disposés en cascade. Chacun reçoit un vecteur d'entrée identique, mais chacun prend aussi sa propre sortie et la sortie de l'autre filtre en entrée. Cette approche en cascade est appliquée au suivi de l'iris de l'œil et à la compensation des mouvements de la tête. Son avantage, par rapport à un filtre centralisé (unique), est de réduire la complexité du problème. Les deux tâches découplées sont traitées séparément par chacun des filtres. La structure en cascade entraîne néanmoins des erreurs d'estimation supérieures, dues à la nature séquentielle du filtrage, mais leurs effets sont évalués par la matrice de covariance.

Il existe bien sûr bien d'autres extension du filtre de Kalman qu'il est impossible d'énumérer dans le cadre de cette thèse. Nous pensons aux filtres découplés (Bouchard 2001; Eaton et al. 2000; Stan et Kamen 2000), ou encore aux filtres hiérarchiques (Rousseaux et al. 1988). Une revue est présentée dans (Sayed et Kailath 1994).

3.2.4 Discussion

a Sur l'optimalité

Le filtre de Kalman est un filtre optimal. Lorsque la matrice de corrélation $\mathbf{K}_k$ n'est pas déterminée à partir de son expression théorique (2.25) mais plutôt par une approximation (pour par exemple limiter le nombre d'opérations), alors le filtre est qualifié de filtre sous-optimal (en anglais de "suboptimal filter"). D'une manière plus générale, dès lors qu'une des hypothèses posées pour dériver les équations du filtre de Kalman n'est pas respectée, ou dès lors qu'un modèle est simplifié par rapport à la réalité, alors on peut parler de filtrage sous-optimal.

L' optimalité n'existe qu'en théorie. Un KF utilise des modèles, déterministes et stochastiques, qui décrivent un comportement plus ou moins proche de celui du processus réel considéré. Les modèles déterministes sont théoriques, alors que le filtre est conçu pour résoudre des problèmes pratiques réels. Les connaissances déterministes interviennent à travers les matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, ou encore la matrice $\mathbf{D}$.

Les connaissances stochastiques sont fournies via les modèles de bruits $\mathbf{w}_k$ et $\mathbf{v}_k$, donc en posant : $\mathbf{w}_k \sim N(0,\mathbf{Q})$ et $\mathbf{v}_k \sim N(0,\mathbf{R})$. Si le modèle déterministe tente de décrire un processus physique d'une façon plus ou moins évidente, au contraire, les connaissances stochastiques permettent d'introduire de l'incertitude. Grâce à elles, le filtre est capable de se rendre compte qu'il n'est pas parfait. Ce concept va déjà bien au delà de celui de bien des modèles conventionnels. Il n'en reste pas moins que c'est à l'utilisateur de fixer les valeurs des covariances $\mathbf{Q}$ et $\mathbf{R}$, et que ceci doit être fait en connaissance de cause. Un filtre proche de l'optimalité est un filtre qui modélise au maximum les sources d'erreur.

Pour tenir compte d'un nombre maximal d'éléments dans l'étape de modélisation, la notion de sensibilité est importante. Nous n'allons pas dans cette section étudier la sensibilité du filtre de Kalman. Une telle étude a déjà été effectuée dans (Gelb 1974). Dans ce dernier ouvrage, l'auteur donne comme définition de l'analyse de la sensibilité du filtre, l'étude qui consiste à évaluer la différence entre le comportement du filtre appliqué au problème concret et celui d'un filtre optimal. Dans un langage plus mathématique, l'étude de la sensibilité revient à étudier le comportement de la covariance de l'erreur par rapport à une simplification ou une approximation (avec toujours comme référence le comportement optimal).

Le fonctionnement d'un filtre, lorsqu'il est appliqué a un problème concret, doit donc toujours se faire en rapport avec un filtre théorique, un filtre optimal.

b Sur la stabilité

Dans un environnement rude, le comportement du filtre n'est pas toujours celui qu'il devrait être. Dans l'application du filtre de Kalman à un problème concret, un comportement instable peut parfois être observé. Cette instabilité, qui peut être observée dans l'évolution de la matrice $\mathbf{K}_k$, est appelée phénomène de divergence.

S'ils restent des événements rares, plusieurs raisons peuvent néanmoins expliquer les phénomènes de divergence. Ainsi, peuvent être source de divergence :

- les erreurs de modélisation ou le fait d'utiliser un modèle du système trop simplifié (la linéarisation par exemple est une simplification du modèle qui n'est par toujours correcte) ;
- le fait de simplifier la procédure du calcul du gain pour l'alléger en nombre d'opérations ;
- le choix inadéquat des matrices de covariances ;
- le problème d'observabilité, c'est-à-dire lorsqu'on a pas assez de variables du vecteur d'état qui sont reprises dans la mesure, ou en d'autres termes lorsque trop de variables d'état sont cachées ;
- enfin les erreurs d'arrondi ou d'imprécision numérique.

Dans (Chen et Peng 1993), les auteurs tentent d'analyser le comportement et la robustesse du filtre par rapport aux incertitudes faites sur les paramètres.

Gelb (Gelb 1974) distingue deux catégories, les divergences apparentes et les divergences vraies. Dans le cas de la divergence apparente, l'erreur d'estimation est large (l'estimation est bien trop grande par rapport à l'ordre de grandeur de l'estimation théorique). Une divergence vraie se distingue par le fait que l'erreur d'estimation tend vers l'infini. La première résulte typiquement d'un comportement largement sous-optimal du filtre, alors que la seconde provient essentiellement d'un comportement instable du filtre (lorsqu'il manque des variables d'état, lorsque le modèle ne décrit pas le fonctionnement du système réel, etc.).

L'analyse théorique des phénomènes de divergence reste relativement complexe et l'ingéniosité des travaux de recherche pour les détecter ou même pour y parer se révèle assez limitée. Rappelons juste que ces techniques doivent être simples et efficaces. Les méthodes, pour prévenir les cas de divergences, consistent à inclure un processus de bruit fictif dans le modèle du système, ou à estimer des états qui ne sont pas modélisés. Cette dernière approche est souvent compliquée et difficile à mettre en œuvre : la complexité du calcul est accrue et rien n'assure la bonne prise en compte de tous les états. Pour éviter les phénomènes de divergence, on emploie également des filtres qui ne tiennent pas compte des anciennes valeurs par le biais de pondérations, ou des filtres qui effectuent la tâche de filtrage sur une fenêtre d'apodisation (Jazwinsky 1968). Cette dernière méthode est basée sur le raisonnement qui considère que les données les plus anciennes n'appartiennent plus au contexte actuelle de filtrage. Elle est sans doute la méthode la plus facile à mettre en œuvre, mais en contrepartie est la plus empirique ((Gelb 1974) page 280).

Ce principe trouve des applications très concrètes. Pour limiter les phénomènes de divergence d'un filtre utilisé pour estimer les grandeurs d'une machine à papier (Xia et al. 1994), la mémoire du filtre est limitée dans le temps. Dans cette étude, un critère de performance est utilisé pour mesurer l'optimalité du filtre. Ce filtre est dénommé adaptatif (très précisément AFKF, pour "Adaptive Fading Kalman Filter") en raison de son facteur d'oubli, qui est adapté en ligne via le critère de performance et grâce aux observations. Cette réalisation se base sur les études faites dans (Fagin 1964) et dans (Sorenson et Sacks 1971) qui ont toutes deux montré qu'un système linéaire n'est valide que pour une période de temps bien limitée, et qu'il n'est par conséquent pas adéquat sur de longues périodes de temps. Ils ont donc suggéré de limiter la mémoire du filtre, en pondérant les données passées par un facteur d'oubli décroissant exponentiellement (d'où le terme "fading").

Nous avons vu que les phénomènes de divergence résultaient souvent d'une erreur de modélisation, soit dans la conception du filtre, soit dans sa réalisation. Une autre source de divergence, bien plus rare celle-ci, apparaît lorsque les valeurs des matrices $\mathbf{P}_k$ et $\mathbf{K}_k$ deviennent anormalement petites. Le filtre ne peut plus alors tenir compte des nouvelles mesures pour se corriger. Les études qui se sont déjà consacrées à ce problème se trouvent dans (Fitzgerald 1967; Fitzgerald 1971; Price 1968). Une analyse des divergences est également présentée dans (Bryson 1978).

c Sur l'initialisation

Il faut deux choses pour mettre en œuvre un filtre de Kalman : un espace d'état connu et une estimation initiale du vecteur d'état avec la covariance associée à cet état. Le choix des valeurs initiales n'est pas toujours évident. Ces informations ne sont pas forcément disponibles et plus les valeurs initiales sont proches de la réalité, plus le filtre converge rapidement, c'est-à-dire fournit des estimations correctes.

Lorsque l'environnement est stationnaire, une initialisation courante consiste à utiliser la moyenne et la covariance inconditionnelle de l'état initial ($\hat{\mathbf{x}}_0 = \overline{\mathbf{x}}_0$ et $\mathbf{P}_0 = \mathbf{Q}_0$). Lorsque l'environnement est non stationnaire, cette approche n'est plus valable dans la mesure où ces deux moments dépendent du temps et ne sont de ce fait valables que pendant une certaine période.

Il existe plusieurs manières empiriques pour initialiser un filtre de Kalman lorsque celui-ci est placé dans un environnement non stationnaire (Hillmer et Bell 1987). On peut :

- choisir une variance élevée $\mathbf{P}_0$, comme préconisé dans (Ansley et Kohn 1985) ;
- utiliser un "information filter" avec comme matrice inverse de la variance de l'état initial, une matrice nulle (Anderson et Moore 1979) ;
- augmenter le vecteur d'état par les variables différentiées avec *d* observations (lorsque la différentiation est d'ordre *d*) et initialiser le filtre au temps *d* quand les variables différenciées seront complètement connues ;
- ou encore utiliser un filtre avec une estimation initiale et sa covariance nulle.

Toutes ces approches sont difficiles à justifier et entraînent des problèmes d'implémentation. La première, lorsque la variance est élevée, donne des difficultés numériques et la convergence n'est pas assurée. L' "information filter" ne s'applique pas à tous les problèmes. L'augmentation du vecteur d'état n'est pas toujours triviale, en plus elle oblige à utiliser des vecteurs d'états avec un nombre élevé de variables qui alourdissent les procédures de calculs. La dernière approche est très restrictive, et elle ne se justifie que partiellement.

Parmi les méthodes justifiées, on peut compter :

- l'approche par transformation de Kohn et Ansley (Kohn et Ansley 1986) ;
- l'approche DKF appelée en anglais "Diffuse Kalman Filter" (Jong 1991) ;
- l'initialisation de Bell et Hillmer (Bell et Hillmer 1991) ;
- l'initialisation de Gomez et Maravall (Gómez et Maravall 1994).

L'initialisation du filtre de Kalman pour un environnement non stationnaire proposée dans (Kohn et Ansley 1986) est une approche rigoureuse. Ce filtre de Kalman modifié autorise une valeur infinie pour la variance de certaines variables de l'espace d'état. Les auteurs justifient leur démarche en montrant qu'elle produit les mêmes résultats qu'une approche par transformation, qui élimine les effets dus à la non stationnarité des valeurs initiales. La complexité de cette initialisation reste un inconvénient majeur.

La dernière méthode, appelée EKF, est basée sur une distribution définie conditionnellement sur des observations préalables (plus précisément, sur le logarithme d'une vraisemblance). Son inconvénient majeur provient des longues procédures de calculs matriciels. Le DKF est un cas particulier de l'EKF, il est donc plus pratique à mettre en œuvre.

L'initialisation de Bell et Hillmer est également basée sur le logarithme d'une vraisemblance mais au lieu d'utiliser les données observées sur une période préalable bien définie, elle emploie

l'approche par transformation. Cette approche est très légère au niveau calculatoire, mais elle est aussi dépendante des particularités du système considéré.

D'autres méthodes sont énumérées dans (Gómez et Maravall 1993). Les méthodes empiriques sont les plus simples à mettre en œuvre. Comme elles ne respectent pas forcément les hypothèses faites sur le filtre, elles ont tendance à diverger rapidement et sont donc peu robustes. En général, lorsque la distribution n'est pas connue, on prend par défaut des distributions uniformes. Le maximum d'entropie ou l'estimation de Fisher semblent être à ce jour les techniques les plus robustes pour initialiser un filtre de Kalman.

Une extension de l'initialisation de Fisher est proposée dans (Catlin 1993). L'auteur propose une estimation de Fisher pour fixer les valeurs des grandeurs a priori de la phase d'initialisation du filtre de Kalman, lorsque le vecteur d'état initial n'est pas connu ou ne peut pas être estimé, et lorsque les hypothèses effectuées sur le processus de mesure ne sont pas toutes respectées. Cette méthode requiert un certains nombre de mesures préalables.

3.3 Le filtrage de Kalman adaptatif

Parmi les techniques de traitement des systèmes et des signaux, le filtrage adaptatif est certainement le plus utilisé pour sa capacité à traiter un environnement non stationnaire. Issu des méthodes de programmation mathématique et d'estimation paramétrique, son évolution s'est accélérée ces dernières décennies avec l'apparition des technologies numériques rapides.

L'estimation par le filtrage de Kalman a été formulée avec une connaissance parfaite des propriétés du système dynamique. Cependant, il existe toujours un certain degré d'incertitude et de méconnaissance. Nous avons vu qu'un filtre utilisant des valeurs initiales différentes des valeurs réelles conduit à un comportement non optimal, ou éventuellement à des divergences. Les performances peuvent être améliorées si, en plus de l'estimation des variables d'état, certains paramètres du filtre sont mis à jours simultanément pour constituer un filtre adaptatif. Un filtre adaptatif peut présenter, dans une certaine limite, un comportement optimal. Il faut bien sûr se rendre à l'évidence, plus les informations sur le système sont disponibles, et plus le comportement du filtre est proche de l'optimalité. Le manque de connaissance se paie.

Plusieurs sens peuvent être distingués dans l'adaptation des paramètres d'un filtre de Kalman (Stengel 1986). Le filtre peut être adaptatif au sens des paramètres, au sens du bruit, ou au sens de la multiplicité des modèles. A ses trois sens étaient associées trois techniques : l'adaptation au sens des paramètres était réalisée par une augmentation du vecteur d'état qui nécessite beaucoup de connaissances sur les paramètres à estimer, l'adaptation au sens du bruit passait par évaluation des propriétés statistiques de l'innovation, et l'adaptation au sens de la multiplicité des modèles se faisait par un banc de filtres dont la principale difficulté de réalisation était le choix du filtre.

Pendant longtemps, l'augmentation du vecteur d'état et les nombreux inconvénients induits n'admettaient aucune alternative. Ne pas disposer d'un modèle suffisant était une difficulté quasi-insurmontable. Très peu de travaux s'en préoccupaient. De nombreux auteurs proposaient d'utiliser des matrices qui dépendent du temps, mais sans définir une approche complète (Savkin et Petersen 1996; Savkin et Petersen 1998). Comme les approches à modèle multiple étaient des applications irréalistes, les seules méthodes réellement employées étaient les méthodes par correction de covariance, de sorte que parler d'AKF revenait systématiquement à faire de l'estimation de bruit – appelée ainsi à cause de l'identification des propriétés statistiques de l'innovation – et à prendre comme référence le filtre optimal qui implique un bruit blanc en tant qu'innovation.

Cette association n'est aujourd'hui plus aussi franche. De nouvelles méthodes sont venues enrichir la palette des outils existants, toutes essentiellement issues de la théorie des statistiques. A l'augmentation du vecteur d'état, aux méthodes de correction des covariances et aux approches à modèle multiple, se sont rajoutées une méthode d'estimation des paramètres par l'algorithme EM

("Expectation Maximization"), ainsi que de nouveaux concepts qui méritent le détour, comme les filtres particulaires, le filtre dit sans odeur (UKF, "Unscented Kalman Filter" en anglais), ou les filtres adaptatifs temporels.

3.3.1 Estimation des paramètres par augmentation du vecteur d'état

L'idée simple de l'augmentation du vecteur d'état consiste à utiliser l'estimateur du filtre de Kalman pour estimer, en plus du vecteur d'état, ses propres paramètres. Les grandeurs à évaluer sont alors intégrées dans un vecteur d'état augmenté $\mathbf{x}_{augm,k} = [\mathbf{x}_k \quad \mathbf{p}_k]^T$. De cette façon, à chaque itération, le vecteur des paramètres $\mathbf{p}_k$ du filtre est mis à jour en même temps que les variables d'état $\mathbf{x}_k$.

Si l'idée est simple, elle n'est par contre pas très exploitable. En fait, cette manière de procéder requiert de bien connaître les paramètres à estimer. Dans le cas général et non linéaire, il faut non seulement une fonction de transition augmentée pour faire évoluer le vecteur paramètre dans le temps,

$$\mathbf{x}_{augm,\,k+1} = \mathbf{f}_{augm}(\mathbf{x}_{augm,\,k}, \mathbf{u}_k, \mathbf{w}_k)\,, \tag{2.48}$$

mais il faut également connaître les valeurs initiales $\mathbf{x}_0$ des variables d'état, les valeurs initiales $\mathbf{p}_0$ des paramètres, et il faut surtout connaître la matrice de covariance augmentée.

En raison de ces inconvénients, cette méthode est très peu utilisée.

3.3.2 Méthodes par correction de covariance

Les matrices de covariance $\mathbf{Q}_k$ et $\mathbf{R}_k$ sont les deux grandeurs qui caractérisent respectivement les bruits $\mathbf{w}_k$ et $\mathbf{v}_k$, puisque ces bruits sont Gaussiens et centrés. On peut néanmoins considérer que les propriétés des bruits ne sont pas forcément constantes et qu'elles peuvent évoluer dans le temps. Le filtre doit pouvoir en tenir compte pour obtenir des performances meilleures.

Fort de ce principe, dans les expressions (2.24) et (2.25) qui permettent de calculer la matrice de corrélation $\mathbf{K}_k$, les covariances $\mathbf{Q}_k$ et $\mathbf{R}_k$ sont volontairement indexées dans l'espace temporel. La pondération $\mathbf{K}_k$ dépend directement de ces deux grandeurs. Notons juste que, si ces deux covariances évoluent dans le temps, il est impossible de prédéterminer une valeur de la matrice $\mathbf{K}_k$, donc de calculer un estimateur $\alpha-\beta$, ou un filtre linéarisé.

a Méthodes empiriques

Pour compenser la dérive de la prise de mesure (dans un environnement non stationnaire), on peut vouloir ajuster la covariance du bruit $\mathbf{v}_k$. La motivation la plus simple consiste à vouloir lui assigner une valeur plus significative que celle qui est utilisée à l'initiation du filtre, et qui n'est peut être plus représentative après un certain temps. Dans son rapport (Kelly 1994), Kelly propose par exemple d'ajuster $\mathbf{R}_k$ en fonction de la vitesse des signaux.

Tandis que ce raisonnement peut sembler relativement correct pour la covariance $\mathbf{R}_k$, la même démarche n'est pas transposable au bruit $\mathbf{w}_k$ et à sa covariance $\mathbf{Q}_k$ (Franklin et al. 1998). Ce bruit en effet, est un artifice mathématique et il possède une représentation moins évidente que celle du bruit de mesure. La compensation de ce bruit lorsqu'il est corrélé dans le temps – il s'agit alors d'un bruit coloré – peut se faire en en tenant compte dans le modèle d'état et en effectuant une augmentation du vecteur d'état. Si cette démarche reste simple, elle est néanmoins coûteuse en calculs. La méthode pratique consiste à conserver un modèle de bruit non corrélé et à procéder par essai/erreur afin de trouver des coefficients satisfaisants pour la matrice $\mathbf{D}$.

Utiliser, pour la matrice $\mathbf{Q}_k$, une matrice nulle peut entraîner une valeur nulle de la matrice de gain. Ceci a théoriquement pour conséquence un filtre qui ne sait plus tenir compte des observations faites, puisque $\mathbf{K}_k$ pondère le terme d'innovation, et conduit alors à un filtre inefficace. Dans la pratique, où apparaissent toujours des erreurs et où se produisent toujours des imperfections, la conséquence est un filtre divergent. Sans connaissances a priori, $\mathbf{Q}_k$ et $\mathbf{D}$ sont réglés empiriquement.

L'expérience des auteurs de (Franklin et al. 1998) leur dicte d'utiliser un historique de $\mathbf{P}_k$ comme indication. Ils ne proposent cependant pas de méthode précise.

Généralement, $\mathbf{Q}_k$ et $\mathbf{R}_k$ sont initialisées à des valeurs non nulles, puis baissent dans le temps, le plus souvent selon une loi exponentielle. Dans son rapport (Kelly 1994), Kelly propose d'utiliser la vitesse des signaux pour faire varier $\mathbf{Q}_k$ et $\mathbf{R}_k$ sans donner de justification. Si $\mathbf{Q}_k$ et $\mathbf{R}_k$ sont choisis empiriquement, le sens physique de $\mathbf{P}$ n'est plus correct, et si $\mathbf{Q}_k$ n'est pas nulle, la matrice $\mathbf{P}$ sera forcée de croître.

Pour résumer, le réglage empirique des covariances $\mathbf{Q}_k$ et $\mathbf{R}_k$ n'est sans doute pas la meilleure des solutions. La matrice $\mathbf{P}$ perd son sens physique et les phénomènes de divergence ne sont plus maîtrisés.

b Méthodes basées sur le principe d'optimalité du filtre

Des méthodes rigoureuses et robustes, visant à l'estimation des matrices de bruits $\mathbf{Q}_k$ et $\mathbf{R}_k$ en supposant tous les autres paramètres connus, se fondent sur l'observation suivante : si le filtre est optimal, la séquence d'innovation $\hat{\mathbf{s}}_{k+1|k}$ est un bruit blanc Gaussien. Ici l'hypothèse Gaussienne des bruits $\mathbf{w}_k$ et $\mathbf{v}_k$ est nécessaire. La séquence d'innovation doit de surcroît être stationnaire et ergodique. A chaque fois que le filtre ne satisfait plus cette hypothèse, les estimations de $\mathbf{Q}_k$ et $\mathbf{R}_k$ sont modifiées. Ces méthodes partent du principe d'optimalité du filtre : la séquence d'innovation doit alors être blanche et l'expression du gain $\mathbf{K}_k$ a été calculée de façon à remplir cette condition (Gelb 1974).

Les premiers travaux à utiliser la séquence d'innovation comme critère d'optimalité du filtre se trouvent dans (Mehra 1970). Toutes ces méthodes ont été ordonnancées en quatre classes dans (Mehra 1972) qui contient une bibliographie assez complète. Une revue très détaillée est disponible dans (Moghaddamjoo 1989) et une plus récente se trouve dans (Moghaddamjoo et Kirlin 1993). Parmi les quatre classes de méthodes, il y a :

- celles qui utilise l'estimation Bayésienne,
- celles qui utilisent le maximum de vraisemblance,
- les méthodes de corrélation (dont le filtre de Jazwinsky (Jazwinsky 1969) et celui de Belanger (Belanger 1974)),
- et les méthodes d'identification de covariance.

Le principe de ces méthodes consiste à compenser les erreurs de modélisation par les bruits. Elles comportent un certain nombre d'inconvénients, et non des moindres. Ce sont des méthodes itératives qui ne concernent que les environnements stationnaires et des statistiques qui évoluent doucement. Elles sont coûteuses en calculs, et souvent inadéquates pour des tâches d'estimation en ligne. Le test de blancheur (sur la moyenne et la variance) ne sert à estimer que $\mathbf{Q}_k$ et $\mathbf{R}_k$; $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$s et $\mathbf{D}$ doivent être des paramètres connus et constants. Si ces paramètres ne sont pas tous connus, le test de blancheur n'est plus un critère suffisant.

Les méthodes de correction de covariance ont été étendues aux bruits non Gaussiens (Kirlin et Moghaddamjoo 1986a; Kirlin et Moghaddamjoo 1986b), ainsi qu'à la détection et la compensation automatique d'entrées déterministes (Moghaddamjoo et Kirlin 1989). Les contraintes restent cependant les mêmes.

3.3.3 Les filtres à modèle multiple

L'initiateur de cette approche est Magill (Magill 1965) qui propose l'idée suivante : considérer le système comme décrit par un paramètre qui peut prendre plusieurs valeurs.

L'estimation est conséquemment construite en tenant compte de chaque valeur hypothétique que peut prendre ce paramètre à travers plusieurs filtre et, celui qui répond le mieux est celui qui utilise la valeur la plus juste du paramètre. Ce raisonnement peut être généralisé en proposant plusieurs

modèles au sein d'une même structure pour réaliser la tâche d'estimation ou de filtrage. Ce raisonnement reprend le principe des bancs de filtres ou le concept de filtres en parallèle.

Dans l'approche du filtre de Kalman à modèle multiple, les hypothèses peuvent être placées sur les valeurs de $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, ainsi que sur celles de $\mathbf{Q}_k$ et $\mathbf{R}_k$, et a fortiori sur les conditions initiales $(\mathbf{x}_0, \mathbf{P}_0)$.

Le succès de cette approche dépend des conditions initiales, mais surtout du critère utilisé pour le choix des hypothèses. Un certain nombre de travaux (Hanlon et Maybeck 1998) suggère l'utilisation des propriétés statistiques de la séquence d'innovation mais, ni le test de blancheur, ni une méthode de comparaison de la matrice de covariance du résidu, ne semblent être suffisamment précis pour effectuer un choix judicieux entre plusieurs modèles. Une approche supérieure en performance consiste à utiliser les deux premiers moments pour estimer la vraisemblance que l'hypothèse associée à un modèle est la meilleure. L'estimation optimale est alors la somme des estimations fournies par les différents filtres pondérées par leur probabilité. L'estimation récursive des probabilités des estimations est réalisée via la règle de Bayes.

Pour I vecteurs de paramètres $\mathbf{p}$ de dimension l, à l'instant discrétisé k, la probabilité conditionnelle du paramètre j s'écrit :

$$p(\mathbf{p}_j|\mathbf{z}_k) = \frac{p(\mathbf{z}_k|\mathbf{p}_j)p(\mathbf{p}_j)_{k-1}}{\sum_{i=1}^{I}\left(p(\mathbf{z}_k|\mathbf{p}_i)p(\mathbf{p}_i)_{k-1}\right)}. \tag{2.49}$$

Vraisemblablement, la probabilité d'avoir obtenu la mesure à l'instant k-1 est de un, d'où

$$p(\mathbf{p}_j)_{k-1} = p(\mathbf{p}_j|\mathbf{z}_{k-1}). \tag{2.50}$$

A partir d'une valeur initiale $p(\mathbf{p}_j|0)$, $1 \le j \le I$, l'expression (2.49) constitue une formulation récursive de la probabilité du vecteur paramètres $\mathbf{p}_j$ conditionnée par l'observation $\mathbf{z}_k$.

On cherche $p(\mathbf{p}_j|\mathbf{z}_k)$. Pour cela il faut dans un premier lieu exprimer $p(\mathbf{z}_k|\mathbf{p})$:

$$\begin{aligned} p(\mathbf{z}_k|\mathbf{p}) &= p\left(\mathbf{z}_k|\mathbf{x}_k(\mathbf{p})\right) \\ &= \frac{1}{(2\pi)^{n/2}|\mathbf{R}_k|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{z}_k - \mathbf{C}\mathbf{x}_k)^T\mathbf{R}_k^{-1}(\mathbf{z}_k - \mathbf{C}\mathbf{x}_k)\right) \\ &= \frac{1}{(2\pi)^{n/2}|\mathbf{R}_k|^{1/2}}\exp\left(-\frac{1}{2}\mathbf{v}_k^T\mathbf{R}_k^{-1}\mathbf{v}_k\right). \end{aligned} \tag{2.51}$$

Le vecteur d'état vrai n'est cependant pas connu. On cherchera alors plutôt à estimer la probabilité conditionnelle suivante $p\left(\mathbf{z}_k|\hat{\mathbf{x}}_k(\mathbf{p})\right)$:

$$p\left(\mathbf{z}_k|\hat{\mathbf{x}}_k(\mathbf{p})\right) = \frac{1}{(2\pi)^{n/2}|\mathbf{S}_k|^{1/2}}\exp\left(-\frac{1}{2}\hat{\mathbf{s}}_{k|k-1}^T\mathbf{S}_k^{-1}\hat{\mathbf{s}}_{k|k-1}\right). \tag{2.52}$$

$\hat{\mathbf{s}}_{k|k-1}$ est le terme d'innovation, auquel est associé la matrice de covariance $\mathbf{S}_k$:

$$\mathbf{S}_k = \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + \mathbf{R}_k. \tag{2.53}$$

Ainsi, la densité de probabilité associée à chaque observation basée sur une hypothèse est estimée par :

$$p\left(\mathbf{z}_k|\hat{\mathbf{x}}_k(\mathbf{p}_i)\right) = \frac{1}{(2\pi)^{n/2}|\mathbf{S}_k(\mathbf{p}_i)|^{1/2}}\exp\left(-\frac{1}{2}\hat{\mathbf{s}}_{k|k-1}^T(\mathbf{p}_i)\mathbf{S}_k^{-1}(\mathbf{p}_i)\hat{\mathbf{s}}_{k|k-1}(\mathbf{p}_i)\right), \tag{2.54}$$

avec l'innovation et la covariance associée pour chaque hypothèse i=1,…, I du vecteur paramètres $\mathbf{p}_i$:

- $\hat{\mathbf{s}}_{k|k-1}(\mathbf{p}_i) = \mathbf{z}_k - \mathbf{C}\hat{\mathbf{x}}_{k|k-1}(\mathbf{p}_i)$,
- $\mathbf{S}_k(\mathbf{p}_i) = \mathbf{C}\mathbf{P}_k(\mathbf{p}_i)\mathbf{C}^T + \mathbf{R}_k$.

La probabilité conditionnelle liée au vecteur $\mathbf{p}_j$, $1 \le j \le I$, se reformule ainsi :

$$p(\mathbf{p}_j|\mathbf{z}_k) = \frac{p\left(\mathbf{z}_k|\hat{\mathbf{x}}_k(\mathbf{p}_j)\right)p\left(\mathbf{p}_j|\mathbf{z}_{k-1}\right)}{\sum_{i=1}^{I}\left(p\left(\mathbf{z}_k|\hat{\mathbf{x}}_k(\mathbf{p}_i)\right)p\left(\mathbf{p}_i|\mathbf{z}_{k-1}\right)\right)}. \tag{2.55}$$

L'estimation est donnée par la somme pondérée des estimations :

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=1}^{I}\left(p(\mathbf{p}_i|\mathbf{z}_k)\hat{\mathbf{x}}_{k|k-1}(\mathbf{p}_i)\right). \tag{2.56}$$

Dans sa publication, Magill (Magill 1965) valide son approche sur un système simulé. Il existe des applications réelles du principe qu'il a exposé. Un filtre de Kalman à modèle multiple pour du suivi de cible est par exemple présenté dans (Bar-Shalom et Birmiwal 1983).

L'application directe de l'approche à modèle multiple concerne la fusion de données. Dans (Drolet et al. 2000) par exemple, le positionnement d'un véhicule sous-marin est effectué à l'aide d'un estimateur à modèle multiple qui se justifie dans cette application, par le fait que parmi les capteurs utilisés certains ne fournissent pas de mesure à toutes les itérations. Un filtre de Kalman seul ne permet d'utiliser qu'un unique modèle du système. Les capteurs en cours d'utilisation valident le modèle appropriés parmi les différents modèles (sept en tout) qui sont disponibles.

Au fur et à mesure que les domaines d'applications se multiplient et que les performances des calculateurs croissent, l'usage des approches à modèle multiple en traitement du signal se répand dans des disciplines aussi variées que la compression d'images, le codage, la vidéo numérique, l'estimation et l'analyse du signal en temps-fréquence. Maintenant que la théorie des bancs de filtres arrive à maturité (Vaidyanathan 1990; Vaidyanathan 1993), et afin de la mettre en œuvre pour des applications en temps réel, des algorithmes adaptatifs pour l'optimisation des bancs de filtres doivent être développés.

3.3.4 Adaptation avec l'algorithme EM

a Principe de l'algorithme EM

L'algorithme EM ("Expectation Maximization algorithm") réalise une estimation au sens du maximum de vraisemblance lorsque certaines variables d'un système ne sont pas disponibles. C'est un algorithme itératif d'estimation de paramètres, partant de choix initiaux (Dempster et al. 1977). Chaque itération est composée de deux étapes :

- une étape E, qui détermine la distribution des variables non observables pour les valeurs connues des variables mesurables, et pour l'estimation courante connue ;
- et une étape M, qui recalcule l'estimation comme étant celle qui maximise la vraisemblance, sous l'hypothèse que la distribution déterminée durant l'étape E soit correcte.

Soit $\mathbf{r}$ un vecteur complet, constitué des données non observables et de données observables d. On peut alors considérer deux espaces $\mathcal{R}$ et $\mathcal{D}$ ainsi qu'une fonction non bijective (seulement surjective) de $\mathcal{R}$ vers $\mathcal{D}$ les reliant. Seulement, au lieu d'avoir accès au vecteur $\mathbf{r}$ complet, seul $d = d(\mathbf{r})$ est observable dans $\mathcal{D}$.

Si $f_c(\mathbf{r}|\boldsymbol{\theta})$ est la densité de probabilité conditionnelle de $\mathbf{r}$, pour un vecteur paramètre $\boldsymbol{\theta}$ donné, il s'ensuit que la densité de probabilité de la variable aléatoire D, pour $\boldsymbol{\theta}$ donné, est définie par :

$$f_D(d|\boldsymbol{\theta}) = \int_{\mathcal{R}(d)} f_c(\mathbf{r}|\boldsymbol{\theta})d\mathbf{r}, \tag{2.57}$$

où d est une réalisation de D et $\mathcal{R}(d)$ est un sous-espace de $\mathcal{R}$ déterminé par $d = d(\mathbf{r})$. L'algorithme EM se doit de chercher le vecteur paramètres $\boldsymbol{\theta}$ qui maximise la fonction de vraisemblance des données incomplètes (le logarithme en fait) :

$$\mathcal{L}(\boldsymbol{\theta}) = \log f_D(d|\boldsymbol{\theta}). \tag{2.58}$$

Ce problème est résolu de manière indirecte, en utilisant de façon itérative le logarithme de la fonction de vraisemblance des données complètes :

$$\mathcal{L}_c(\boldsymbol{\theta}) = \log f_c(\mathbf{r}|\boldsymbol{\theta}) , \tag{2.59}$$

qui est une variable aléatoire, puisque les données non observables ne sont pas connues. Pour être plus précis, si $\hat{\boldsymbol{\theta}}_k$ représente le vecteur estimé par l'algorithme EM des paramètres à l'itération *k*, durant l'étape E l'espérance mathématique suivante est calculée par rapport à $\hat{\boldsymbol{\theta}}_k$:

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_k) = E[\mathcal{L}_c(\boldsymbol{\theta})] . \tag{2.60}$$

Au cours de l'étape M de la même itération, $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_k)$ est maximisé par rapport à $\boldsymbol{\theta}$ (on cherche le vecteur paramètre $\boldsymbol{\theta}$ qui donne la valeur la plus grande de $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_k)$ parmi tout l'ensemble des $\boldsymbol{\theta}$ possibles). La mise à jour se fait alors selon :

$$\hat{\boldsymbol{\theta}}_{k+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_k) . \tag{2.61}$$

L'algorithme débute avec des paramètres initiaux $\boldsymbol{\theta}_0$. Les étapes E et M de l'algorithme sont ensuite répétées itérativement, à travers respectivement les expressions (2.60) et (2.61), jusqu'à obtenir une valeur arbitrairement faible pour la différence $\mathcal{L}(\hat{\boldsymbol{\theta}}_{k+1}) - \mathcal{L}(\hat{\boldsymbol{\theta}}_k)$.

L'algorithme EM est un estimateur au sens du maximum de vraisemblance qui sied idéalement à la détermination des paramètres d'un mélange de densités de probabilité (Hofmann et Puzicha 1998). Il est donc logiquement utilisé dans le concept d'apprentissage des Mélanges d'Experts hiérarchiques (Haykin 1999, page 381).

L'algorithme EM peut réaliser un estimateur Bayésien, au sens du maximum a posteriori (MAP), en incluant des informations a priori sur le vecteur des paramètres $\boldsymbol{\theta}$ (Haykin 1999). Lorsqu'à l'étape M, au lieu de maximiser $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_k)$, on choisit une valeur de $\hat{\boldsymbol{\theta}}_{k+1}$ qui augmente la valeur de la fonction à optimiser, on parle d'algorithme GEM ("Generalized EM") (Fiordaliso 1999, page 101).

b Algorithme EM et estimation de paramètres

La première utilisation de l'algorithme EM pour l'estimation des paramètres d'un système dynamique linéaire a été présentée dans (Shumway et Stoffer 1982). La partie E de la méthode se base sur les probabilités a posteriori estimées par le filtre de Kalman lissé. Cependant, la technique proposée suppose que la matrice d'observation $\mathbf{C}$ soit connue. Ce travail a ensuite été étendu au cas général par (Ghahramani et Hinton 1996). Dans ce dernier travail, l'étape E est toujours composée par le KF lissé alors que la phase M permet l'estimation des matrices $\mathbf{C}$, $\mathbf{A}$, $\mathbf{R}$, et $\mathbf{Q}$ qui sont alors des matrices supposées constantes pour toute la séquence d'observation.

L'utilisation de l'algorithme EM pour l'estimation des paramètres des filtres de Kalman, étendus ou non, est aujourd'hui une technique rodée (Freitas et al. 1998a; Roweis et Ghahramani 1999). Elle est même une approche essentielle de l'apprentissage avec le KF et le EKF. Dans (Freitas et al. 1998b) est fournie une approche typique où la règle d'adaptation des poids d'un réseau MLP est réalisée par un filtre de Kalman étendu, dont les paramètres sont estimés via l'algorithme EM. L'avantage de cette approche par rapport à la règle BP, vient du fait qu'elle tient compte du moment statistique d'ordre deux. Leur approche est validée sur l'apprentissage d'une transformation robotique (un robot deux axes), et sur un problème de classification de données médicales.

Dans (Freitas et al. 1997), les auteurs s'intéressent à la implémente la détection automatique de pertinence (ARD, "Automatic Relevance Determination"). Les auteurs distinguent notamment trois niveaux d'inférence dans la hiérarchie de la modélisation : la sélection du modèle, l'estimation de paramètres et l'estimation du bruit. L'estimation des paramètres se fait par EKF.

Lee et Jung (Lee et Jung 2000) s'intéressent au problème du débruitage et à l'amélioration de signaux auditifs (la voix, un signal non stationnaire). L'algorithme EM est utilisé pour estimer le signal et identifier le système. Dans la phase E, l'estimation du signal provient d'une somme pondérée de moyennes conditionnelles issues d'un filtrage de Kalman étendu à modèle multiple. Les pondérations

sont faites par les probabilités conditionnelles a posteriori des états pour le signal bruité donné, qui sont calculées par l'algorithme de Viterbi. Dans la phase M, les paramètres des bruits sont estimés par un algorithme adaptatif modifié basé sur une descente de gradient.

Dans (Strandlie et Zerubia 1999) l'algorithme PMHT ("Probabilistic Multi-Hypothesis Tracking"), un algorithme pour suivre la trajectoire de particules dans des détecteurs utilisés en physique des hautes énergies, est présenté. Il est fondé sur l'estimateur du maximum de vraisemblance, et s'appuie sur un algorithme de type EM. L'algorithme résultant correspond à l'utilisation en parallèle de plusieurs filtres de Kalman itératifs couplés. Par rapport à l'approche originale, plusieurs modifications ont été apportées. En particulier, les probabilités reliant les événements élémentaires aux trajectoires ont été transformées afin d'obtenir une compétition entre ces événements dans une même couche du détecteur.

Il existe de nombreuses autres applications, dont la modélisation de séries temporelles non linéaires (Tresp et al. 1999). Le lien vers les filtres particulaires semble assez logique.

3.3.5 Les filtres particulaires

La recherche en filtrage non linéaire est fortement motivée par ses nombreuses applications. Cependant, il s'agit d'un problème qui n'admet en général aucune solution de dimension finie. Avec le filtre de Kalman étendu et sensible aux non linéarités du système, et les méthodes numériques de maillage fortement dépendantes de la dimension de l'espace, les méthodes particulaires semblent a priori des méthodes particulièrement adaptées à ce problème. Elles proposent d'approcher la loi conditionnelle recherchée par une mesure empirique et par un système de particules aléatoires évoluant avec le signal et les observations. Néanmoins, les méthodes particulaires classiques présentent elles aussi des faiblesses qui les rendent parfois inefficaces.

a Concept

L'estimation d'un point de vue concret est souvent confrontés à la nature non Gaussienne des systèmes, à la non linéarité et à la non stationnarité. La conséquence directe qui en ressort, est l'impossibilité de mettre au point des modèles suffisants, c'est-à-dire sachant tenir compte de toutes ces complexités. Les critères standards, tel le maximum de vraisemblance, le maximum a posteriori ou l'estimateur des moindres carrés, ne sont plus valides dans un tel environnement.

Les méthodes sous-optimales tel que le filtre de Kalman étendu ou telles que les méthodes basées sur des sommes de Gaussiennes ("Gaussian sum approximations"), s'appliquent relativement facilement d'un point de vue pratique. Elle ne savent cependant pas tenir compte des transformations subies par les variables aléatoires inhérentes à ce type d'environnement et aboutissent à des performances moindres.

Les méthodes de Monte Carlo se posent alors comme une alternative a priori séduisante. Ce type de méthodes est basé sur les récentes avancées en statistiques appliquées et profitent surtout largement des augmentations de la puissance des derniers calculateurs. Elles sont intéressantes du fait qu'elles ne nécessitent au départ aucune hypothèse restrictive sur la probabilité des distributions des données. D'un point de vue Bayésien, les méthodes séquentielles de Monte Carlo permettent de déterminer la distribution de probabilité a posteriori d'une variable. Elles s'emploient également dans le contexte du maximum de vraisemblance.

Les filtres particulaires (en anglais "Particle Filter") sont des méthodes séquentielles de Monte Carlo. Ils permettent une représentation complète de la distribution a priori des états, de telle sorte que les estimations statistiques puissent être calculées facilement. Forts de ces propriétés, les filtres particulaires peuvent ainsi faire face à toutes sortes de non linéarités et toutes sortes de distributions. Ils sont efficaces dans des environnements non stationnaires et trouvent rapidement des applications

concrètes très convaincantes. Un certain nombre de ces filtres est récapitulé en annexe de (Oudjane 2000). La thèse de Oudjane explore d'ailleurs la stabilité de ce type de filtre.

b Principe du filtre optimal

L'objectif fixé ici, consiste à estimer à l'instant courant, la valeur d'un processus aléatoire (2.62) indirectement lié à un processus d'observation (2.63). En fait, c'est une autre façon de considérer le modèle donné par les expressions (2.34) et (2.35).

$$X_k = F_{\theta_k}(X_{k-1}, W_k)\,, \tag{2.62}$$

$$Y_k = H_{\theta_k}(X_k) + V_k\,. \tag{2.63}$$

θ_k est une chaîne de Markov homogène qui détermine à chaque instant le modèle dynamique du processus et celui de l'observation. θ_k est supposé indépendant de X_k sachant θ_{k-1}. Les distributions de θ_0 et de X_0 sont supposées connues ainsi que les lois de transition $\mathcal{L}(\theta_k | \theta_{k-1})$ et $\mathcal{L}(X_k | \theta_k, X_{k-1})$. Le filtrage particulaire consiste à simuler ces lois.

A chaque instant, on cherche donc à approcher la probabilité conditionnelle de l'état X_k, sachant les mesures passées $(Y_1, Y_2, \ldots, Y_k) = Y_1^k$, c'est-à-dire la suite des lois conditionnelles

$$\left(\mathcal{L}(X_k | Y_1^k)\right)_{k \geq 1}\,. \tag{2.64}$$

La principale difficulté réside dans le fait que le modèle, caractérisé par le vecteur paramètre θ_k, est a priori inconnu. Pour palier à ce problème, un vecteur augmenté, $E_k = (X_k, \theta_k)$, est introduit pour estimer itérativement les états et les paramètres du modèle.

La loi conditionnelle $\mathcal{L}(X_k | Y_1^k)$ se déduit alors de $\mathcal{L}(E_k | Y_1^k)$ puisque :

$$p(X_k | Y_1^k) = \sum_{k=1}^{M} p(X_k, \theta_k = t | Y_1^k)\,. \tag{2.65}$$

Entre deux instants d'observation, l'évolution de $\mathcal{L}(E_k | Y_1^k)$ se déroule en deux étapes :

- la prédiction, qui consiste à mettre à jour $\mathcal{L}(E_k | Y_1^{k-1})$ à partir de $\mathcal{L}(E_{k-1} | Y_1^{k-1})$,

$$p(E_k | E_{k-1}) = p(X_k | \theta_k, X_{k-1}) p(\theta_k | \theta_{k-1})\,, \tag{2.66}$$

- la correction, qui consiste à tenir compte de la dernière observation, dans $\mathcal{L}(E_k | Y_1^k)$ à partir de $\mathcal{L}(E_k | Y_1^{k-1})$, soit :

$$p(E_k | Y_1^k) = \frac{p(Y_k | E_k)}{p(Y_k)} p(E_k | Y_1^k \theta_k)\,. \tag{2.67}$$

Notons que dans le cas du bruit Gaussien, centré, et de covariance $\mathbf{R}$, on a pour $e = (x, t)$:

$$p(Y_k | E_k = e) = \frac{1}{(2\pi)^{d/2}\sqrt{\det(\mathbf{R})}} \exp\left(-\frac{1}{2}\left(Y_k - h_k(x)\right)\mathbf{R}^{-1}\left(Y_k - h_k(x)\right)\right). \tag{2.68}$$

On vient de décrire le principe du filtre optimal. La probabilité conditionnelle est approchée à l'aide d'un système de particules distribuées quasiment selon cette loi (principe de Monte Carlo). Pour cela, entre deux observations successives, *N* trajectoires satisfaisant à l'évolution du filtre optimal sont simulées.

c Principe du filtre particulaire

Concrètement, l'étape de prédiction revient à simuler *N* variables aléatoires indépendantes et identiquement distribuées $(E_{k-1}^1, \ldots, E_{k-1}^N)$ suivant l'approximation de $\mathcal{L}(E_{k-1} | Y_1^{k-1})$ ainsi qu'une variable aléatoire $E_{k|k-1}^i$ suivant $\mathcal{L}(E_k | E_{k-1} = E_{k-1}^i)$ pour chaque variable aléatoire E_{k-1}^i. Un nouveau système de particules est obtenu $(E_{k|k-1}^1, \ldots, E_{k|k-1}^N)$, qui permet d'approcher la loi prédite $\mathcal{L}(E_k | Y_1^{k-1})$ par la loi

$$\frac{1}{N}\delta_{E_{k|k-1}^1} + \ldots + \frac{1}{N}\delta_{E_{k|k-1}^N}\,. \tag{2.69}$$

L'étape de correction pondère chaque élément $E^i_{k|k-1}$ du système de particules prédites par un poids proportionnel à sa vraisemblance :

$$p^i_k = \frac{p(Y_k \left| E_k = E^i_{k|k-1}\right.)}{\sum_{j=1}^{N} p(Y_k \left| E_k = E^j_{k|k-1}\right.)}. \tag{2.70}$$

La loi $\mathcal{L}(E_k \left| Y_1^k\right.)$ est alors approchée par

$$p^1_k \delta_{E^1_{k|k-1}} + \ldots + p^N_k \delta_{E^N_{k|k-1}} p^N_k . \tag{2.71}$$

A l'instant k+1 suivant dans l'étape de prédiction, un échantillon de particules issues de l'approximation de $\mathcal{L}(E_k \left| Y_1^k\right.)$ donné à l'instant k, est redistribué. Cette *redistribution* multiplie les particules les plus vraisemblables par rapport à l'observation courante et élimine les autres, en gardant un nombre constant de particules. Elle a pour effet de naturellement concentrer l'estimateur dans les zones intéressantes de l'espace d'état. Cet effet est illustré par la Figure 2-5. Le filtre classique se trouve sous le nom de "Interacting Particle Filter" (IPF) (Salut 1999) mais peut aussi être désigné de "Bootstrap Filter" (Gordon et al. 1995).

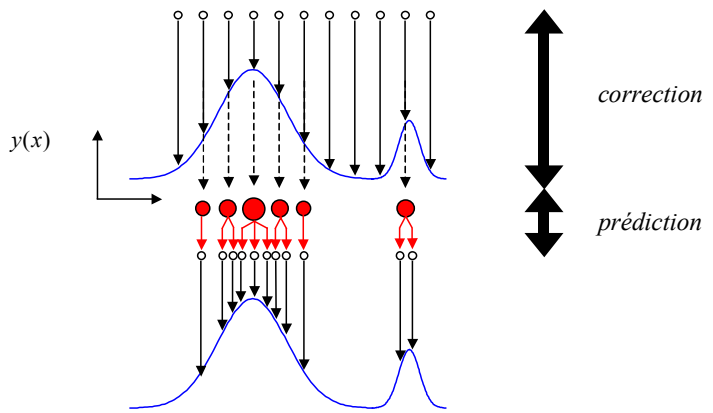


Figure 2-5 : Principe séquentiel de l'estimation de la vraisemblance par la simulation de particules (variables aléatoires).

d Des exemples

Les filtres particulaires nécessitent la conception de distributions suggérées ("proposal distributions") capables d'approximer raisonnablement la distribution a posteriori. En général, concevoir ces suggestions est difficile et peut même conduire à des échecs. Des alternatives ont été proposées (Freitas 1999; Freitas et al. 2000), toutes basées sur une linéarisation. Dans ces travaux, l'approximation d'un EKF est utilisée pour suggérer la distribution du filtre particulaire. La même approche est reprise dans (Merwe et al. 2001), où l'approximation par EKF est remplacée par celle d'un "Unscented Particle Filter" (UPF). Dans ce dernier cas, le filtre résultant voit ses performances accrues, non seulement parce que l'UPF est plus précis que l'EKF, mais aussi parce que la vitesse à laquelle la distribution tend vers zéro est contrôlable. L'approche est basée sur l'intuition qu'il est plus facile d'approximer une distribution Gaussienne, qu'approximer une fonction non linéaire arbitraire. Ils ont montré que l'UKF est supérieur à l'EKF en terme de performance, notamment qu'il est capable de fournir une meilleure estimation de la covariance de l'état. Son inconvénient majeur réside dans le fait qu'il n'est pas applicable aux distributions non Gaussiennes (Doucet et al. 2001).

Cette approche découle des travaux de Julier (Julier et Uhlmann 1997) et de son UKF. Le champ applicatif de l'UKF a été étendu à de nouveaux problèmes non linéaires tels que l'identification de systèmes, ou l'apprentissage des réseaux de neurones (Wan et Merwe 2001).

L'utilisation des filtres particulaires semble particulièrement adaptée au filtrage non linéaire puisqu'ils sont quasiment insensibles aux non linéarités du système, et puisqu'ils dépendent à peine de la dimension de l'espace. Cependant, les algorithmes particulaires développés jusqu'à présent montrent un comportement instable en dehors de certaines hypothèses. On observe notamment la divergence de ces méthodes dans les cas de systèmes faiblement bruités.

Une approche à modèle multiple du filtre particulaire ("Multiple Model Particle Filter", MMPF) est présentée dans (Oudjane et Musso 1999), où l'on s'intéresse au problème du filtrage pour les systèmes à modèles dynamiques multiples appliqués à la poursuite de cible mobile. L'approche particulaire (du type Monte Carlo) est choisie, car elle permet d'approcher de manière non paramétrique et indépendamment de la dimension, la loi conditionnelle de l'état connaissant les mesures, même dans les cas non linéaires ou non Gaussiens.

3.3.6 Un filtre adaptatif temporel

Les approches précédentes utilisent les théories statistiques basées sur des ensembles de mesures. En théorie, elles sont des approximations (parfois même au sens fort) d'un principe d'optimalité. Pour la plupart d'entre elles, elles requièrent une puissance de calcul pas toujours disponible pour des applications temps réel.

Une approche totalement différente du filtre adaptatif de Kalman est proposée dans (Turney et al. 1999). Elle est désignée par le terme anglais "Adaptive Temporal Kalman Filter". Elle est simple et utilise des informations locales, l'adaptation se fait par rapport aux variations dans les signaux et selon les propriétés des bruits. Cette approche est présentée pour avoir une application directe : le filtrage temps réel d'un signal vidéo (avec une implémentation VLSI par FPGA, "Field Programmable Gate Arrays"). Le seul cas scalaire est traité, puisque le filtrage est effectué dans l'espace temporel pour chaque pixel.

Le modèle suivant du système est considéré avec toutes les hypothèses habituellement faites (cf. précédemment) :

$$x_{k+1} = ax_k + w_k \,, \tag{2.72}$$

$$z_k = x_k + v_k \,. \tag{2.73}$$

A partir de ce modèle, à chaque itération et en plus du calcul de l'estimation $\hat{x}_k$ de l'état, les paramètres $a = E[z_k z_{k-1}]/E[z_k^2]$, $\mathbf{R}_k = E[(z_k - \hat{a}_k \hat{x}_{k-1})^2]$ et $\mathbf{Q}_k = E[(\hat{x}_k - \hat{a}_k \hat{x}_{k-1})^2]$ sont estimés (ce sont des scalaires) par :

$$\hat{a}_k = \frac{z_k z_{k-1}}{(z_k^2 + z_{k-1}^2)/2} \,, \tag{2.74}$$

$$\hat{\mathbf{R}}_k = (z_k - \hat{a}_k \hat{x}_{k-1})^2 \,, \tag{2.75}$$

$$\hat{\mathbf{Q}}_k = (\hat{x}_k - \hat{a}_k \hat{x}_{k-1})^2 = \mathbf{K}^2 \hat{\mathbf{R}}_k \,. \tag{2.76}$$

De plus, les auteurs expliquent que pour des raisons de stabilité, quelques mesures sont prises pour déterminer $\hat{a} = \hat{a}_k$, qui est ensuite maintenu constant pendant l'estimation en ligne. La robustesse est accrûe avec une détection de mouvement à l'aide de la grandeur $\gamma = |z_k - a\hat{x}_{k-1}| / \hat{\mathbf{R}}_k^{1/2}$, qui est comparée à une valeur seuil Γ. Le choix de la valeur de Γ est important. Son influence s'observe sur la robustesse du filtre. Il se règle empiriquement en fonction de l'application par essai/erreur. Il dépend surtout du type de bruits qui sont présents dans l'environnement du système.

D'autres auteurs proposent d'estimer la matrice de transition du processus d'évolution. Wojcik par exemple, présente une autre façon de déterminer la matrice de transition (Wojcik 1993). Sa méthode est par contre très restrictive. Le système concerné par l'estimation d'état doit être contrôlable, observable, à minimum de phase, doit comporter une seule entrée et une sortie unique. La matrice de covariance du bruit du processus doit être identité ($\mathbf{Q} = \mathbf{I}$) pour que l'algorithme proposé puisse calculer les estimations des paramètres $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{C}_k$, et $\mathbf{R}_k$. Toutes les hypothèses

généralement considérées dans le cadre du filtrage de Kalman doivent également être strictement respectées.

3.3.7 Conclusion

La substitution d'un KF par un AKF provient généralement d'un manque de connaissance sur les grandeurs à estimer. L'idée est d'utiliser des paramètres du filtre qui évoluent dans le temps en fonction des observations qui sont disponibles à chaque itération.

L'estimation des paramètres supplémentaires peut se faire par une augmentation du vecteur d'état. Les paramètres à estimer sont alors placés dans l'espace d'état pour être mis à jour régulièrement. Les méthodes par correction de covariances compensent les défauts d'une mauvaise modélisation ou les imperfections de l'adéquation modèle/système par des changements de modèles de bruits. Les approches à modèle multiple proposent plusieurs modèles avec des paramètres différents et la problématique revient à choisir lequel convient le mieux à la situation courante. L'algorithme EM permet d'estimer les paramètres non connus du filtre selon le principe du maximum de vraisemblance. Cette approche statistique propose une estimation sur la base de la mise à jour permanente de la densité de probabilité conditionnelle des grandeurs cachées pour un vecteur d'observation connu. Les filtres particulaires sont également une approche probabiliste pour estimer certains paramètres. La probabilité conditionnelle est approchée à l'aide de la technique de Monte Carlo. La discrétisation spatiale n'est pas basée sur un schéma aux différences finies, mais sur une approche par un système de particules (c'est-à-dire une combinaison linéaire de mesures de Dirac).

Beaucoup de ces méthodes sont lourdes. Les approches à modèle multiple nécessitent de la place mémoire et un coût algorithmique non négligeable pour effectuer la sélection du modèle. Les filtres particulaires cherchent à tirer profit des technologies multiprocesseur et explorent de nouveaux algorithmes, parallèles et/ou moins coûteux. L'attrait principal de ces méthodes est que la complexité de l'implémentation est à peu près indépendante de la dimension de l'équation d'état. La mise à jour optimale des matrices de covariance implique un nombre très important d'opérations et de calculs. Certains auteurs préconisent d'alléger la complexité des calculs en utilisant des covariances approximatives qui baissent dans le temps (Gelb 1974; Kelly 1994).

Les simplifications et les méthodes empiriques, qui sont parfois tout aussi efficaces, peuvent néanmoins être associées à un risque, celui de ne plus être capables de prendre en compte les phénomènes de non stationnarité. Ce risque existe aussi avec un filtre de Kalman classique. Comme le filtre optimal est utopique, le choix d'un modèle s'effectue toujours avec un certain nombre de simplifications. Un filtre dont les paramètres ne correspondent pas aux grandeurs réelles, peut rapidement devenir un filtre trop sensible à certaines grandeurs ou à l'évolution d'autres grandeurs.

3.4 Filtrage de Kalman et théorie de l'apprentissage

Il y a une dizaine d'année, Sutton (Sutton 1992a; Sutton 1992b) a montré que l'apprentissage des réseaux linéaires avec la règle BP peut être surpassé en utilisant des coefficients d'apprentissage adaptatifs et distribués. C'est ainsi qu'est apparue l'idée de corriger les poids des réseaux MLP par des méthodes basées sur l'EKF (Puskorius et Feldkamp 1991). Pour une description complète et détaillée de la formulation de l'EKF appliqué à l'apprentissage, on se référera à (Williams 1992). L'apprentissage d'un réseau MLP est vu dans cette démarche, comme un problème d'estimation de l'état d'un système non linéaire. Certains auteurs utilisent d'ailleurs la désignation de "state space neural network modelling". Le modèle générique suivant est utilisé :

$$\mathbf{w}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{w}_k) + \mathbf{r}_k \,, \tag{2.77}$$

$$\mathbf{y}_k = \mathbf{G}(\mathbf{x}_k, \mathbf{w}_k) + \mathbf{v}_k \,, \tag{2.78}$$

où $\mathbf{x}_k$ est l'entrée, $\mathbf{y}_k$ la sortie et $\mathbf{G}(.)$ le "mapping" non linéaire approché par le réseau neuronal paramétré par $\mathbf{w}_k$. Le vecteur $\mathbf{v}_k$ représente l'erreur (l'écart) du réseau par rapport à la réponse

correcte idéale. $\mathbf{F}(.)$ définit l'évolution des paramètres du modèle neuronal qui sont supposés dépendre d'une composante déterministe, et d'une composante aléatoire $\mathbf{r}_k$. Le processus $\mathbf{r}_k$ représente l'incertitude faite sur l'évolution des paramètres. Dans de nombreux travaux aucune connaissance sur la fonction $\mathbf{F}(.)$ n'est requise, puisque les paramètres sont sensés évoluer selon un processus de Markov du premier ordre : $\mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{r}_k$ comme dans (Merwe et Wan 2001) par exemple.

Les premiers développements se contentent de mettre à jour itérativement et individuellement les valeurs des poids des neurones (Puskorius et Feldkamp 1994 ; Williams 1992). Dans (Wan et Nelson 1996), le principe du filtre (linéaire) dual est utilisé pour entraîner un réseau multicouche et mettre à jour simultanément les vecteurs poids des couches cachées. L'idée de se servir de l'EKF pour ajuster la taille du réseau, plus précisément de supprimer les neurones les moins performants, est proposée et explorée.

Depuis, dans (Sum et al. 1998) et dans (Sum et al. 1999), les auteurs gèrent à la fois l'apprentissage et la taille d'un MLP avec un EKF. Ils explorent cette voie, car l'apprentissage basé sur l'algorithme EKF converge plus rapidement que celui basé sur l'algorithme BP, et car le "pruning" est une des méthodes qui permet d'améliorer les capacités de généralisation d'un réseau de neurones. En fait, l'EKF permet d'utiliser des moments d'ordre deux (la règle BP est une méthode d'apprentissage du premier ordre), notamment pour détecter les neurones peu efficaces ou inutiles. L'apprentissage par l'algorithme EKF est meilleur que la conventionnelle règle BP, dans le sens où il utilise les moments statistiques d'ordre deux (les covariances). Ces moments sont essentiels pour qualifier les erreurs faites sur les estimations, ou pour combiner des réseaux séparés en "comittees networks". De plus, il a été démontré dans (Ruck et al. 1992), puis dans (Freitas et al. 1997), que l'algorithme BP est une forme dégénérée de l'algorithme EKF. Un inconvénient des autres approches de "pruning", est qu'elles sont obligées de mesurer explicitement l'importance de chaque poids notamment dans l'OBD ("Optimal Brain Damage") de (LeCun et al. 1990).

L'apprentissage par l'EKF souffre néanmoins de quelques inconvénients, à savoir les choix des conditions initiales (les valeurs initiales des deux premiers moments des poids synaptiques), les matrices de covariances des bruits, $\mathbf{R}_k$ et $\mathbf{Q}_k$ ainsi que la représentation de la dynamique du réseau à travers la matrice de transition $\mathbf{A}$. Dans (Freitas et al. 1997), pour effectuer l'apprentissage d'un réseau MLP, les auteurs estiment ces paramètres ainsi que les conditions initiales avec l'algorithme EM (l'ensemble des paramètres s'écrit alors $\boldsymbol{\theta} = \{\mathbf{R}, \mathbf{Q}, \mathbf{A}\}$). Dans (Sum et al. 1998) les paramètres du systèmes sont estimés récursivement. Les paramètres des bruits sont ajustés séparément par une procédure récursive simple, proposée en premier dans (Iiguni et al. 1992). Ormoneit propose l'utilisation de l'algorithme IEKF pour l'apprentissage d'un "regularized network" (Ormoneit 1999).

Les développements de l'apprentissage par le KF ne se limitent pas aux réseaux à couches (Bouchard 2001; Leung et al. 2001; Nechyba et Xu 1997; Shah et al. 1992). Une démarche similaire à celle développée pour les MLP a récemment été mise au point pour un réseau du type RBF (Yonghong et al. 2000). Nous avons déjà évoqué l'utilisation de l'EKF pour l'optimisation des paramètres d'un réseau SOM durant la phase d'apprentissage (Haese 1998; Haese 1999; Haese et Goodhill 2001).

L'introduction du KF dans l'apprentissage des ANN n'est que le balbutiement de l'apparition des techniques statistiques pour guider l'apprentissage, pour déterminer les poids optimaux d'un réseau. L'enthousiasme de cette nouvelle façon de réaliser un apprentissage est illustré dans l'ouvrage récent de Haykin (Haykin 2001).

4 Conclusion

Les statistiques, longtemps mises de côté et considérées comme impossibles à utiliser dans la pratique (inapplicables), viennent depuis quelques années seulement épauler les sciences de l'ingénieur dans la résolution des problèmes et des défis de demain. Ce retour en force est tout à fait justifié. En effet, bien souvent les résultats de la théorie du traitement des signaux déterministes ne sont pas

suffisants pour aborder certains problèmes. Dans de nombreuses applications les signaux considérés sont aléatoires, en ce sens que la reproduction de la procédure expérimentale ne conduit pas à la même observation du fait de la présence de perturbations que l'opérateur ne peut pas maîtriser totalement. Ces perturbations peuvent être de natures variées : phénomènes de bruit additif ou de distorsion agissant sur des signaux transmis, erreurs liées aux appareils de mesure, etc. L'observation comporte donc certains aléas. La théorie des probabilités et des processus aléatoires offre un cadre satisfaisant pour modéliser ces aléas. Dans la pratique du traitement du signal, on emploie d'ailleurs souvent indifféremment le terme de signal pour désigner un processus aléatoire, ou une de ces réalisations.

De la même façon, les contributions des théories statistiques au connexionisme ne sont plus négligeables, tant au niveau de l'analyse des propriétés et de la convergence d'un apprentissage, qu'au niveau de l'élaboration de nouvelles règles d'apprentissage.

L'étude des caractéristiques d'un apprentissage par les méthodes statistiques a débuté avec les travaux portant sur le biais et la variance d'un réseau (Geman et al. 1992). Ces grandeurs ont aujourd'hui montré qu'elles étaient de bonnes représentations respectivement des capacités de généralisation et de la sensibilité d'un réseau de neurones (Haykin 1999). Grâce à ces outils, il est possible de qualifier un apprentissage. Des études plus récentes ont conduit à la théorie de l'apprentissage. Vapnik (Vapnik 1995) cherche à connaître à l'avance la convergence d'un apprentissage, connaissant sa structure et la base d'exemples. Partant d'un risque empirique, il propose une nouvelle grandeur, la dimension VC qui constitue un renseignement précieux sur la stabilité d'un réseau. D'autres approches proposent l'utilisation d'outils différents pour d'abord modéliser une règle d'apprentissage, puis déduire la convergence vers une solution satisfaisante. Elles utilisent pour la plupart les chaînes de Markov. La voie des statistiques est bien engagée et laisse entrevoir une issue théorique au problème de convergence des réseaux de neurones artificiels.

Ce chapitre nous a également permis d'introduire une méthode Bayésienne pour résoudre les problèmes de filtrage, de lissage, et de prédiction, ainsi que ceux de l'identification de fonctions de transfert. Le filtre de Kalman est aussi une technique d'apprentissage. Cette méthode permet d'incorporer des informations a priori sur les paramètres recherchés. Cet avantage peut cependant aussi se transformer en inconvénient si aucune information a priori n'est disponible. C'est pourquoi nous nous sommes intéressé au filtrage de Kalman adaptatif. Ce filtre possède la particularité d'estimer lui-même ses propres paramètres. Nous avons dressé une revue générale des différentes techniques employées pour l'estimation des paramètres des filtres de Kalman adaptatifs.

Pour pouvoir calculer numériquement les valeurs des estimateurs, on utilise des méthodes numériques afin de résoudre les problèmes d'intégration et de maximisation qui se posent en estimation Bayésienne. Les plus courantes sont l'algorithme EM et les méthodes de Monte Carlo. Ces dernières consistent à résoudre les problèmes d'intégration et d'optimisation de façon approchée, en faisant appel à la simulation de variables aléatoires pour générer des estimateurs des grandeurs recherchées. Elles requièrent cependant de grandes capacités de calculs.

Les théories statistiques et les approches neuronales tentent de résoudre les mêmes problèmes. Ces deux approches peuvent être complémentaires, surtout dans les architectures modulaires de réseaux de neurones comme on le verra dans le chapitre suivant.

Chapitre 3

Apprentissage modulaire

1 Comment introduire plus de connaissances dans les réseaux de neurones

Plus un problème d'apprentissage est complexe, plus il requiert des réseaux de taille élevée. Cette vérité, bien établie depuis quelques dizaines d'années maintenant, a largement freiné l'optimisme naissant et le développement des réseaux de neurones artificiels.

La logique voudrait qu'à un nombre d'entrées et de sorties croissant, il faudrait une interconnexion croissante. La solution systématique pour réaliser un apprentissage qui prend en compte des paramètres supplémentaires (tenir compte d'un environnement non stationnaire, par exemple) consistait jusqu'alors à rajouter des poids supplémentaires, avec tous les inconvénients que cela entraîne : une convergence de l'apprentissage plus difficile, une augmentation du coût calculatoire, des temps de réponse plus importants, etc. Même s'il faut poursuivre l'améliorer les architectures pour le calcul neuronal (Corne 1999), il faut aujourd'hui des idées nouvelles pour ne pas augmenter indéfiniment la taille des réseaux de neurones. Ces idées, qui visent essentiellement à stocker de la connaissance (donc de l'information) autrement qu'à travers les poids synaptiques, doivent concerner indifféremment le neurone lui-même, ou doivent toucher à son organisation au sein de la structure.

Nous allons voir alternativement dans quelle mesure il est possible d'intégrer dans les réseaux connexionnistes des connaissances supplémentaires, soit en concevant des neurones plus complexes, soit en élaborant des architectures plus complexes.

L'élaboration de neurones plus complexes fait appel à toutes sortes de techniques, telles que rajouter des paramètres ou utiliser des fonctions d'activation plus complexes. Elle passe également par la mise au point d'autres types de neurones formels, des neurones plus proches de la réalité biologique, des neurones polynomiaux, des neurones incluant des propriétés statistiques, des neurones comportant des phénomènes oscillatoires, etc. Cette complexification possède un côté dissuasif puisqu'elle rend souvent l'apprentissage plus difficile. La convergence est parfois même compromise.

Une alternative provient de la conception d'architectures neuronales modulaires. Plutôt que d'utiliser un réseau unique pour réaliser un apprentissage, cet apprentissage est confié à des modules de réseaux neuromimétiques organisés de façon cohérente. Après avoir précisé le sens de la modularité neuronale – ce terme est employé souvent avec excès – nous distinguerons alors deux formes essentielles de modularité : la modularité à modèle multiple et la modularité vraie. La modularité à modèle multiple conçoit plusieurs modules qui traitent tous le même problème. Les sorties de ces modules doivent être combinées pour le calcul de la sortie globale. Un certain nombre de méthodes de fusion des sorties seront présentées ; une grande partie découle directement de la théorie des statistiques. La modularité vraie désigne un regroupement de modules qui apprennent séparément

les sous-problèmes d'un problème complexe. Pour cette forme de modularité vraie, nous présentons successivement les types de décomposition et les différents apprentissages.

Une forme particulière de modularité vraie, l'approche des ME (Mélanges d'Experts), sera détaillée. Cette approche est la forme modulaire la plus usuelle des architectures neuronales. Elle combine et reprend deux principes importants des réseaux de neurones, l'apprentissage associatif et l'apprentissage compétitif. La description de l'approche des ME est suivie d'une présentation de certaines de ses extensions.

2 Des neurones plus complexes

La place des poids synaptiques dans une architecture neuromimétique est primordiale. Le modèle de connaissance élaboré par le réseau pendant l'apprentissage est complètement mémorisé et contenu dans les poids synaptiques des neurones. Les poids sont la mémoire du réseau. Améliorer les capacités de mémorisation d'un réseau peut se faire en développant des réseaux de neurones plus élaborés, avec plus de paramètres. Il faut cependant garder à l'esprit qu'au départ, les paramètres d'un neurone ont été regroupés sous la désignation de poids synaptiques. La confusion entre paramètres et poids est dès lors immédiate (Igelnik et al. 2001).

L'objectif à terme n'est pas d'obtenir des modèles neuronaux forcément plus complexes, mais d'aboutir à des modèles de neurones plus riches, c'est-à-dire capables de mémoriser plus d'informations. Et cette richesse va forcément de pair avec un certain degré de complexité. Remarquons sans plus tarder que ce principe va à l'encontre de l'idée de base du connexionisme, qui préconise des unités des calculs distribuées aussi simples que possibles.

Un modèle de neurone plus sophistiqué est un modèle qui tient compte de plus de paramètres. Un certain nombre de travaux de recherche explore l'intégration d'un coefficient d'apprentissage ou d'un facteur d'oubli local, c'est-à-dire propre à chaque neurone. Une autre technique courante pour introduire explicitement des connaissances dans une unité neuronale consiste à utiliser une forme judicieuse de la fonction d'activation. Certains travaux se concentrent donc sur les fonctions de transfert (Duch et Jankowski 2001). Dans (Jankowski et Duch 2001), les auteurs tentent de tirer profit du nombre de neurones, du nombre de connections, et surtout du type de neurones. Ils cherchent à optimiser d'une certaine manière la complexité de l'architecture du réseau par rapport à un problème donné.

Les neurones proches des modèles biologiques sont également des candidats potentiels qui peuvent prétendre être des neurones complexes. Les neurones formels traditionnellement utilisés sont pour l'essentiel des simplifications extrêmes du neurone biologique. Modéliser et intégrer le processus chimique de transmission de l'information entre les neurones ou le point de vue électrique de transmission de l'information au sein même d'un neurone (Sejnowski et al. 1988), permet aussi d'intégrer une forme de connaissance supplémentaire.

Parmi les autres axes de recherche, citons ceux qui explorent et développent des neurones flous (Jin et Sendhoff 1999), des neurones qui intègrent des phénomènes oscillatoires (Cesmeli et Wang 2001), des neurones basés sur la transformée en ondelettes (Reyneri 1999), des neurones polynomiaux (Ronco 1997; Shin et Ghosh 1995), ou des neurones probabilistes (Anouar et al. 1998; Specht 1990). Ces dernières années ont vu apparaître des réseaux de neurones utilisant des statistiques d'ordre supérieur (Mathis et al. 2001). Dans (Güler 2001) l'objectif est clairement affiché : ne pas faire augmenter la taille des ANN avec la complexité de la fonction à apprendre. Le réseau proposé, un réseau à couches qui respecte la théorie de Vapnik, grâce aux statistiques d'ordre supérieur, possède de meilleures capacités de généralisation qu'un réseau multicouche auquel il est comparé. La dimension VC du réseau utilisant les statistiques d'ordre supérieur est inférieure à celle du réseau "feedforward".

L'approche de Miyamoto-Kawato-Setoyama-Suzuki (Miyamoto et al. 1988; Soucek 1989) se situe également dans ce contexte. Le principe, représenté Figure 3-1, reprend le schéma

d'apprentissage d'un réseau Adaline, à la différence près que les grandeurs d'entrée sont transformées par un ou des modèles de connaissance. En effet, les sous-systèmes de la Figure 3-1 sont des fonctions non linéaires, or ces fonctions sont une manière très simple d'introduire explicitement un modèle de connaissance dans le processus d'apprentissage. L'approche est validée dans le cadre du contrôle robotique, 43 fonctions (des sous-systèmes) sont utilisées pour contrôler un robot à trois degrés de liberté. Ces fonctions, ainsi que d'autres paramètres et des détails supplémentaires, sont donnés dans (Miyamoto et al. 1988). Si Kawato (Kawato et al. 1987) qualifie son approche neuronale de hiérarchique, c'est que le réseau de la Figure 3-1 est inscrit dans un schéma plus complexe d'un contrôleur robotique. L'auteur considère trois étapes : 1) la détermination d'une trajectoire dans l'espace des coordonnées visuelles, 2) la conversion des coordonnées visuelles en coordonnées articulaires, et 3) la détermination de commandes moteur (Kawato 1989).

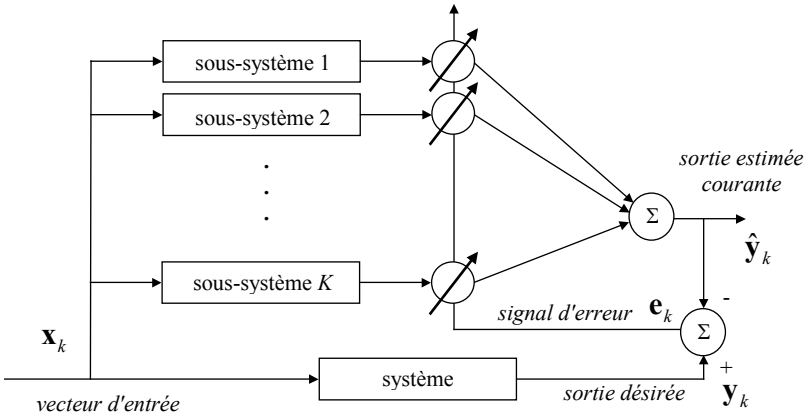


Figure 3-1 : Le modèle de Miyamoto-Kawato-Setoyama-Suzuki.

Le leitmotiv récurrent à toutes les approches précédentes est qu'une tâche ne peut être apprise avec un réseau de neurones artificiels que grâce à une certaine flexibilité, et celle-ci n'apparaît qu'avec un nombre suffisant de paramètres latents.

La complexification à outrance des neurones est cependant vite freinée. Les gens familiers des réseaux neuromimétiques savent que plus un réseau est complexe, plus il est difficile de l'adapter et donc d'apprendre. Le simple fait d'utiliser des neurones non linéaires plutôt que des neurones dont la fonction d'activation est linéaire, peut rendre dans certains cas l'apprentissage impossible.

Une alternative pour apprendre plus sans complexifier les neurones, consiste à mieux les organiser. C'est à quoi s'attachent les approches neuronales modulaires.

3 La modularité

3.1 Introduction et définition

Les sciences de la vie sont et restent une source d'inspiration riche pour l'homme.

La modularité est présente à toutes les échelles de la nature et est à la base de la résolution d'un bon nombre de problèmes complexes. En physique les exemples ne manquent pas avec les galaxies, les planètes, les solides, les molécules, etc. L'atome peut être vu comme un module de base. La biologie est également basée sur une structure modulaire. Le corps humain est non seulement constitué de différents modules, mais son cerveau est également construit sur un principe modulaire (Churchland et Sejnowski 1992) : la division du cerveau en deux hémisphères est une première étape

de modularité, chaque hémisphère est elle-même constituée de différents blocs fonctionnels plus ou moins autonomes.

On retrouve dans les théories de l'intelligence artificielle, dans la théorie de David Marr sur l'organisation du processus de la vision chez l'homme (Marr 1982), trois niveaux : le premier envisage une théorie « computationnelle » abstraite du système qui concerne la nature des informations à traiter ; le second, la sélection d'une forme de représentation pour les entrées, les sorties ainsi que les processus de transformation ; et le troisième une description de la machinerie nerveuse. Ainsi, les travaux de Marr sur les mécanismes de la vision lui ont suggéré que le cerveau est entièrement organisé de manière modulaire.

La modularité est le paradigme de base de la plus populaire des thèses de psychophysiologie (Caelli et al. 1999). Le principe de la modularité a aussi fortement influencé la psychologie cognitive. Il s'est logiquement inscrit dans le développement de la science de l'apprentissage, mais n'a été que très récemment adopté. L'idée de base est pourtant louable : la modularité fait ressembler un réseau de neurones artificiels plus encore à son pendant biologique.

Dans un premier temps, il faut donner une définition précise de la modularité, puisqu'il existe dans la littérature du connexionisme plusieurs sens différents de la modularité. Nombreux sont ceux qui utilisent le terme d'apprentissage modulaire, avec à chaque fois des connotations différentes.

Deux définitions se contredisent même, celle de De Francesco (De Francesco 1994) et celle de Ronco et Gawthrop (Ronco et Gawthrop 1995). Le premier auteur considère que le réseau multicouche est un réseau modulaire qui comprend autant de modules que de couches. Les seconds voient un lien direct entre modularité et notion de localité. Pour eux, un module neuronal est un neurone ou un réseau qui utilisent certaines des entrées disponibles à son niveau. Selon cette deuxième définition, les couches d'un réseau multicouche ne sont donc pas des modules.

Dans (Boers et al. 1993), une approche plus superficielle est donnée. Un module neuronal est une décomposition en parties identifiables, chacune avec son propre but ou fonction. Il se trouve que cette définition s'applique à presque tout. Rouzier (Rouzier 1998) propose également une définition sur la modularité neuronale. Selon elle, un module neuronal est un neurone ou un réseau qui assume une certaine fonction au sein d'un système. De ce point de vue, la notion de fonction est primordiale. Une couche d'un réseau multicouche est un ensemble de neurones non interconnectés, et ne peut donc pas être un module.

Notre vision se conforme à cette dernière définition de la modularité neuronale et nous y ajouterons le fait qu'un module doit être à la fois interprétable et pertinent. Cela signifie qu'un module doit remplir un rôle bien défini au sein de l'architecture générale, et que ce rôle doit être justifié au sein de la tâche à réaliser. Un neurone est alors une forme de micro-modularité, un réseau de neurones une forme de modularité simplifiée à l'extrême, et plusieurs réseaux constituent une forme de modularité plus évoluée. Notons que d'après cette définition, les réseaux de réseaux de neurones (comme le réseau SSOM présenté dans le Chapitre 1) apparaissent comme une forme de macro-modularité.

Caelli, Guan et Wen (Caelli et al. 1999) distinguent deux façons d'aborder une décomposition modulaire. La première consiste à :

- concevoir des sous-réseaux fonctionnellement indépendants, chacun ayant des fonctions propres, des communications propres, et des caractéristiques d'adaptation propres ;
- développer les représentations et les modèles des modules neuronaux qui ont des fonctions identifiables, soit par rapport au canevas de la décomposition globale, soit par rapport à la nature de la tâche à résoudre et de son environnement.

La seconde façon cherche à :

- organiser des ensembles de modules neuronaux en une décomposition hiérarchique ;

- ajuster progressivement les modules par des méthodes d'optimisation tel que l' "Evolutionary Programming" (Yao et Liu 1997), ou en utilisant des réseaux évolutifs tels que les "Self-Generation Neural Networks", ou les "Self-Generation Tree Network". Cette dernière approche comprend notamment les arbres de Perceptrons, les arbres de SOM, ou encore les arbres de LVQ. L'approche est propice aux tâches de classification, puisque les modules qui servent de classificateur peuvent être disposés en étages (Wen et al. 1992).

Plusieurs auteurs ont antérieurement proposé des vues d'ensemble ou des classifications (Buessler 1999; Caelli et al. 1999; Rouzier 1998). L'issue de ces travaux nous permet de distinguer deux formes de modularité dans la conception des architectures neuronales utilisant plusieurs réseaux, deux approches opposées :

- la *modularité à modèle multiple* ou modularité itérative, désignée en anglais par "ensemble-based approach" ou "committee framework", lorsque plusieurs modules traitent tous le même problème et que leurs sorties sont combinées pour calculer la sortie globale ;
- la *modularité vraie*, encore appelée modularité différenciée et en anglais "neural nets ensemble", lorsque un regroupement de réseaux est composé de modules qui apprennent séparément les sous-problèmes d'un problème complexe.

Ces deux formes de modularité sont présentées d'une manière plus poussée dans la section suivante.

3.2 La modularité à modèle multiple

Une première forme de modularité, l'approche à modèle multiple (MM, "Mixture Model"), est constituée d'un ensemble de réseaux ayant appris à résoudre la même tâche. Les sorties de tous les réseaux sont combinées pour donner une sortie globale.

Réaliser un ensemble de réseaux performant revient à une sélection de modèle, donc bien souvent à un problème de classification. La sélection d'un modèle est d'autant plus cruciale qu'elle conduit à de meilleures capacités de généralisation (Sugiyama et Ogawa 2000).

3.2.1 Motivations

L'avantage de l'emploi de plusieurs réseaux de neurones dédiés à la même tâche, avec les mêmes entrées, plutôt qu'un seul, tient surtout à la robustesse de l'approche. Au lieu d'utiliser l'avis d'un réseau unique, on préfère utiliser l'avis de plusieurs. Les risques d'obtenir une sortie erronée seront dans ce cas plus faibles.

Il n'est pas intéressant de combiner un ensemble de réseaux identiques. Par contre, il est avantageux de combiner les réponses de plusieurs réseaux différents. Les réseaux neuromimétiques peuvent être différenciés par des poids initiaux différents, par l'architecture (avec un nombre de couches cachées différent, avec un nombre de neurones différent pour les couches cachées, etc.), par l'algorithme d'apprentissage, par une base d'apprentissage différente. Les techniques de "bootstrapping" (Breiman 1996a) utilisent des ensembles d'apprentissage disjoints ou mutuellement exclusifs.

Ce dernier principe, utiliser des données différentes pour faire apprendre les différents réseaux, se rapproche fortement de la seconde forme de modularité. Si chaque partition est représentative de l'ensemble du problème, la modularité peut être considérée comme étant à modèle multiple, autrement, si chaque partition est représentative d'une distribution différente, alors la modularité peut être qualifiée de modularité vraie.

3.2.2 Quelques méthodes de fusion des sorties

Les techniques pour calculer la sortie globale d'une assemblée de réseaux sont diverses et variées. Il y a entre autres, les techniques de vote, la moyenne et la moyenne pondérée, les combinaisons non linéaires, les techniques de méta-learning, l'approche supra-Bayésienne ou encore l'approche des généralisations empilées.

Parmi les techniques de vote, on dénombre le vote moyen ou "average voting", le principe du vote majoritaire ("majority voting scheme") ou le principe du vote à l'unisson. Il existe également des votes pondérés. Un poids est alors associé à chaque classificateur, il représente une mesure de confiance. Une méthode de vote pondéré est présentée dans (Drucker et al. 1994) pour combiner des classificateurs entraînés sur des bases d'apprentissage obtenues par "boosting". Deux autres approches sont présentées dans (Alpaydin 1993), toutes deux basées sur des mesures de fiabilité. Dans (Battiti et Colla 1994), la technique du vote est également utilisée dans une assemblée de réseaux appliquée à un problème de classification.

Concernant la moyenne, la sortie est donnée par la combinaison linéaire des sorties des réseaux. La méthode est appelée BEM ("Basic Ensemble Method") lorsqu'elle utilise la moyenne. Lorsqu'elle utilise une moyenne pondérée, elle est appelée méthode GEM "Generalized Ensemble Method" (Perrone et Cooper 1993). Pour les pondérations, on peut utiliser l'inverse de la variance de chaque réseau. Le principe est, dans ce cas, identique à celui des moindres carrés à minimum de variance. On peut également citer Hashem (Hashem 1997) qui cherche en fait la meilleure combinaison possible. Il introduit la notion de OLC ("Optimal Linear Combinations"). Sa technique s'appelle MSE-OLC et elle consiste à chercher la solution qui minimise l'erreur quadratique moyenne de la sortie globale. La robustesse de la méthode se dégrade fortement en présence de dépendances linéaires parmi les sorties (et/ou les erreurs) des réseaux. Deux autres méthodes sont présentées dans (Tresp et Taniguchi 1995) ; elles utilisent toutes les deux une pondération basée sur la variance des réseaux. Par contre, si dans le premier cas la modularité est à modèle multiple (les bases d'apprentissages sont différentes mais les distributions représentent un même problème), dans le second les auteurs se placent dans le contexte de la modularité vraie. Cette dernière méthode s'approche du concept des Mélanges d'Experts.

La « théorie du consensus » (Benediktsson et Swain 1992) constitue un exemple de combinaison non linéaire des sorties des réseaux. La méthode la plus courante reste cependant celle basée sur la théorie de l'évidence de Dempster-Shafer (Rogova 1994). Des approches plus récentes utilisent les règles de la logique floue (Cho et Kim 1995) ou le concept d'évolution des algorithmes génétiques (Cho 1997).

Le principe du "méta-learning" consiste à faire apprendre de nouveaux réseaux à partir des réponses d'un ensemble de réseaux de base. Ce concept est développé et mis en pratique par Wolpert (Wolpert 1992).

L'approche supra-Bayésienne est basée sur une combinaison de probabilités de distribution (Jacobs 1995). Si les bases théoriques de cette approche sont solides, sa mise en œuvre est moins évidente et bute généralement sur le problème du choix et de l'évaluation de la fonction de vraisemblance. Un algorithme d'inspiration biologique est proposé dans (Fleuret et Brunet 2000). Les auteurs ont montré qu'il était un processus de décision Markovien convainquant.

L'approche des "stacked generalizations" (généralisations empilées) est une méthode d'empilement de réseaux (Wolpert 1992) qui distingue deux niveaux de réseaux : l'ensemble des réseaux que l'on veut regrouper sont considérés comme des réseaux de niveau 0, le réseau qui combine les sorties des réseaux de niveau 0 est le réseau de niveau 1. Les réseaux de niveau 0 sont successivement entraînés sur des bases d'apprentissage différentes. La base d'apprentissage du réseau de niveau 1 est progressivement construite à partir des sorties des réseaux de niveau 0. En effet pour un exemple de cette base, l'entrée est constituée par les sorties des réseaux de niveau 0 pour un

exemple non appris, et la sortie de cet exemple est la sortie désirée. Le réseau de niveau 1 apprend donc à calculer la sortie globale à partir des sorties des réseaux du niveau 0. Ce principe est un moyen de corriger les biais des réseaux pour fournir la réponse globale.

De nouvelles familles de méthodes sont prises en considération depuis quelques années pour fusionner les sorties des réseaux constituant une assemblée. Parmi elles, on compte les *k* plus proches voisins (Ripley 1996), les réseaux RBF (Ukrainec et Haykin 1996), les méthodes à noyaux (Hérault et Jutten 1994), ou encore la logique floue (Fiordaliso 1999).

3.2.3 Conclusion

Utiliser les sorties combinées est une alternative efficace à la méthode qui consiste à choisir le réseau qui présente les meilleures performances. Or, combiner les sorties de plusieurs réseaux n'est utile que si ces réseaux délivrent des sorties différentes, des temps de convergence différents, ou même des architectures différentes. L'approche à modèle multiple repose donc essentiellement sur l'habilité de disposer de réseaux distincts et sur la méthode de fusion de leurs sorties.

L'élément déterminant de l'efficacité d'un regroupement de réseaux de neurones est l'indépendance de ses membres en terme d'erreur (dans le sens où ils génèrent des erreurs différentes). Avec des réseaux indépendants, si l'un d'eux fait des erreurs sur la base de généralisation, ces erreurs sont propres à ce réseau et ne doivent pas être partagées aux autres réseaux du regroupement. La variance est alors élevée. Le fait de les regrouper permet de faire baisser cette variance, ce qui signifie une diminution de l'erreur globale et des performances meilleures.

Pour qu'un ensemble de réseaux donne des performances supérieures à celles d'un réseau unique, il faut faire en sorte que ces réseaux restent indépendants. Il existe deux méthodes pour cela :

- la première consiste à mesurer explicitement le niveau d'indépendance des réseaux puis à en tenir compte dans le calcul de la sortie globale comme le fait (Hashem 1997),
- la seconde repose sur la sélection ou la création de réseaux indépendants. Une étude comparative est fournie dans (Tumer et Ghosh 1996). Il existe diverses techniques, empiriques ou originales, comme celles basées sur les algorithmes génétiques (Amaral et al. 1995) avec des opérateurs tels que la mutation et le "cross-over" pour créer de nouveaux réseaux. D'autres sont basées sur la méthode du "bootstraping" (avec un ajout de bruit aux entrées), et certaines vont jusqu'à supprimer, parmi les réseaux liés, ceux qui présentent les plus fortes erreurs.

Il semble que faire varier les conditions initiales soit probablement moins efficace qu'entraîner les réseaux avec des bases d'apprentissage différentes. Des exemples de méthodes qui fournissent des bases d'apprentissage différentes sont le "bootstrap" (Lebart 1997), la validation croisée ou l'ajout de bruit. L'inconvénient majeur de ces méthodes provient du fait qu'elles peuvent échafauder les mêmes fonctions même si leurs bases d'apprentissage sont constituées d'exemples distincts. Ceci entraîne des erreurs d'estimation similaires et de détruire l'indépendance qui doit régner au sein de l'ensemble. Il faut donc trouver un bon compromis.

3.3 La modularité vraie

La modularité vraie désigne le regroupement de réseaux, où chaque élément apprend à résoudre un aspect du problème seulement. Chaque élément ou module devient après apprentissage, un spécialiste d'une partie du problème appelée sous-tâche.

La modularité vraie, de surcroît, suppose que les modules sont autonomes, c'est-à-dire que les liens qu'ils entretiennent entre eux passent exclusivement par des échanges (communications) via les flux d'entrées/sorties. Des paramètres latents d'un réseau ne peuvent pas servir d'entrée à un autre

réseau. La problématique est formulée en anglais en terme de "modular learning". La Figure 3-2 illustre la modularité vraie.

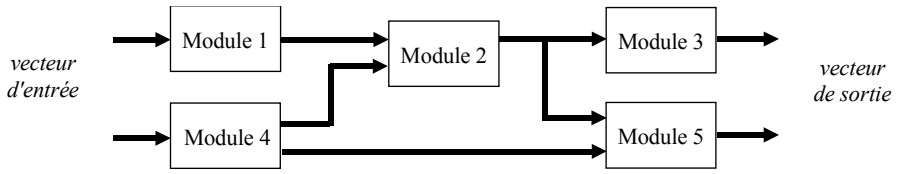


Figure 3-2 : Exemple d'une architecture modulaire (modularité vraie).

3.3.1 Motivations

Chaque module se spécialise sur un aspect du problème à résoudre. Utiliser un ensemble de « spécialistes » peut présenter plusieurs avantages. Les principales motivations qui peuvent justifier ce choix de modularité sont l'impossibilité de résoudre une tâche avec un réseau unique, l'amélioration des performances, une réduction de la complexité (plusieurs réseaux de taille moindre utilisent moins d'unités neuronales qu'un seul réseau), la possibilité d'introduire des connaissances a priori, la réutilisation d'un savoir déjà acquis, et la simplification de l'apprentissage.

La complexité d'un réseau est parfois trop grande pour résoudre un problème. En effet, il existe, selon les applications, des problèmes de mise en œuvre. Un réseau de taille importante n'est supporté que par très peu de calculateurs, son temps de réponse peut alors être trop élevé. Des problèmes dans l'apprentissage lui-même peuvent également survenir : un réseau de grande taille contient forcément beaucoup de paramètres latents, ce qui entraîne une augmentation des risques de divergence.

La réduction de la complexité et l'interprétabilité du système est également une motivation de la modularité. Examinons par exemple la méthode "Structural Learning With Forgetting" (SLWF) dans (Ishikawa 1996) qui aboutit à la formation de modules, en supprimant les connexions inutiles dues à l'oubli. C'est le même principe que le mécanisme d'oubli dans les réseaux : l'oubli ou le déclin de certaines connexions diminue d'une quantité constante à chaque mise à jour des poids entre des neurones non sollicités. L'idée est de conserver un réseau final constitué uniquement de connexions absolument nécessaires. Le principal objectif de la méthode SLWF est de découvrir des régularités et des règles dans la base d'apprentissage. Le fait que le système soit modulaire permet de mieux le comprendre. Ici, on peut l'interpréter en analysant la façon dont les modules sont connectés entre eux.

La modularité vraie est un moyen ingénieux d'insérer des connaissances a priori dans le système d'apprentissage. La motivation est d'introduire de la connaissance (ici par le biais de la structure neuronale elle-même) afin d'aboutir à de meilleures performances. Pour Bennani et Gallinari (Bennani et Gallinari 1992), la décomposition de la tâche doit refléter la structure du problème. Introduire des connaissances a priori se fait généralement en utilisant la suggestion d'une décomposition adéquate de la tâche globale. Il en résulte un inconvénient : la décomposition est directement tributaire de l'application ou du problème considéré. Pour chaque problème ou sous-problème, pour être efficace, la décomposition doit être spécifique. Certains problèmes sont plus aptes à être décomposés que d'autres : la robotique avec de grands mouvements, avec de petits déplacements du type tâtonnement, ou avec le positionnement en vue d'une préhension d'objet ; la reconnaissance de mots écrits ; l'identification de locuteurs ; ou les problèmes de classification sur de larges populations préfèrent en général une décomposition du type hiérarchique.

La modularité vraie permet la réutilisation d'un savoir déjà acquis (appris). On parle aussi de transfert de connaissances ou de "adaptive generalisation" (Sharkey et Sharkey 1992). L'idée de base

consiste à réutiliser des modules qui ont déjà appris, ce qui permet de réduire les temps d'apprentissage. Comme exemple, on peut citer l'algorithme "Modular Iterative Atrophy" (MIA) (Smotroff et al. 1991) qui conserve les avantages du principe "Cascade Correlation" (apprentissage incrémental) et en élimine ses inconvénients (obtention de réseaux complexes car profonds et avec de nombreuses couches) grâce aux principes de compétition et atrophie des connexions inutiles.

Si pour une application donnée, la modularité vraie est possible, alors bien souvent elle est une méthode de simplification de l'apprentissage.

3.3.2 La décomposition

La décomposition d'une tâche en sous-tâches peut se faire, soit explicitement, soit automatiquement.

La décomposition explicite doit être faite par le concepteur du réseau. Elle est un excellent moyen d'insérer dans l'architecture neuronale de la connaissance a priori sur le problème. La décomposition neuronale est alors directement déduite de la structure du problème. Pour un problème donné, la décomposition est particulière et ne sera pas transposable à un autre problème. On voit fleurir des décompositions hiérarchique et série (les plus simples et les plus évidentes). Les décompositions parallèles s'approchent beaucoup de la modularité itérative.

Lorsqu'on ne dispose d'aucune information préalable sur le problème, on opte généralement pour une décomposition automatique. Pour ce type de décomposition, une partition des données de la base d'apprentissage est souvent plus performante qu'un apprentissage sur tout un espace d'entrée. Il existe deux façons de procéder : la décomposition automatique peut se faire, soit avant, soit pendant la phase d'apprentissage.

- Avant l'apprentissage :
  Dans (Aussem et Murtagh 1997), Aussem et Murtagh proposent dans un premier temps de partitionner la base d'apprentissage, puis de faire apprendre chaque module séparément sur un sous-ensemble de la base d'apprentissage. Son application porte sur les séries des taches solaires, et son approche modulaire aboutit à une meilleure précision des prédictions par rapport à un seul réseau. La méthode des "*k*-means" (Chinrungrueng et Séquin 1995) est une méthode pratique pour diviser la base d'apprentissage (Bennani 1995). L'application concerne un problème classique : l'identification de locuteurs. Une approche différente, le "Cooperative Modular Neural Network" (CMNN), est proposée dans (Auda et Kamel 1998) pour des problèmes de classification dont les classes se chevauchent.
- Pendant l'apprentissage :
  Un partitionnement pendant la phase d'apprentissage signifie que les modules se répartissent et s'organisent en cours d'apprentissage. L'exemple le plus connu est l'approche des Mélanges d'Experts (ME), où un module annexe ("gating network", parfois appelé module auxiliaire) supervise la répartition des experts au cours de l'apprentissage et contrôle la décomposition en sous-problèmes.

Une autre façon d'analyser les formes de modularité revient à étudier la nature de la décomposition, c'est-à-dire l'agencement des modules les uns par rapports aux autres.

3.3.3 Nature de la décomposition

Une décomposition peut être soit séquentielle, soit parallèle et la combinaison de décompositions séquentielles et parallèles conduit à des structures dites hiérarchiques.

Dans une décomposition séquentielle (ou série), le problème est découpé en sous-problèmes successifs, chacun appris par un module spécialisé. La sortie du module k forme l'entrée du réseau k+1 suivant. Cette forme de décomposition est très répandue. En robotique, le contrôleur issu de deux

réseaux neuronaux qui se succèdent (un pour la reconstruction 3-D, l'autre pour l'élaboration des commandes à partir des grandeurs reconstruites) est un exemple de décomposition séquentielle (Breton 1999). Un autre exemple concret est présenté dans (Bennani et Bossaert 2001) où un module de détection suivi d'un module de diagnostic, puis d'un module de décision, permet de diagnostiquer les problèmes de trafic dans les réseaux téléphoniques.

Une décomposition est dite parallèle lorsque les sous-tâches s'exercent en même temps. Il y a deux décompositions parallèles distinctes, la relation de coopération et celle de surveillance.

- La coopération :
  Il existent au moins deux façons d'instancier une relation de coopération entre modules (Bennani 1995) : la décomposition ET et la décomposition OU. Une décomposition est appelée décomposition ET lorsque la résolution d'une tâche implique la résolution de toutes les sous-tâches. Une décomposition est qualifiée de décomposition OU lorsque la résolution d'une tâche revient à résoudre une des sous-tâches. Par conséquent, les modules d'une décomposition ET ne traitent qu'une partie du vecteur d'entrée global et la sortie globale nécessite la réponse de tous les modules. Inversement, les modules d'une décomposition OU prennent le vecteur d'entrée en entier et la réponse d'un seul module est suffisant pour calculer la réponse globale. Alors que dans le cas de la décomposition ET, où chaque module apprend à traiter une partie du vecteur d'entrée, chaque module de la décomposition OU apprend une partie de l'ensemble des vecteurs présentés en entrée, c'est-à-dire une portion de l'espace d'entrée.
- La surveillance :
  On parle aussi de supervision quand un module en contrôle un autre. Ce principe permet de coordonner les activités du système d'apprentissage : le travail peut être réparti parmi différents réseaux, les sorties de chacun sont fusionnées pour le calcul de la réponse globale. Comme exemple, on peut citer le réseau sélecteur de l'approche ME qui supervise les modules et pondère leur sorties.

Les décompositions hiérarchiques se servent à la fois des combinaisons parallèles et séries. On obtient alors une organisation en arbre. Ces décompositions sont les plus courantes et parmi elles sont rangés les modèles HMEs, autrement dit les Mélanges d'Experts Hiérarchiques.

3.3.4 Nature de l'apprentissage

Dans le principe d'apprentissage d'un système modulaire, trois types d'apprentissage peuvent être dissociés :

- L'apprentissage indépendant :
  L'apprentissage d'un module est réalisé de façon indépendante lorsque les autres modules n'interviennent pas et n'influent pas pendant la phase d'apprentissage. Les modules ne sont reliés que pour la phase d'utilisation/restitution. Ce principe est doté d'un avantage intéressant : il est possible d'apprendre des sous-problèmes plus simples, mais (et c'est un inconvénient majeur) il faut savoir apprendre séparément un sous-problème. Un exemple particulier est l'algorithme "parallel-modular Recurrent Cascade Correlation" (RCC) dans (Kirschning et al. 1995) qui est basé sur le principe de (Fahlman et Lebiere 1990). C'est le plus courant dans la littérature. L'architecture est fixe et déterminée au départ.
- L'apprentissage coopératif :
  Cette forme d'apprentissage ne s'applique qu'aux architectures fixes et déterminées dès le départ mais la fonction de chacun des modules est déterminée au cours de l'apprentissage. Autrement dit, le nombre de modules ainsi que leur topologie est

prédéterminée, et chaque module apprend par lui-même à se spécialiser sur une sous-tâche. L'apprentissage est réalisé sur tous les modules en même temps, à l'aide des entrées et des sorties globales. Un exemple est donné par les Mélanges d'Experts, où les différents experts se voient attribuer un rôle par le superviseur en cours d'apprentissage. Ils se spécialisent alors sur une portion de l'espace des entrées.

- L'apprentissage incrémental :
  Cet apprentissage supporte une architecture évolutive. Lorsque le nombre de modules, leurs connectivités et leurs fonctions changent au cours de la phase d'apprentissage, l'apprentissage est appelé incrémental. Des exemples peuvent être tirés de l'apprentissage incrémental donné par les réseaux "growing/pruning".

Les réseaux neuronaux modulaires sont de toute évidence une avancée significative dans la théorie de l'apprentissage. L'apprentissage modulaire ne va pas sans quelques inconvénients, il pose de nouveaux problèmes à la communauté des sciences cognitives.

La nature de l'apprentissage dépend de la nature de la décomposition, des connaissances a priori, mais aussi de la disponibilité de certaines grandeurs internes du système à apprendre. Ces dernières sont rarement disponibles, et c'est pour cette raison que sont développées des architectures d'aide à l'apprentissage, tel que l'apprentissage bidirectionnel.

3.3.5 Une forme particulière : la modularité récursive ou bidirectionnelle

Les formes élémentaires de décomposition modulaires sont la décomposition séquentielle et la décomposition parallèle. La combinaison de ces deux formes conduit à des architectures complexes qualifiées de hiérarchiques. Ces structures hiérarchiques complexes ne sont pas souhaitables, à moins de disposer de suffisamment de connaissances préalables (Castillo 1998; Castillo et al. 2000). Une architecture, même modulaire doit rester aussi simple que possible.

La règle d'apprentissage d'un réseau dépend de son architecture. Le principe se répète pour les architectures modulaires puisque la nature de l'apprentissage dépend de la décomposition résultante. Or, l'organisation des réseaux ne va pas sans apporter des problèmes nouveaux dans l'apprentissage. Si la décomposition permet de simplifier l'apprentissage de manière locale, au niveau modulaire donc, il n'est réellement possible que si un certain nombre d'éléments lui sont fournis. Ceci est d'autant plus vrai que l'apprentissage des modules se fait selon un apprentissage supervisé, qui nécessite obligatoirement un vecteur d'entrée et un vecteur de sortie désirée. Le module se doit alors de reproduire un sous-système du système à apprendre. Un apprentissage peut être fortement compromis, voir impossible si ces informations ne sont pas accessibles. C'est en cela qu'il ne faut pas, lors de la conception d'une architecture modulaire, chercher à décomposer à outrance. Comme précisé dans la définition adoptée de la modularité, il s'agit d'obtenir des modules qui sont pertinents.

L'inaccessibilité des grandeurs intermédiaires nécessaires à l'apprentissage peut être illustré grâce à la Figure 3-2. Si l'apprentissage des cinq modules est supervisé, chaque module doit disposer, en plus d'un vecteur d'entrée, d'un vecteur de sortie désirée.

De là, part le concept d'introduire des modules supplémentaires. Ces modules ne participent qu'à l'apprentissage de l'ensemble neuronal et n'interviennent pas dans le calcul de la sortie. Ce principe d'aide à l'apprentissage est présenté par l'exemple de la Figure 3-3.

L'originalité de cette approche réside dans le fait qu'elle utilise un flux d'information inverse de celui qui existe naturellement. On voit en effet sur la Figure 3-3, que le flux du module 3 prend son vecteur d'entrée dans l'espace d'arrivée du système. C'est pour cette raison qu'est employé le terme d'apprentissage bidirectionnel. Le principe a permis, depuis quelques années, de renforcer l'apprentissage des mémoires associatives (Kosko 1988; Neubauer 1995; Park et al. 2001). Il a été

étendu pour résoudre le problème de l'apprentissage direct et inverse de fonctions robotiques (Buessler 1999; Wada et al. 2001; Wada et Kawato 1993).

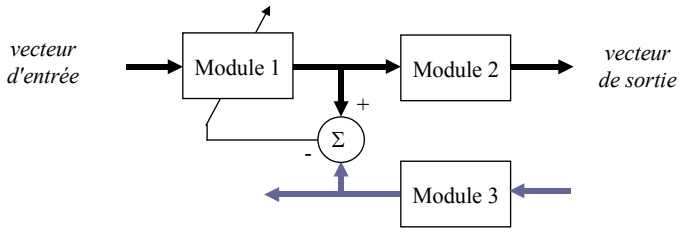


Figure 3-3 : Exemple d'une architecture d'aide à l'apprentissage.

Enfin, notons que les mémoires associatives sont souvent par essence même bidirectionnelles. En plus de relier une sortie à une entrée, la plupart savent apparier une entrée à une sortie. D'autres réseaux de neurones reproduisent ce principe. Le réseau ESOM présenté au Chapitre 1 est par exemple, parfaitement bidirectionnel.

3.4 Conclusion sur la modularité

La modularité est une façon d'introduire des connaissances explicites dans les réseaux de neurones dans l'objectif d'en accroître les performances. Elle est une nouvelle façon d'organiser et de structurer une forme de connaissance.

Améliorer les performances d'un système neuronal par une approche modulaire peut se faire, soit en créant un ensemble de solutions en « parallèle » et en les combinant, soit en décomposant la tâche et en spécialisant les modules sur des sous-tâches. Un gain de performance est également possible en combinant ces deux solutions. Par contre, il est à ce jour, impossible de savoir à l'avance quelle solution est plus performante que l'autre, ou quelle décomposition est plus judicieuse qu'une autre. En fait, il n'existe pas de méthode pour choisir le type de modularité à utiliser face à un problème d'apprentissage.

Pour les problèmes de grande taille, on préfère généralement la modularité vraie. Son point fort est de permettre l'introduction d'informations supplémentaires dans le système d'apprentissage. Une bonne décomposition est un critère essentiel à un bon apprentissage.

Lorsqu'aucune information n'est disponible préalablement, on préfère un apprentissage incrémental. Pour des tâches de classification, on choisira plutôt l'apprentissage modulaire à modèle multiple, afin d'effectuer un choix final en disposant des choix des différents modules. Mais pour la même tâche de classification, avec les connaissances préalables suffisantes, on conçoit de préférence une approche hiérarchique adéquate.

Les approches modulaires ne sont pas forcément plus performantes qu'un réseau unique. Il arrive parfois de ne pas pouvoir mettre en œuvre une architecture modulaire pour des raisons pratiques (trop peu de connaissances, pas assez d'informations disponibles pour réaliser les apprentissage des éventuels modules), ou pour des contraintes liées à l'application (temps de calcul, capacité mémoire, etc.).

Lorsqu'elle utilisée en bonne connaissance de cause, et dans de nombreux cas, l'approche modulaire est une bonne solution. Les exemples dans lesquels l'approche modulaire est plus performante qu'un réseau unique sont nombreux (Guan et al. 1997; Lin et al. 1997).

Nous allons nous intéresser, dans la section suivante, à une approche modulaire particulière, les Mélanges d'Experts.

4 Les Mélanges d'Experts

La forme la plus populaire de modularité est appelée Mélanges d'Experts, et a été proposée il y a un dizaine d'années par (Jacobs et Jordan 1991a; Jacobs et al. 1991a; Jacobs et al. 1991b). Elle met en oeuvre un apprentissage coopératif, qui permet de traiter des problèmes complexes pour lesquels on ne dispose d'aucune connaissance a priori.

Les ME utilisent le principe de « diviser pour mieux régner ». Un problème donné est divisé en sous-problèmes plus simples. Ce système modulaire apprend à décomposer la tâche globale dans le sens où il met en compétition des réseaux de neurones différents pour apprendre des exemples provenant de différentes régions de l'espace d'entrée (ces régions peuvent même se chevaucher). Le résultat de cette division se traduit par le fait que les différents réseaux apprennent des exemples différents, ils apprennent donc des fonctions différentes. Les ME sont ainsi capables de décomposer une fonction et d'assigner à chaque fonction partielle le réseau dont la structure est la plus adaptée.

Dans cette section, nous expliquons le fonctionnement de ces modèles.

4.1 Description

L'architecture des ME se compose de deux types de réseaux neuronaux : plusieurs réseaux experts et un réseau superviseur (appelé aussi sélecteur ou "gating network" en anglais). Ce principe est montré sur la Figure 3-4. Les réseaux experts apprennent tous les mêmes exemples de la base d'apprentissage. Chaque expert mesure ses performances. Le réseau dont la réponse est la plus proche de la réponse désirée (c'est-à-dire le gagnant de la compétition) reçoit plus de "feedback", alors que les autres en reçoivent très peu. Le réseau superviseur pondère les sorties des experts de telle sorte que pour une entrée donnée, l'expert ayant la plus grande chance de fournir la bonne réponse obtient le poids le plus élevé.

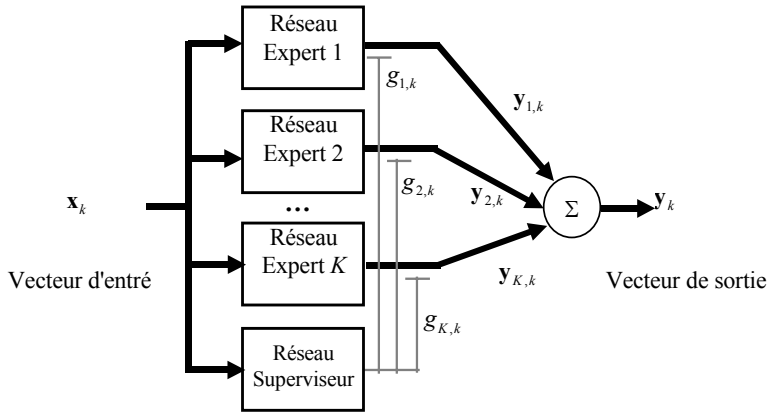


Figure 3-4 : Le principe du Mélange d'Experts.

Tous les experts prennent la même entrée, le réseau superviseur également. Dans certaines approches, le réseau superviseur peut prendre des entrées différentes que celles présentées aux experts (Jacobs et al. 1991a). Le réseau superviseur possède autant de sortie qu'il y a d'experts dans le mélange. L'algorithme original considérait le réseau superviseur ainsi que chaque expert comme un Perceptron multicouche. D'autres réseaux ont depuis été proposés et utilisés.

La fonction d'erreur est telle qu'il existe plutôt un phénomène de compétition qu'un phénomène de coopération entre des experts locaux. Même si la sortie globale est une combinaison linéaire des sorties des différents experts, le réseau superviseur doit choisir quel expert « croire » plus que les autres pour une entrée donnée. Le résultat d'un expert pour une entrée présentée n'est pas

directement affecté par les poids des autres experts locaux. Il y a cependant un lien indirect : si certains experts changent leurs poids, le réseau superviseur change alors les « responsabilités » assignés aux experts.

Pour situer l'approche des ME dans la classification proposée, nous dirons qu'elle est une forme de modularité vraie puisque chaque réseau expert se spécialise sur un sous-problème. Elle est basée sur une décomposition parallèle du type OU car chaque réseau expert reçoit la totalité du vecteur d'entrée et apprend à traiter différents exemples appartenant à différentes régions de l'espace d'entrée. L'apprentissage est coopératif : tous les modules – les réseaux experts – apprennent en même temps et se répartissent la tâche au cours de l'apprentissage. Il y a également une forme d'apprentissage supervisé dans l'approche des ME : un réseau auxiliaire – appelé réseau superviseur – contrôle les réseaux experts et coordonne les activités de l'ensemble de l'architecture.

La principale originalité de ce système réside dans le fait qu'il combine un apprentissage associatif (association d'entrées/sorties) et un apprentissage compétitif (compétition entre les experts) dans le but d'apprendre une décomposition de la tâche. Les experts entrent en compétition pour apprendre les exemples, et le réseau auxiliaire supervise cette compétition.

4.2 L'apprentissage

La démonstration de l'apprentissage des ME n'est pas reproduite dans ce document, on se référera à (Haykin 1994). Considérons l'architecture de la Figure 3-4, composée de K réseaux experts, avec les notations qu'elle comporte. Plus spécifiquement, le i-ème réseau expert approxime à l'itération courante k le problème par sa réponse $\mathbf{y}_{i,k}$ (c'est une relation linéaire par rapport au vecteur d'entrée) :

$$\mathbf{y}_{i,k} = \mathbf{w}_{i,k}^T \mathbf{x}_k , \qquad i = 1,2,...,K . \tag{3.1}$$

Le réseau superviseur est un réseau neuronal à couche unique avec K neurones, chacun étant assigné à un expert. Contrairement aux experts, les neurones du réseau superviseur sont non linéaires, leur sortie est scalaire et est donnée par :

$$g_{i,k} = \left(\mathrm{e}^{\mathbf{u}_{i,k}/T}\right)\left(\sum_{j=1}^{K} \mathrm{e}^{\mathbf{u}_{j,k}/T}\right)^{-1} , \qquad i = 1,2,...,K , \tag{3.2}$$

où T est un paramètre appelé température et où $\mathbf{u}_{i,k}$ est le produit vectoriel interne entre le vecteur d'entrée et le vecteur poids synaptiques $\mathbf{a}_{i,k}$ assigné au i-ème expert :

$$\mathbf{u}_{i,k} = \mathbf{x}_k^T \mathbf{a}_{i,k} , \qquad i = 1,2,...,K . \tag{3.3}$$

La transformation exponentielle normalisée (3.2) est appelée *softmax*. On trouvera une justification de cet outil très utilisé dans le domaine des statistiques dans (Bridle 1990; Yuille et Geiger 1995). La fonction softmax préserve l'ordre de grandeur des valeurs d'entrées et réalise une sélection par la valeur maximum. Cette fonction possède une interprétation probabilistique (Jordan et Jacobs 1995b). En effet, les sorties $g_{i,k}$ du réseau superviseur représentent les probabilités conditionnelles a priori que les sorties des experts estiment correctement une solution désirée $\mathbf{d}_k$ pour un vecteur d'entrée $\mathbf{x}_k$ connu. Pour ces probabilités, la fonction (3.2) garantie que

$$0 < g_i < 1 , \ \forall i , \text{ et } \sum_{i=1}^{K} g_i = 1 . \tag{3.4}$$

La sortie globale $\mathbf{y}_k$ est donnée par la combinaison linéaire des sorties des K experts du mélange :

$$\mathbf{y}_k = \sum_{i=1}^{K} g_{i,k} \mathbf{y}_{i,k} . \tag{3.5}$$

Chaque sortie $\mathbf{y}_{i,k}$ est pondérée par la sortie $g_{i,k}$ de réseau superviseur qui lui est associée au même instant. Ces pondérations sont une solution optimale d'une fonction d'erreur, le logarithme de la vraisemblance :

$$l(\mathbf{w},\mathbf{g}) = \ln\left(\sum_{i=1}^{K} g_{i,k}\, \mathrm{e}^{-\frac{1}{2\sigma_i^2}\left\|\mathbf{d}_k - \mathbf{y}_{i,k}\right\|^2}\right). \tag{3.6}$$

Dans cette fonction, il est évident que $\mathbf{y}_i$ dépend de $\mathbf{w}_i$. Le logarithme de la vraisemblance $l(\mathbf{w},\mathbf{g})$ peut donc être vu comme une fonction de coût, qui maximisée, aboutit aux estimations des paramètres latents de la structure modulaire, à savoir les poids synaptiques des experts et ceux du réseau superviseur, et cela au sens du maximum de vraisemblance.

Pour l'approximation, une probabilité a posteriori est introduite pour chaque expert :

$$h_{i,k} = \left(\frac{g_{i,k}}{\sigma_i} \mathrm{e}^{-\frac{1}{2\sigma_i^2}\left\|\mathbf{d}_k - \mathbf{y}_{i,k-1}\right\|^2} \right) \left(\sum_{j=1}^{K} \frac{g_{i,k}}{\sigma_j} \mathrm{e}^{-\frac{1}{2\sigma_j^2}\left\|\mathbf{d}_k - \mathbf{y}_{i,k-1}\right\|^2} \right)^{-1}, \quad i = 1,2,...,K . \tag{3.7}$$

Les σ_i sont des paramètres scalaires associés à chaque expert. Les probabilités h_i sont conditionnelles à la fois par rapport au vecteur d'entrée $\mathbf{x}_k$ et par rapport à la sortie désirée $\mathbf{d}_k$. Elles doivent satisfaire aux mêmes conditions que les probabilités $g_{i,k}$, à savoir,

$$0 < h_i < 1, \ \forall i, \text{ et } \sum_{i=1}^{K} h_i = 1. \tag{3.8}$$

Le gradient stochastique est généralement utiliser pour ajuster les paramètres d'apprentissage du ME, les poids synaptiques des experts ainsi que les poids synaptiques du réseau superviseur (il y a là un problème de régression et un problème de classification).

La dérivation de l'expression (3.6) permet de trouver les poids des réseaux. Ainsi les dérivées par rapport aux poids $\mathbf{w}_i$ et $\mathbf{a}_i$, $\partial l / \partial \mathbf{w}_i$ et $\partial l / \partial \mathbf{a}_i$, conduisent respectivement aux corrections suivantes :

$$\Delta \mathbf{w}_i = \eta \frac{\partial l}{\partial \mathbf{w}_i} = h_i \mathbf{e}_i \mathbf{x}^T, \quad i = 1,2,...,K , \tag{3.9}$$

$$\Delta \mathbf{a}_i = \eta \frac{\partial l}{\partial \mathbf{a}_i} = \eta (h_i - g_i) \mathbf{x}, \quad i = 1,2,...,K . \tag{3.10}$$

La loi d'apprentissage pour les poids des experts s'écrit donc :

$$\mathbf{w}_{i,k+1} = \mathbf{w}_{i,k} + \eta h_{i,k} \mathbf{e}_k \mathbf{x}_k^T, \quad i = 1,2,...,K . \tag{3.11}$$

La loi d'apprentissage des poids synaptiques des neurones du réseau superviseur se formule selon :

$$\mathbf{a}_{i,k+1} = \mathbf{a}_{i,k} + \eta \left(h_{i,k} - g_{i,k} \right) \mathbf{x}_k, \quad i = 1,2,...,K . \tag{3.12}$$

Pendant l'apprentissage, la montée de gradient effectuée sur $l(\mathbf{w},\mathbf{g})$ entraîne deux conséquences. Premièrement, pour un exemple donné, elle renforce les poids des experts qui prédisent mieux que la moyenne, et deuxièmement, elle aide les experts à mieux estimer la sortie désirée des exemples pour lesquels ils ont un poids élevé. Après apprentissage, le résultat se traduit par le fait que le réseau superviseur affecte toujours un poids proche de 1 à un expert pour chaque exemple. Chaque expert peut de cette façon « se concentrer » à ne modéliser que les exemples pour lesquels il est fait. Son apprentissage n'est d'aucune façon perturbé par les exemples qui ne lui conviennent pas. Ce principe permet de ne pas dégrader la mémoire d'un expert qui a déjà appris, comme cela pourrait se produire dans le cas où un seul réseau de neurones est utilisé pour apprendre un problème dans sa globalité. Les "outliers" (autrement dit des observations atypiques et peu fréquentes, des points qui semblent ne pas obéir aux caractéristiques d'une distribution représentative d'un nuage de points), au lieu d'affecter les poids des neurones d'un expert qui a déjà appris une zone, se voient pris en charge par un autre expert, plus apte à les considérer.

Le terme d'erreur, qui va servir à adapter les poids d'un expert, est pondéré par la probabilité a posteriori associée au *i*-ème expert. Pour un vecteur d'entrée, un seul expert possède une très forte probabilité a posteriori. Il en résulte qu'un exemple d'apprentissage n'est appris que par un seul expert à la fois.

4.3 Discussion

Le but de l'architecture ME est de modéliser la distribution des exemples d'apprentissage, autrement dit, de déterminer quelle règle a généré tel exemple. Ceci est réalisé par une montée de gradient sur la fonction logarithmique de la vraisemblance à générer un vecteur de sortie particulier $\mathbf{d}_k$.

Les ME concrétisent l'idée que la compétition peut conduire à une spécialisation fonctionnelle. Chaque module de l'architecture entre en compétition pour obtenir le droit d'apprendre à réaliser une sous-tâche de la tâche globale. Grâce à cette compétition, les modules sont obligés de se spécialiser : ils apprennent à accomplir des sous-tâches différentes.

Une approche proche des ME est le modèle AGM ("Associative Gaussian Mixture"), à l'origine développé pour expliquer les ME (Jacobs et Jordan 1991a) bien que le principe puisse être appliqué à d'autres réseaux connexionnistes. Une excellente synthèse sur les mélanges de fonctions Gaussiennes est faite dans (Haykin 1994). En fait, le principe du "divide and conquer" peut être mis en opposition au problème de superposition (de fonctions simples, du type Gaussiennes par exemple). Ces deux principes sont des principes de sélection WTA, et choisir entre les deux pour réaliser la supervision n'est pas trivial. D'après (Jordan et Jacobs 1995a), le premier converge plus rapidement.

Une alternative au gradient stochastique, pour réaliser l'apprentissage itératif du ME, est l'algorithme EM présenté dans le chapitre précédent. Son approche plus générique rend l'estimation de paramètres plus robuste (Jordan et Jacobs 1994; Xu et Jordan 1996).

Encore une fois, la contribution des méthodes statistiques est non négligeable dans l'approche modulaire des réseaux de neurones. Les statistiques sont confrontées depuis de nombreuses années à l'estimation de la valeur d'une variable aléatoire en combinant les estimations de plusieurs modèles (appelés des experts dans le cadre de l'approche ME). Des résultats empiriques, mais aussi des études théoriques plus récentes comme celles dans (Battiti et Colla 1994) ou dans (Mani 1991), ont établi que la combinaison des estimations non biaisées est la plus efficace lorsque les experts ne sont pas corrélés, et seulement peu efficace s'ils sont positivement corrélés.

Les ME adoptent une stratégie différente : au lieu de produire des estimateurs non biaisés, ils produisent des experts biaisés dont les estimations sont corrélées négativement (Jacobs 1997). Cette corrélation négative qui est observée, provient du fait qu'il y a un processus de compétition entre les experts, et que les ME partitionnent l'espace des entrées en régions (qui peuvent localement se superposer) telles que la fonction à approximer ait des propriétés différentes dans chacune d'elles.

Les inconvénients de l'approche ME concerne essentiellement son architecture : elle est fixe et déterminée préalablement alors qu'il est impossible de connaître par avance le nombre d'experts nécessaires pour la résolution d'un problème. L'expérience montre que si le mélange ne comporte pas assez d'experts, les résultats sont médiocres. En fait, le problème n'est pas assez simplifié avec un nombre insuffisant d'experts. Lorsqu'il y a beaucoup d'experts, le mélange n'utilise pas forcément tous les experts qui sont à sa disposition. Les experts en surplus vont jouer un rôle négligeable : pour tous les exemples, un poids de valeur proche de 0 leur est affecté. Le problème du nombre d'experts est souvent traité en autorisant un surplus d'expert (un ME surdimensionné), ou avec des approches évolutives.

4.4 Les extensions du modèle du Mélange d'Experts

Les apports concernant les modèles ME peuvent être, soit des extensions structurelles, soit des contributions théoriques. Nous allons introduire deux extensions structurelles classiques que sont les mélanges hiérarchiques et les mélanges évolutifs. Nous allons également faire un tour d'horizon des différentes contributions théoriques apportées ces dernières années.

4.4.1 Les mélanges hiérarchiques

Une conception naturelle consiste à remplacer les experts d'un mélange par des ME eux-mêmes pour aboutir à un Mélange d'Experts Hiérarchique, HME (Jordan et Jacobs 1994). La Figure 3-5 montre ce système arborescent. Les feuilles de l'arbres sont des réseaux d'experts simples. Cette architecture structurée en arbre permet de multiples subdivisions de l'espace des entrées.

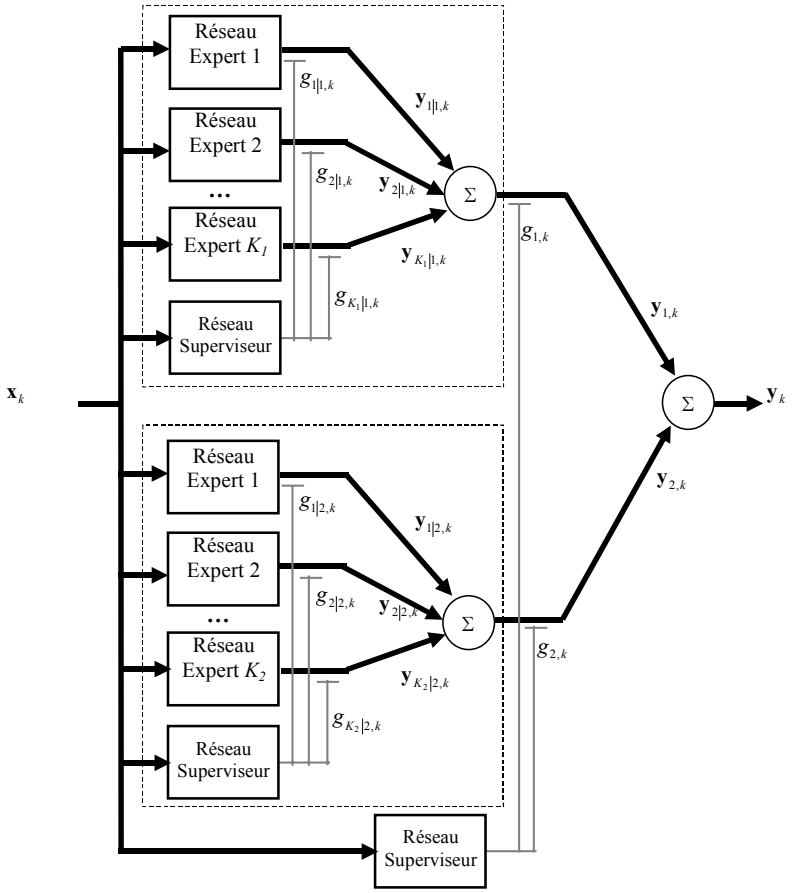


Figure 3-5 : Un Mélange d'Experts hiérarchique.
Deux mélanges composent les experts d'un troisième mélange considéré comme étant à un niveau supérieur.

On peut voir les ME comme un cas particulier des HME. L'algorithme d'apprentissage des HME est le plus souvent l'algorithme EM comme initialement proposé dans (Jordan et Jacobs 1995b). Dans (Waterhouse 1997) plusieurs extensions sont proposées.

L'inconvénient majeur des structures HME est évidemment la complexité du modèle résultant et les problèmes liés à la détermination de la structure hiérarchique optimale (nombre de modules ME, nombre d'experts par module, etc.). Aussi, l'algorithme EM est en général lent car chaque itération engendre un problème d'optimisation à résoudre. Finalement, il ne faut pas perdre de vue qu'étant d'inspiration Bayésienne, l'algorithme EM réalise de manière sous-jacente un mécanisme de sélection de l'expert le plus adéquat (i.e., le plus vraisemblable).

4.4.2 Les mélanges évolutifs

Pour répondre au problème du nombre d'experts que doit comporter un mélange pour apprendre un système, la majorité des travaux proposent des architectures ME évolutives.

Dans (Waterhouse et Robinson 1996), une méthode incrémentale est introduite. Cet algorithme se propose de construire progressivement, c'est-à-dire au cours de l'apprentissage, l'architecture d'un HME à partir d'une configuration initiale minimale (typiquement constituée de seulement deux experts).

Le HME est considéré comme un arbre, et un critère de vraisemblance permet, après un certain temps d'apprentissage, de diviser une feuille de l'arbre (un expert devient un mélange de deux experts). Après cette première phase qui fait grandir l'arbre, une seconde phase, en cherchant dans l'arbre le chemin le plus probable, supprime des branches de l'arbre, soit temporairement, soit irrémédiablement. Ce principe introduit une très grande flexibilité dans la structure du HME. Il a été évalué sur un problème de parité à 8 bits et sur le problème de la double spirale (2 spirales entrelacées qu'il s'agit de distinguer ou de séparer). Avec ces applications, un apprentissage plus rapide a été mis en évidence, ainsi qu'une utilisation plus efficace des paramètres. Ce concept tend vers le principe utilisé dans les approches utilisant l'algorithme CART ("Classification And Regression Trees").

Dans (Jacobs et al. 1997), ce problème est également abordé : une méthode est présentée pour détecter si l'architecture ME est trop petite, ou pour déterminer et localiser les experts superflus. Ceci est réalisé après la phase d'apprentissage. Un inconvénient persiste, puisqu'à chaque fois que l'architecture est modifiée, il faut ré-apprendre puis évaluer si la nouvelle taille est adéquate.

Une version incrémentale est développée dans (Rouzier 1998). MILSYS, "Modular Incremental Learning SYStem", est un système modulaire d'apprentissage coopératif et incrémental qui effectue une décomposition automatique d'un problème. Au départ, l'architecture n'est composée que d'un seul expert et d'un réseau superviseur. Des experts sont ensuite rajoutés tant que l'erreur n'est pas inférieure à un seuil fixé. Un expert est ajouté lorsque l'erreur ne baisse plus, ou après un certain nombre d'itérations. L'expert est ré-initialisé si, une fois inséré, il ne participe pas à l'élaboration de la sortie globale. On part de l'hypothèse que s'il n'est pas utilisé par le superviseur, c'est à cause d'une mauvaise initialisation. Le critère d'ajout d'un expert est difficile à manipuler, il dépend fortement de l'application. L'auteur procède, pour l'ajuster, par essai/erreur.

Cette approche est appliquée en simulation sur les problèmes bien connus du XOR et sur le problème du moine ("the monk's problem", qui est un problème standard en classification). MILSYS est également validé sur une application concrète d'association de piste (une application militaire qui est un problème particulier de fusion de données). Pour chaque application, MILSYS est systématiquement comparé à un réseau MLP, à un ME avec le même nombre d'experts, et à un ME avec un surplus d'experts.

Les inconvénient de MILSYS sont le réglage empirique du critère d'ajout d'un expert, et le fait qu'il doive se rendre compte sur plusieurs itérations que l'erreur stagne avant d'ajouter un expert supplémentaire. Il est par contre plus avantageux en terme de rapidité puisqu'il consomme moins de temps CPU qu'un ME sur-dimensionné. De plus, un ME, en fonction de l'initialisation des experts, se comporte de façon différente : il sépare les entrées de façon différente et chaque expert considère le problème de façon différente ; il semble que MILSYS non. La spécialisation fonctionnelle est la même quelle que soit l'initialisation des experts de MILSYS.

Ramamurti et Ghosh proposent d'adapter les structures des experts pour les ajuster dynamiquement à la complexité du problème (Ramamurti et Ghosh 1999). Dans cette approche, les réseaux experts sont des réseaux mono-couche, et le réseau superviseur réalise un apprentissage sur la base d'un algorithme EM itératif. Les avantages de cette approche proviennent à la fois d'une modélisation locale et du concept à modèle multiple. En cela elle s'approche des méthodes CART et MARS ("Multivariate Adaptive Regression Splines"). La différence marquante réside dans le fait qu'avec ces dernières, l'expert est entraîné sur l'ensemble de la base d'apprentissage, alors que l'approche ME réalise l'apprentissage des experts de manière locale, sur des régions de l'espace d'entrée. Comme application, les auteurs proposent un apprentissage en ligne de fonctions non

linéaires dans un environnement non stationnaire, et fournissent un exemple simulé avec la série temporelle de Mackey-Glass.

4.4.3 Extensions théoriques

Les extensions théoriques de l'approche ME sont nombreuses, et des apports de toutes sortes ont été effectués, provenant surtout des domaines extérieurs au connexionisme.

Ainsi après l'apparition des ME, des travaux proposaient d'utiliser les estimateurs statistiques hiérarchiques bien connus en statistiques. Parmi ces estimateurs, citons l'algorithme CART présenté dans (Breiman et al. 1984), ou les "voting methods with decision-tree learning algorithms" tel l'algorithme C4.5 de (Quinlan 1993), ou encore l'algorithme MARS dû à (Friedman 1991). L'utilisation des algorithmes CART ou MARS pour l'approche HME est reprise dans (Waterhouse 1997), respectivement page 23 et page 25. L'approche est également bien synthétisée dans (Haykin 1999), page 374.

Un avantage indéniable des ME, par rapport aux estimateurs statistiques hiérarchiques, provient du fait que l'espace des entrées est découpé en régions qui peuvent se recouper et se recouvrir (on parle, dans la littérature anglophone de "soft split" et de "hard split" pour la notion inverse). Les estimateurs qui utilisent des "soft splits" ont des chances plus grandes d'avoir une faible variance que ceux qui utilisent des "hard splits" (Jordan et Jacobs 1994).

Un autre type de contribution provient de (Xu et al. 1992) et (Rogova 1994). Cette approche utilise la théorie des possibilités, la théorie de l'évidence ou théorie de Dempster-Shafer. C'est un outil mathématique de représentation et de combinaison de mesures d'évidences, une généralisation du raisonnement Bayésien, en fait plus flexible lorsque les connaissances sont incomplètes ou lorsqu'elles sont en présence d'incertitudes.

Wolpert utilise des réseaux RBF (Ghahramani et Wolpert 1997). Par rapport au développement précédent, le calcul des pondérations change quelque peu :

$$g_{i,k} = \left(\mathrm{e}^{S_i(\mathbf{x}_k,\boldsymbol{\theta})}\right)\left(\sum_{j=1}^{K} \mathrm{e}^{S_j(\mathbf{x}_k,\boldsymbol{\theta})}\right)^{-1}, \quad i = 1,2,...,K\ , \tag{3.13}$$

où $s_i(.)$ est une fonction (du type RBF) et $\boldsymbol{\theta}$ un vecteur paramètre.

On constate en effet que ces modèles résultent d'une superposition de composants élémentaires simples (les experts) ayant un champ d'action localisé. Ces experts locaux peuvent par exemple modéliser une densité de probabilité (c'est parfaitement le cas des ME neuronaux), une fonction déterministe (modèles RBF) ou éventuellement aléatoire (modèles SRM, "Switching Regression Models"). Cette dernière caractéristique n'est pas souhaitable en approximation, puisqu'en général des comportements différents de la fonction à approcher sont clairement localisables dans l'espace d'entrée.

La logique floue n'est pas en reste puisqu'elle s'est également appropriée le concept du ME (Akhmetov et al. 2001; Wang et al. 1995). La principale différence avec l'approche neuronale se situe au niveau des pondérations du réseau superviseur. Contrairement aux réseaux neuronaux qui pondèrent les estimations des experts par des valeurs scalaires, les approches floues utilisent quant à elles des fonctions multivariables du type B-splines, Gaussiennes ou sigmoïdales (comme la fonction $s_i(.)$ dans (3.13)). Dans (Gan et Harris 2001) une approche hybride est proposée avec des FLL ("Fuzzy Local Linearization") et l'algorithme EM qui sert à paramétrer les modèles linéaires locaux. Le fait d'utiliser des fonctions est sensé rendre l'approximation de fonctions "smooth" plus facile car le problème est mieux conditionné (Fiordaliso 1999).

La différence essentielle entre l'approche floue (ou l'approche qui utilise les réseaux RBF) et les méthodes classiques, réside dans la volonté de considérer les degrés d'appartenance des individus aux classes dans l'écriture du critère à minimiser. Une classe représente un expert et les degrés d'appartenance sont traduits par les fonctions $s_i(.)$.

Il existe des méthodes qui ne respectent pas la seconde contrainte de (3.4) et qui possèdent l'avantage (par rapport aux méthodes contraintes) de pouvoir identifier les points éloignés de tout "cluster" (des "outliers"). Ceci est impossible pour les méthodes contraintes à cause de la normalisation (3.13). L'inconvénient majeur des méthodes non contraintes est les convergences possibles vers des partitions dégénérées ou inexploitables, pour lesquelles tous les "clusters" obtenus sont approximativement identiques au niveau de la forme et de la position. Cet inconvénient n'apparaît pas pour les algorithmes où la contrainte est active, car celle-ci favorise la séparation des "clusters".

Pour finir, l'extension la plus récente est sans doute celle de Yin et de Allinson (Yin et Allinson 2001). Ces derniers proposent un réseau SOMN ("Self-Organizing Mixture Network") pour estimer une densité de probabilité. Le concept du réseau SOMN est à mi-chemin entre une carte auto-organisatrice et un "mixture distribution model". Cette dernière approche est proche des ME mais ne reprend pas complètement leur principe. La supervision n'est pas réalisée par un réseau auxiliaire mais le rôle de chaque expert est déterminé par une fonction à noyau qui lui est associée. L'originalité du réseau SOMN réside dans la coexistence de la notion de voisinage de la carte auto-organisatrice et des probabilités conditionnelles estimées par des fonctions Gausiennes. Les neurones de la carte sont des experts pour le mélange. Chaque neurone est à la fois caractérisé par des grandeurs géométriques (dans une grille topologique) et par des propriétés statistiques.

4.5 Conclusion

Les ME sont la forme de modularité la plus populaire dans la communauté des réseaux de neurones artificiels. Ils sont une forme de modularité vraie. Les réseaux experts sont des modules qui se spécialisent sur une partie seulement d'un problème. Ensemble, ils sont capables d'apprendre le problème dans son intégralité. L'apprentissage des réseaux experts s'effectue en même temps et la répartition des tâches est réalisée au cours de l'apprentissage. C'est un apprentissage coopératif, qui est assujetti à un apprentissage supervisé. Un réseau superviseur supplémentaire contrôle l'apprentissage de tous les réseaux experts.

Les expressions récurrentes de la règle d'apprentissage ont été obtenues parce que les densités de probabilité associées à chaque réseau expert sont supposées Gaussiennes (Jordan et Jacobs 1995b). Une densité de probabilité différente induit le calcul d'une nouvelle règle d'apprentissage.

Le principe général de l'approche des ME est astucieux : c'est une combinaison d'un apprentissage associatif et d'un apprentissage compétitif. En d'autres termes, cette architecture modulaire permet d'apprendre une association d'entrées et de sorties, donc une transformation entre deux espaces, et réalise une compétition entre les réseaux experts pour améliorer son modèle de connaissance.

La combinaison linéaire optimale de réseaux de neurones de (Hashem 1997) peut sembler être une approche très proche de celle des ME. Cependant, l'approche MSE-OLC ("Mean Square Error Optimal Linear Combinations") suppose que les modules résolvent le même problème. Elle est une approche à modèle multiple, alors que l'approche des ME concentre ses modules sur des parties du problème. Ce dernier point est réalisé grâce aux capacités d'apprentissage et de mémorisation du réseau superviseur, et la problématique auquel il est confronté est un problème de "clustering". Il est aisé de se rendre compte que des méthodes statistiques peuvent réaliser la supervision des réseaux experts. Ainsi, la frontière entre l'approche ME, une forme de modularité vraie, et l'approche à modèle multiple, n'est parfois pas évidente.

L'approche ME permet de façon logique et naturelle, l'estimation de densité de probabilité (Moerland 1997). Cette faculté en fait un outil de classification performant. L'approche des ME ne s'est cependant pas imposée dans le domaines de la robotique. Les applications à la robotique sont même rares. Parmi ces travaux, citons ceux de Ghahramani et Wolpert (Ghahramani et Wolpert 1997), ceux de Jacobs et Jordan (Jacobs et Jordan 1991b), et ceux de Gomi et Kawato (Gomi et Kawato

1993). Ces derniers appliquent bien l'approche ME au contrôle robotique, mais effectuent une reconnaissance des objets manipulés pour paramétrer le contrôleur. Deux approches sont présentées, et les deux se distinguent des ME originaux par le fait que le réseau superviseur possède un vecteur d'entrée différent de celui des réseaux experts. Jacobs et Jordan (Jacobs et Jordan 1993; Jordan et Jacobs 1994), ont montré dans le problème de l'identification d'un système non linéaire, le modèle cinématique direct d'un robot 4 axes, qu'un HME converge deux fois plus rapidement que la règle BP d'un réseau multicouche de taille comparable. Les deux approches sont du type "Adaptive Gain Scheduling" (GS). Les auteurs les utilisent en y incluant en plus des informations a piori. Lorsque l'approche des ME est utilisée pour effectuer une tâche de commande, c'est pour ses capacités à compenser la non stationnarité de l'environnement ou des systèmes (Cacciatore et Nowlan 1994).

5 Conclusion du chapitre

Il a été montré qu'en théorie un réseau multicouche est capable d'approcher toute fonction continue s'il dispose d'un nombre suffisant de neurones cachés. Les réseaux multicouches sont des approximateurs universels, capables de reproduire les comportements les plus difficiles avec un nombre suffisant de couches et de neurones. Dans la pratique, un comportement complexe suppose une taille trop importante et bien souvent irréaliste.

Pour ne pas laisser leur taille croître indéfiniment avec la complexité de la tâche à apprendre, et pour chercher à accroître leurs performances, on peut penser introduire explicitement plus de connaissances dans les réseaux de neurones artificiels. Introduire plus de connaissances dans chaque unité neuronale revient à concevoir des modèles de neurones plus sophistiqués, contenant plus de paramètres. Or, il se trouve que plus un neurone comprend de paramètres, plus son apprentissage est difficile et plus la convergence de l'apprentissage vers une solution idéale est hypothétique.

Fournir de l'information dans la structure d'un réseau de neurones artificiels peut se faire à travers la notion de modularité. Le découpage d'un réseau en modules est une façon d'organiser et de structurer de la connaissance. L'organisation des modules ne doit pas être laissée au hasard. Elle est une façon de stocker de l'information uniquement lorsqu'elle est heuristique. L'organisation n'est-elle pas une forme d'intelligence (Bonabeau et Theraulaz 1994) ?

Nous avons défini un module neuronal comme un élément ayant une fonction interprétable et pertinente au sein d'un système d'apprentissage. De cette définition découle la micro-modularité, elle concerne une unité neuronale, et la macro-modularité qui concerne les réseaux neuronaux et les réseaux de réseaux neuronaux. Nous nous sommes intéressé à la macro-modularité, que nous avons séparée en deux catégories, la modularité à modèle multiple et la modularité vraie. Les modules de la première apprennent tous le même problème et une technique de fusion des sorties de tous les modules fournit la sortie globale de l'architecture modulaire. Les modules de la seconde forme de modularité se partagent et n'apprennent que les sous-problèmes ; mis ensembles ils disposent d'une connaissance du problème entier.

Si la modularité existe ailleurs (sous les deux aspects, à modèle multiple et modularité vraie), dans d'autres sciences comme le traitement du signal ou l'automatique, la modularité à modèle multiple est confrontée aux mêmes problèmes que dans ces sciences connexes, alors que la modularité vraie est plus difficilement applicable. La contrainte vient de l'apprentissage des modules. Lorsqu'une tâche est divisée en sous-tâches, pour les apprendre il faut disposer soit des entrées, soit des entrées et des sorties de ces sous-tâches, pour pouvoir construire les bases d'apprentissages des réseaux. Or, il est souvent difficile, voire impossible de disposer de ces données, qui sont souvent internes au système et donc inaccessibles.

Les approches à modèle multiple visent essentiellement un gain de performances en réalisant un apprentissage multiple grâce à plusieurs modules. Une combinaison des réponses de plusieurs systèmes d'apprentissage conduit logiquement à une réponse plus fiable. Il est intéressant de remarquer

que ce principe va à l'encontre des méthodes Bayesiennes qui consistaient jusqu'alors à isoler, donc à choisir le modèle le plus vraisemblable sur une base de données passées.

La modularité vraie cherche plutôt à arriver à des tailles de réseaux moindres en tentant de reproduire un découpage logique du problème. La modularité vraie n'est pas une approche endogène. Plus particulièrement, si un réseau de neurones unique ne peut pas résoudre tout un problème, alors il faut le diviser en sous-problèmes et les résoudre localement car ils sont alors plus simples. Un exemple de cette démarche est donné par la reconnaissance de la parole (Gallinari 1995), où l'on utilise une structure hiérarchique de modèles. Un groupe de mots, une proposition ou une phrase sont des structures. S'il est possible de les extraire et de les reconnaître, on peut ensuite se concentrer sur la reconnaissance des mots eux-mêmes. On dispose alors d'une information supplémentaire, à savoir la place de chaque mot dans la phrase.

La littérature qui s'est accumulée depuis quelques années ne permet pas d'établir clairement la supériorité d'une approche modulaire vis-à-vis d'une autre. En fait, il est un principe bien admis selon lequel il n'existe pas de « meilleur » modèle, i.e., un modèle meilleur que ses concurrents en toutes circonstances, d'autant plus que le critère permettant de décider qu'un modèle est meilleur qu'un autre est toujours discutable.

Une forme de modularité particulièrement intéressante est donnée par l'approche des ME. Cette approche de modularité vraie est composée d'un certains nombre de réseaux experts dont le rôle est d'apprendre les sous-tâches d'une tâche difficile. Un réseau superviseur contrôle et force chaque réseau expert à une spécialisation sur la sous-tâche qui lui a été assignée.

Son principal inconvénient est l'impossibilité de connaître à l'avance le nombre de réseaux experts nécessaires pour apprendre un problème. On est obligé de procéder par essai/erreur, sachant qu'au delà d'une certaine taille, les performances restent stables. Cette approche a cependant de nombreux avantages. Son apprentissage est automatisé, les réseaux experts se répartissent seuls sur les sous-problèmes grâce au module superviseur et ses capacités d'apprentissage. Cette automatisation évite de devoir formuler les composantes structurelles d'un réseau modulaire comme l'exige les autres approches de modularité vraie. Elle réalise un apprentissage de la transformation directe de l'espace des entrées vers l'espace des sorties sans nécessiter de grandeurs intermédiaires, et surtout, les ME sont capables de faire face à la non stationnarité des signaux. Cette propriété provient de l'architecture même des ME, conçue de sorte à estimer la probabilité que chaque expert fournisse la solution optimale pour une entrée donnée.

Les caractéristiques et les avantages des ME ont décidé notre choix dans l'utilisation par la suite de cette approche pour la conception d'un estimateur modulaire.

Chapitre 4

Estimation par le filtrage de Kalman adaptatif

1 Introduction

Le filtre de Kalman est un algorithme très prisé pour estimer l'état d'un processus ou d'un système. Sa formulation récursive est particulièrement adaptée au calcul numérique, mais surtout, en plus d'une loi d'évolution déterministe, il inclut la notion d'incertitude. Les modèles de bruit, des composantes stochastiques, servent à tenir compte d'un certain nombre de paramètres qu'il n'est possible aujourd'hui de considérer que par des modèles statistiques. On peut y inclure les erreurs de modélisation, les imprécisions et imperfections d'ordre divers, les bruits de mesures, etc.

Pour les raisons évoquées ci-dessus, le KF est couramment appliqué dans les travaux d'asservissement visuel ou de vision active (Hashimoto 1993; Wilson et al. 1996; Wilson 1993). Son utilisation implique obligatoirement une expression analytique des modèles considérés. Le manque de connaissance a priori peut, dans une certaine limite, être compensé par les approches adaptatives du filtre. La plupart des algorithmes sont cependant, soit difficiles à mettre en œuvre, soit nécessitent une puissance de calcul qui n'est pas compatible avec des applications temps réel.

Nous proposons une alternative à ces méthodes à travers un filtre de Kalman adaptatif au sens de la représentation d'état, un SMAKF ("State Model Adaptive Kalman Filter"). Le modèle d'état du filtre, la matrice de transition qui définit la loi d'évolution dynamique du système, est adapté à l'aide du vecteur d'observation au cours du temps.

Après avoir exposé la règle d'adaptation de la matrice de transition, et après une analyse expérimentale de quelques résultats de stabilité, nous proposons un schéma d'estimation plus général. Nous développons une approche à modèle multiple basée sur le principe des ME, où plusieurs SMAKF estiment les mêmes signaux. Un processus d'apprentissage auxiliaire, en approximant la probabilité de chaque filtre de délivrer l'estimation optimale en connaissant les entrées, supervise la participation de chaque filtre. L'estimation globale est alors une somme des estimations des filtres pondérés par la probabilité qui leur est associée. Cette approche modulaire est capable de compenser les propriétés non stationnaires des systèmes ou des signaux pour des tâches d'estimation et de prédiction.

Enfin, une dernière approche est proposée pour réaliser l'apprentissage des transformations entre les espaces de la représentation d'état. Cette approche se justifie essentiellement pour modéliser un système totalement inconnu tout en estimant sa représentation interne.

2 Le filtre de Kalman adaptatif au sens du modèle d'état

2.1 Adaptation du modèle d'état

Nous allons dans cette section, développer notre propre filtre de Kalman adaptatif. En Partant d'une hypothèse statistique, notre démarche conduit à une règle d'adaptation de la matrice de transition du filtre. Le filtre adaptatif qui en résulte est une approche tout à fait générique, qui peut être appliqué à tout système sans modèle ou non linéaire.

Considérons le processus d'évolution suivant, ainsi que la loi d'observation suivante :

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{D}\mathbf{w}_k , \tag{4.1}$$

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k . \tag{4.2}$$

Ce sont les équations (2.21) et (2.22), qui sont reproduites ici dans un souci de clarté. Les hypothèses faites auparavant sur le bruit du processus $\mathbf{w}_k$ et sur le bruit de mesure $\mathbf{v}_k$, sont reprises pour les développements futurs, notamment :

$$E[\mathbf{w}_k \mathbf{w}_l^T] = \begin{cases} \mathbf{Q}_k & l = k \\ 0 & l \neq k \end{cases}, \tag{4.3}$$

$$E[\mathbf{v}_k \mathbf{v}_l^T] = \begin{cases} \mathbf{R}_k & l = k \\ 0 & l \neq k \end{cases}, \tag{4.4}$$

$$E[\mathbf{w}_k \mathbf{v}_l^T] = 0 . \tag{4.5}$$

Les hypothèses (4.3) et (4.4) reviennent respectivement à écrire $\mathbf{w}_k \sim N(0,\mathbf{Q})$ et $\mathbf{v}_k \sim N(0,\mathbf{R})$. Les bruits sont définis dans les espaces suivants, $\mathbf{w}_k \in \mathbb{R}^s$ et $\mathbf{v}_k \in \mathbb{R}^p$; et leur matrice de covariance sont de dimensions : $\mathbf{Q} \in \mathbb{R}^{s\times s}$ et $\mathbf{R} \in \mathbb{R}^{p\times p}$.

Pour le modèle donné, rappelons que $\mathbf{x}_k \in \mathbb{R}^n$ est le vecteur d'état du processus et que le vecteur $\mathbf{z}_k \in \mathbb{R}^p$ est la mesure disponible à l'instant *k*. Les paramètres qui définissent le modèle sont donc les matrices de dimensions suivantes : $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{D} \in \mathbb{R}^{n\times s}$ et $\mathbf{C} \in \mathbb{R}^{p\times n}$.

L'objectif recherché est d'approcher aussi fidèlement que possible l'état $\mathbf{x}_k$ du système à chaque instant *k*. Nous nous proposons de développer une méthode adaptative d'estimation des paramètres du filtre de Kalman capable d'estimer et de prédire la valeur courante du vecteur d'état. Cette méthode est basée sur la formulation itérative du filtre de Kalman, elle reprend donc les expressions (2.23) à (2.27).

Supposons d'abord que l'évolution du processus n'est gouvernée que par une part stochastique, ainsi $\mathbf{u}_k = 0$ pour les instants discrets *k*. On cherche à déterminer une expression de l'estimée $\hat{\mathbf{A}}_k$ de la matrice de transition à chaque instant *k*.

Pour cela, reprenons l'expression (4.1). Avec une linéarisation locale dans l'espace temporel, et sachant que $E\left[\mathbf{w}_k\right] = 0$, on peut écrire que

$$\hat{\mathbf{A}}_k = \mathbf{x}_k \left(\mathbf{x}_{k-1}^T \mathbf{x}_{k-1}\right)^{-1} \mathbf{x}_{k-1}^T . \tag{4.6}$$

Cette estimation dépend essentiellement du vecteur d'état, de ces deux dernières valeurs plus exactement.

Dans la loi d'observation (4.2), si la contribution du bruit $\mathbf{v}_k$ peut être considérée comme négligeable sur une période de temps grande ($E\left[\mathbf{v}_k\right] = 0$), alors

$$\mathbf{x}_k = \hat{\mathbf{C}}_k^+ \mathbf{z}_k . \tag{4.7}$$

La matrice de sortie $\mathbf{C}_k$ doit être une matrice non singulière, son inverse doit exister et nous l'avons notée $\hat{\mathbf{C}}_k^+$.

L'estimation de la matrice de transition peut dans ce cas s'exprimer :

$$\hat{\mathbf{A}}_k = \hat{\mathbf{C}}_k^+ \mathbf{z}_k \left(\left(\hat{\mathbf{C}}_{k-1}^+ \mathbf{z}_{k-1}\right)^T \left(\hat{\mathbf{C}}_{k-1}^+ \mathbf{z}_{k-1}\right)\right)^{-1} \left(\hat{\mathbf{C}}_{k-1}^+ \mathbf{z}_{k-1}\right)^T , \tag{4.8}$$

ou plus simplement

$$\hat{\mathbf{A}}_k = \hat{\mathbf{C}}_k^+ \mathbf{z}_k \left(\mathbf{z}_{k-1}^T \hat{\mathbf{C}}_{k-1}^{+\,T} \hat{\mathbf{C}}_{k-1}^+ \mathbf{z}_{k-1} \right)^{-1} \mathbf{z}_{k-1}^T \hat{\mathbf{C}}_{k-1}^{+\,T} . \tag{4.9}$$

La formulation précédente fait intervenir les dimensions de la matrice $\mathbf{C}$. Rappelons que $\mathbf{C} \in \mathbb{R}^{p*n}$ où p représente la taille du vecteur de sortie $\mathbf{z}_k$ et n est la dimension du vecteur d'état $\mathbf{x}_k$.

Nous allons, dans la suite du raisonnement, distinguer trois cas : lorsque $p < n$, lorsque $p > n$ et lorsque $p = n$.

1er cas, $p < n$: on se place dans le cas où le vecteur de sortie comporte plus d'éléments que le vecteur d'état. En prenant les pseudo-inverses à droite de Moore-Penrose pour le calcul de $\hat{\mathbf{C}}_k^+$ et de $\hat{\mathbf{C}}_{k-1}^+$, et après simplification, il vient :

$$\hat{\mathbf{A}}_k = \left(\mathbf{z}_{k-1}^T \mathbf{G}_{k-1}^{-T} \mathbf{z}_{k-1} \right)^{-1} \hat{\mathbf{C}}_k^T \mathbf{G}_k^{-1} \mathbf{z}_k \mathbf{z}_{k-1}^T \mathbf{G}_{k-1}^{-T} \hat{\mathbf{C}}_{k-1} , \tag{4.10}$$

en posant $\mathbf{G}_k = \mathbf{G}_k^T = \hat{\mathbf{C}}_k \hat{\mathbf{C}}_k^T$ et $\mathbf{G}_{k-1} = \mathbf{G}_{k-1}^T = \hat{\mathbf{C}}_{k-1}^T \hat{\mathbf{C}}_{k-1}$. Les matrices $\mathbf{G}_k$ et $\mathbf{G}_{k-1}$ sont supposées ne pas être des matrices singulières de sorte qu'elles soient inversibles. $\mathbf{G}_k$ est une matrice carrée, $\mathbf{G}_k \in \mathbb{R}^{p*p}$, $\forall k$.

2ème cas, $p > n$: dans ce cas le vecteur d'état contient plus de données que le vecteur de sortie. En utilisant cette fois les pseudo-inverses à gauche de Moore-Penrose pour le calcul de $\hat{\mathbf{C}}_k^+$ et de $\hat{\mathbf{C}}_{k-1}^+$, en simplifiant, il vient :

$$\hat{\mathbf{A}}_k = \left(\mathbf{z}_{k-1}^T \hat{\mathbf{C}}_{k-1} \mathbf{G}_{k-1}^{-1} \mathbf{G}_{k-1}^{-1} \hat{\mathbf{C}}_{k-1}^T \mathbf{z}_{k-1} \right)^{-1} \mathbf{G}_k^{-1} \hat{\mathbf{C}}_k^T \mathbf{z}_k \mathbf{z}_{k-1}^T \hat{\mathbf{C}}_{k-1} \mathbf{G}_{k-1}^{-1} , \tag{4.11}$$

avec cette fois $\mathbf{G}_k = \mathbf{G}_k^T = \hat{\mathbf{C}}_k^T \hat{\mathbf{C}}_k$, $\mathbf{G}_{k-1} = \mathbf{G}_{k-1}^T = \hat{\mathbf{C}}_{k-1}^T \hat{\mathbf{C}}_{k-1}$ et $\mathbf{G}_k \in \mathbb{R}^{n*n}$, $\forall k$. Encore une fois, les matrices $\mathbf{G}_k$ et $\mathbf{G}_{k-1}$ sont supposées être inversibles ou ne pas être des matrices singulières.

3ème cas, $p = n$: les vecteurs d'état et de sortie ont la même taille, ils contiennent le même nombre d'informations. La matrice $\mathbf{C}$, qui relie ces deux espaces (l'espace de sortie et l'espace d'état) est alors une matrice carrée. Si elle n'est pas singulière, les matrices estimées s'inversent avec $\hat{\mathbf{C}}_k^+ = \hat{\mathbf{C}}_k^{-1}$ et $\hat{\mathbf{C}}_{k-1}^+ = \hat{\mathbf{C}}_{k-1}^{-1}$. Cette fois la non singularité porte sur les matrices $\hat{\mathbf{C}}_k$ et $\hat{\mathbf{C}}_{k-1}$. Ceci conduit l'expression (4.9) de l'estimation de la matrice de transition à se reformuler selon l'équation suivante :

$$\hat{\mathbf{A}}_k = \left(\mathbf{z}_{k-1}^T \hat{\mathbf{C}}_{k-1}^{-1\,T} \hat{\mathbf{C}}_{k-1}^{-1} \mathbf{z}_{k-1} \right)^{-1} \hat{\mathbf{C}}_k^{-1} \mathbf{z}_k \mathbf{z}_{k-1}^T \hat{\mathbf{C}}_{k-1}^{-1\,T} . \tag{4.12}$$

Dans les expressions (4.10), (4.11) et (4.12), les termes entre parenthèses sont des grandeurs scalaires, leur inverse se calcule donc aisément. De plus, si dans la loi d'observation on a $\mathbf{C} = \mathbf{I}_n$, alors la formulation (4.12) de l'estimation de $\mathbf{A}_k$ se simplifie encore pour le cas $p = n$:

$$\begin{aligned} \hat{\mathbf{A}}_k &= \mathbf{z}_k \left(\mathbf{z}_{k-1}^T \mathbf{z}_{k-1} \right)^{-1} \mathbf{z}_{k-1}^T \\ &= \left(\mathbf{z}_{k-1}^T \mathbf{z}_{k-1} \right)^{-1} \mathbf{z}_k \mathbf{z}_{k-1}^T \end{aligned} \tag{4.13}$$

Dans cette dernière expression $\mathbf{z}_{k-1}^T \mathbf{z}_k$ est un produit vectoriel interne (autrement dit un produit scalaire qui mesure l'orthogonalité entre deux vecteurs, $\mathbf{z}_{k-1}^T \perp \mathbf{z}_{k-1} \Leftrightarrow \mathbf{z}_{k-1}^T \mathbf{z}_{k-1} = 0$) et $\mathbf{z}_k \mathbf{z}_{k-1}^T$ est un produit vectoriel externe. En fait, ce dernier terme se retrouve également dans les expressions (4.10), (4.11) et (4.12). Le résultat d'un produit vectoriel externe est une matrice dont chaque élément détaille à quel point les différents éléments des vecteurs sont corrélés.

La signification de matrice de transition n'est pas si éloignée de cette définition. Les éléments de la matrice $\mathbf{A}_k$ représentent la dynamique du système, c'est-à-dire le lien entre deux vecteurs d'états successifs dans l'espace temporel.

D'après les expressions (4.10), (4.11) et (4.12), on voit que l'estimation de la matrice de transition dépend des deux dernières valeurs mesurées, des observations aux instants k et k-1. L'estimation $\hat{\mathbf{A}}_k$ est continuellement ajustée au cours de l'évolution du processus. Ceci permet de tenir compte des lentes dérivations des paramètres du système et permet, de manière plus générale, de tenir compte de tous les changements dans la dynamique du système, voire même de changements de comportement. Le système à estimer peut également être un système non linéaire. Nous verrons dans

nos résultats, que l'algorithme proposé est capable de tenir compte naturellement des non linéarités d'un système.

Parce que le filtre utilise une matrice de transition continuellement ajustée et que cette matrice définit l'espace d'état, le filtre qui utilise ce modèle estimé est appelé "State Model Adaptive Kalman Filter" ou SMAKF.

Précisons qu'il n'est pas juste de parler d'apprentissage de la matrice de transition dans le cas du SMAKF, mais plutôt d'adaptation avec le sens défini en annexe.

Ce filtre, adaptatif au sens de la matrice d'état, est un filtre sous-optimal. La matrice de transition est approchée, ce qui signifie que le modèle utilisé par le filtre n'est pas une connaissance parfaite de l'évolution du processus. Compenser ce manque de savoir, peut se faire par le biais des bruits. Une technique optimale d'estimation des matrices de covariances $\mathbf{Q}_k$ et $\mathbf{R}_k$ permettra d'accroître la précision des estimations.

2.2 Estimation des caractéristiques des bruits

Nous avons choisi d'estimer les matrices $\mathbf{Q}_k$ et $\mathbf{R}_k$ itérativement selon une méthode optimale de correction de covariance (Moghaddamjoo 1989). Elle est une suite des travaux de Myers et de Tapley (Myers et Tapley 1976). Cette méthode n'est valide que lorsque l'estimation du vecteur d'état est proche de la vraie valeur de la grandeur à estimer. Des changements brusques dans les entrées dégradent rapidement l'estimation $\hat{\mathbf{x}}_k$ du vecteur d'état qui entraîne lui, une détérioration des matrices estimées de $\mathbf{Q}_k$ et $\mathbf{R}_k$.

Pour éviter la dégradation des performances, Moghaddamjoo recommande de n'utiliser sa méthode que sur des périodes limitées dans le temps, ou de réinitialiser l'estimateur régulièrement. Cette méthode se base sur la connaissance théorique de la matrice de transition. Nous l'avons modifiée afin qu'elle intègre la matrice de transition estimée par la relation (4.9).

Les erreurs suivantes sont définies dans l'espace d'état :

$$\mathbf{f}_k = \hat{\mathbf{x}}_k - \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} . \tag{4.14}$$

Plutôt que d'utiliser une définition de l'erreur utilisant le modèle théorique de la dynamique du système, nous lui préférerons une définition incluant l'estimation de la dynamique. Ainsi $\mathbf{f}_k$ est définie par :

$$\mathbf{f}_k = \hat{\mathbf{x}}_k - \hat{\mathbf{A}}_{k-1}\hat{\mathbf{x}}_{k-1} . \tag{4.15}$$

L'algorithme permet d'approcher les matrices de covariance $\mathbf{Q}_k$ et $\mathbf{R}_k$, ainsi qu'une estimation $\hat{\mathbf{u}}_k$ des signaux d'entrée. Pour les N_s itérations les plus récentes et en utilisant la médiane (med), on peut écrire :

$$\hat{\mathbf{u}}_k = \text{med}\ \{\ \mathbf{f}_k\}_{N_s} . \tag{4.16}$$

La matrice estimée $\hat{\mathbf{Q}}_k$ du bruit du processus est donnée par :

$$\hat{\mathbf{Q}}_k = \hat{\mathbf{q}}_k^{(ij)} - \frac{1}{N_s}\sum_{l=1}^{N_s}\left(\hat{\mathbf{A}}_{k-l}\mathbf{P}_{k-l}\hat{\mathbf{A}}_{k-l}^T - \mathbf{P}_{k-l+1}\right), \tag{4.17}$$

avec les $\hat{\mathbf{q}}_k^{(ij)}$ définis selon

$$\hat{\mathbf{q}}_k^{(ij)} = \frac{N_s\sum_l\left(\mathbf{f}_{k-l+1}^i - \hat{\mathbf{u}}_k^i\right)\left(\mathbf{f}_{k-l+1}^j - \hat{\mathbf{u}}_k^j\right)\left(1-M_{ij}^2(l)\right)^4}{\left(\sum_l\left(1-M_{ij}^2(l)\right)\left(1-5M_{ij}^2(l)\right)\right)\left(-1+\sum_l\left(1-M_{ij}^2(l)\right)\left(1-5M_{ij}^2(l)\right)\right)}, \tag{4.18}$$

où $\sum_l$ désigne la somme pour tous les l pour qui $M_{ij}^2(l) \le 1$. Les termes $\mathbf{f}_k^i$ et $\hat{\mathbf{u}}_k^i$, quant à eux, représentent les i-ème composants, respectivement de $\mathbf{f}_k$ et de $\hat{\mathbf{u}}_k$. Les termes $M_{ij}^2(l)$ sont donnés par :

$$M_{ij}^2(l) = \frac{\left|\mathbf{f}_{k-l+1}^i - \text{med}\ \{\ \mathbf{f}_l^i\}\right|}{6\ \text{med}\ \left\{\ \left|\mathbf{f}_m^i - \text{med}\ \{\ \mathbf{f}_m^i\}\right|\right\}} \cdot \frac{\left|\mathbf{f}_{k-l+1}^j - \text{med}\ \{\ \mathbf{f}_l^j\}\right|}{6\ \text{med}\ \left\{\ \left|\mathbf{f}_m^j - \text{med}\ \{\ \mathbf{f}_m^j\}\right|\right\}}, \qquad k - N_s \le m \le k\ . \tag{4.19}$$

Posons :

$$\mathbf{s}_k = \mathbf{z}_k - \mathbf{C}_k(\hat{\mathbf{A}}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{B}_{k-1}\hat{\mathbf{u}}_{k-1})\ . \tag{4.20}$$

Le terme $\mathbf{s}_k$ est un terme d'innovation mais il est, dans notre cas, défini par des grandeurs estimées plutôt que par des grandeurs théoriques.

Le raisonnement utilisé pour estimer $\hat{\mathbf{Q}}_k$ est transposable pour déterminer la matrice de covariance $\hat{\mathbf{R}}_k$ du bruit de mesure. Ainsi, pour les N_z dernières observations, à l'itération k on a :

$$\hat{\mathbf{R}}_k = \hat{\mathbf{r}}_k^{(ij)} - \frac{1}{N_z}\sum_{l=1}^{N_z}\left(\hat{\mathbf{C}}_{k-l+1}(\hat{\mathbf{A}}_{k-l}\mathbf{P}_{k-l}\hat{\mathbf{A}}_{k-l}^T + \hat{\mathbf{Q}}_{k-l})\hat{\mathbf{C}}_{k-l+1}^{T)}\right), \tag{4.21}$$

avec

$$\hat{\mathbf{r}}_k^{(ij)} = \frac{N_z\sum_l\left(\mathbf{s}_{k-l+1}^i - \text{med}\ \{\ \mathbf{s}_l^i\}\right)\left(\mathbf{s}_{k-l+1}^j - \text{med}\ \{\ \mathbf{s}_l^j\}\right)\left(1 - N_{ij}^2(l)\right)^4}{\left(\sum_l\left(1 - N_{ij}^2(l)\right)\left(1 - 5N_{ij}^2(l)\right)\right)\left(-1 + \sum_l\left(1 - N_{ij}^2(l)\right)\left(1 - 5N_{ij}^2(l)\right)\right)}, \tag{4.22}$$

où $\sum_l$ désigne la somme pour tous les l pour qui $N_{ij}^2(l) \le 1$. Les termes $\mathbf{s}_k^i$ dans l'expression précédente représentent les i-ème composants de $\mathbf{s}_k$. Les grandeurs $N_{ij}^2(l)$ sont définies par :

$$N_{ij}^2(l) = \frac{\left|\mathbf{s}_{k-l+1}^i - \text{med}\ \{\ \mathbf{s}_l^i\}\right|}{6\ \text{med}\ \left\{\ \left|\mathbf{s}_m^i - \text{med}\ \{\ \mathbf{s}_m^i\}\right|\right\}} \cdot \frac{\left|\mathbf{s}_{k-l+1}^j - \text{med}\ \{\ \mathbf{s}_l^j\}\right|}{6\ \text{med}\ \left\{\ \left|\mathbf{s}_m^j - \text{med}\ \{\ \mathbf{s}_m^j\}\right|\right\}}, \qquad k - N_z \le m \le k\ . \tag{4.23}$$

2.3 Une première application simulée

Nous proposons une première étude comparative pour valider l'algorithme proposé, et pour étudier ses limites avant de l'appliquer à l'asservissement visuel.

Le système qui est considéré est repris de (Kirlin et Moghaddamjoo 1986a). Le modèle est identique à celui défini par les expressions précédentes (4.1) et (4.2) à la différence près qu'il comporte une entrée de forçage u_k. Le système considéré est pris dans le cas scalaire, avec les notations et les valeurs suivantes : $\mathbf{A} = a = 0.1$ et $\mathbf{C} = c = 1$.

Pour les simulations qui suivent nous avons utilisé pour les bruits, qui sont eux aussi scalaires :

- $w_k \sim N(0,\ 0.05)$,
- $v_k = 0.95N(0,\ 1) + 0.05N(0,\ 0.51)$.

La particularité de ce système vient du fait que le bruit de mesure est constitué de deux séquences Gaussiennes, centrées et de variances différentes. Le système reçoit également une entrée de commande, un échelon :

$$u_k = \begin{cases} 0 & k \le 199 \\ 10.0 & k \ge 200 \end{cases}. \tag{4.24}$$

Le système est simulé avec un vecteur d'état initial nul, $\mathbf{x}_k = 0$.

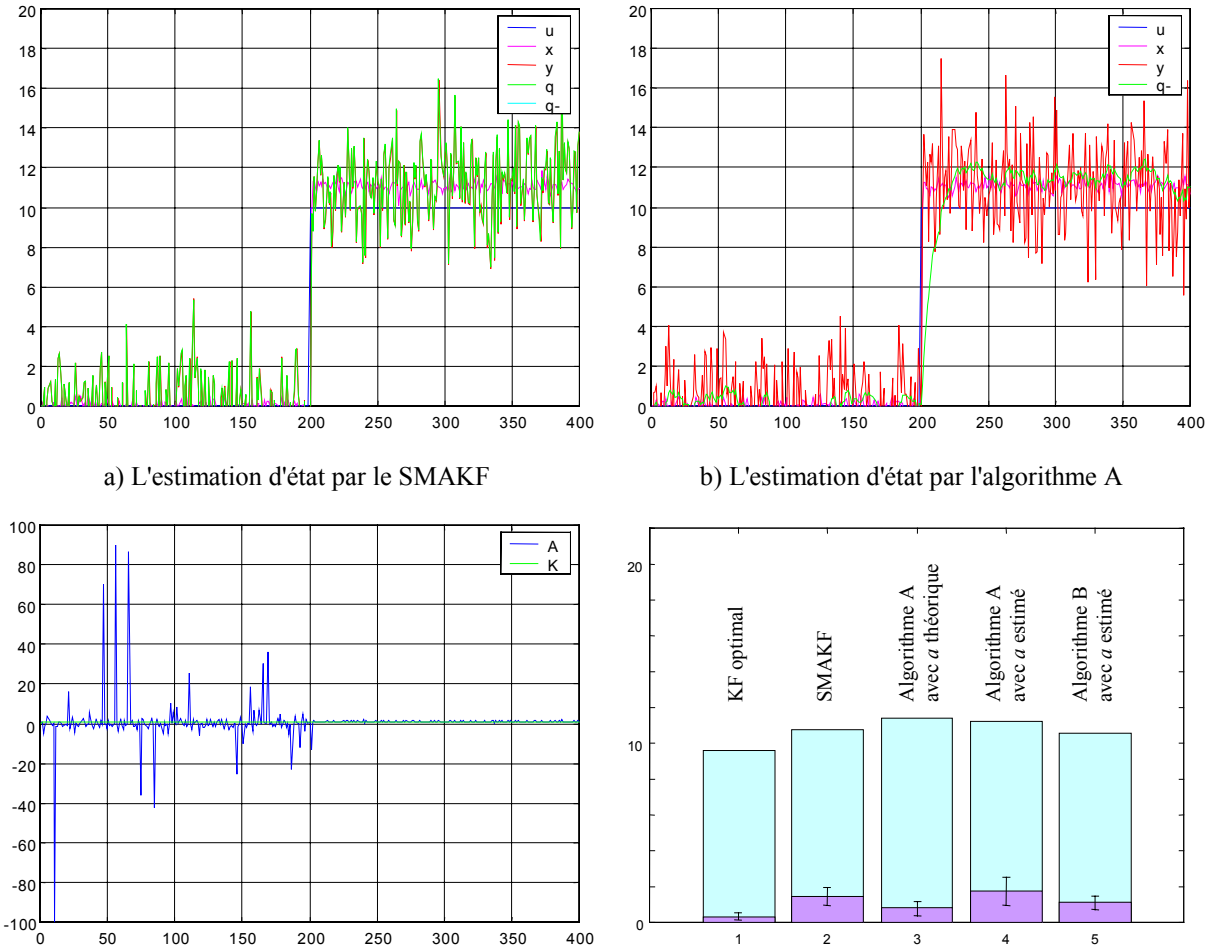
Le vecteur d'état est supposé inaccessible et son estimation est faite par l'algorithme SMAKF. Le SMAKF est sans entrée, c'est l'algorithme d'estimation des covariances qui permet d'estimer l'entrée de forçage. Les performances de cet algorithme sont directement comparées à celles d'un filtre de Kalman optimal, c'est-à-dire qui suppose tout connaître sur le système.

Nous allons également introduire dans les comparaisons le filtre adaptatif temporel présenté dans le Chapitre 2. Plus concrètement, nous allons comparer les performances du SMAKF à l'algorithme A et à l'algorithme B, tous deux exposés dans (Turney et al. 1999). L'algorithme A est l'algorithme qui s'exprime par l'expression (2.74) pour estimer la matrice de transition, et par les

expressions (2.75) et (2.76) pour déterminer les covariances des bruits du processus et de mesure. L'algorithme B reprend la formulation de l'algorithme A et compare la grandeur $\gamma = \left| z_k - a\hat{x}_{k-1} \right| / \hat{\mathbf{R}}_k^{1/2}$ à un seuil Γ puis, en conséquence, effectue une réinitialisation des estimations des covariances $\hat{Q}_k$ et $\hat{R}_0$.

Précisons que les conditions initiales sont les mêmes pour tous les estimateurs : $\hat{x}_0 = 0$, $\hat{u}_0 = 0$, $\hat{Q}_0 = 0.05$, $\hat{R}_0 = 1.0$ et $\mathbf{P}_0 = 10\,\mathbf{I}$.

Pour le SMAKF, nous avons pris $N = N_s = N_z = 15$. Au départ, pendant les deux premières itérations, $\hat{\mathbf{A}}_0 = \mathbf{I}$ est maintenue constante. La matrice de transition ne peut être calculée qu'avec les deux dernières observations ; les deux premières itérations ne servent qu'à faire l'acquisition des observations. Après, la matrice de transition $\hat{\mathbf{A}}_k$ est adaptée selon l'expression précédente (4.13).



a) L'estimation d'état par le SMAKF

b) L'estimation d'état par l'algorithme A

c) L'évolution de la matrice de transition du SMAKF et du gain de Kalman

d) La comparaison des erreurs d'estimation des différentes approches (en bleu les erreurs maximales, en violet les erreurs moyennes)

Figure 4-1 : Comparatif sur une première application simulée.

L'estimation du SMAKF est représentée en haut à gauche ; l'estimation de l'algorithme de A avec estimation en ligne de la matrice de transition est représentée en haut à droite ; l'évolution de la matrice de transition et du gain de Kalman du SMAKF sont montrées en bas à gauche ; les erreurs comparées, moyennes et maximales, ainsi que les écarts types (à l'échelle ½), sont présentés en bas à droite pour les différents estimateurs.

Les performances des techniques évaluées sont basées sur l'erreur quadratique, définie comme étant le carré de la différence entre l'état vrai du système et l'estimée faite par le filtre de Kalman. Les résultats sont résumés par la Figure 4-1, ils représentent les résultats moyens d'une centaine de simulations. Ils permettent de comparer les erreurs d'estimation du SMAKF à celles du filtre optimal, à celles de l'algorithme A avec *a* théorique constant, à celles de l'algorithme A avec *a* estimé en ligne, et à celles de l'algorithme B avec *a* estimé en ligne.

Naturellement, le filtre de Kalman optimal est l'estimateur le plus performant, mais il est également le filtre qui exige le plus de connaissance. Le SMAKF et l'algorithme A avec *a* théorique sont les plus performantes des méthodes réalistes, seulement, à performances quasi-similaires, le SMAKF n'est pas basé sur des connaissances a priori. Les algorithmes A et B, avec *a* estimé en ligne, sont les moins performants des estimateurs. Lorsque la matrice de transition utilisée par le filtre est adaptative et estimée en ligne par la méthode proposée par Turney, Reza et Delva, le filtre est plus lent à converger. Les changements brusques dans le comportement du système sont des pièges systématiques pour les algorithmes A et B, même avec *a* théorique. Le SMAKF est capable de compenser l'entrée de forçage, et constitue la technique la plus efficace des méthodes adaptatives.

D'autres simulations et comparaisons ont été menées pour la validation de cette approche, portant notamment sur l'estimation de la sortie d'un système linéaire du deuxième ordre commandé par un signal en créneaux, ou sur la modélisation et la prédiction de l'oscillateur de van der Pol (Hermann 2000). Ces travaux ont également cherché à augmenter le vecteur d'état pour améliorer les performances de l'estimateur SMAKF.

2.4 Résultats de stabilité

Avant d'aller plus loin, nous allons étudier le comportement de la matrice de transition. Les simulations précédentes ont permis de valider l'approche, à savoir estimer d'une manière sous-optimale le comportement d'un système sans aucune connaissance préalable sur celui-ci.

Un système scalaire comme le système précédent n'étant pas propice à une telle étude, nous allons simplement considérer la prédiction de quatre signaux. Ces signaux sont issus d'un système de vision, un capteur stéréoscopique monté sur une tête robotique. Chaque caméra est dotée de deux degrés de liberté et est asservie par le principe des saccades sur la trajectoire aléatoire d'un point mobile dans un espace 3-D. De cette façon, le système est « connu », mais il n'est pas modélisé.

La mise en œuvre d'un SMAKF vise comme objectif la prédiction de la position future de la cible dans les plans de projections, c'est-à-dire les coordonnées verticales et horizontales dans les images. Ce système est intéressant dans la mesure où les signaux observés ont une signification réelle, et dans la mesure où ils ne sont pas indépendants l'un par rapport à l'autre. Un exemple de tels signaux est représenté sur la Figure 4-2. Cette même figure montre l'évolution des coefficients a_{ij} de la matrice $\hat{\mathbf{A}}_k$ (Figure 4-2c).

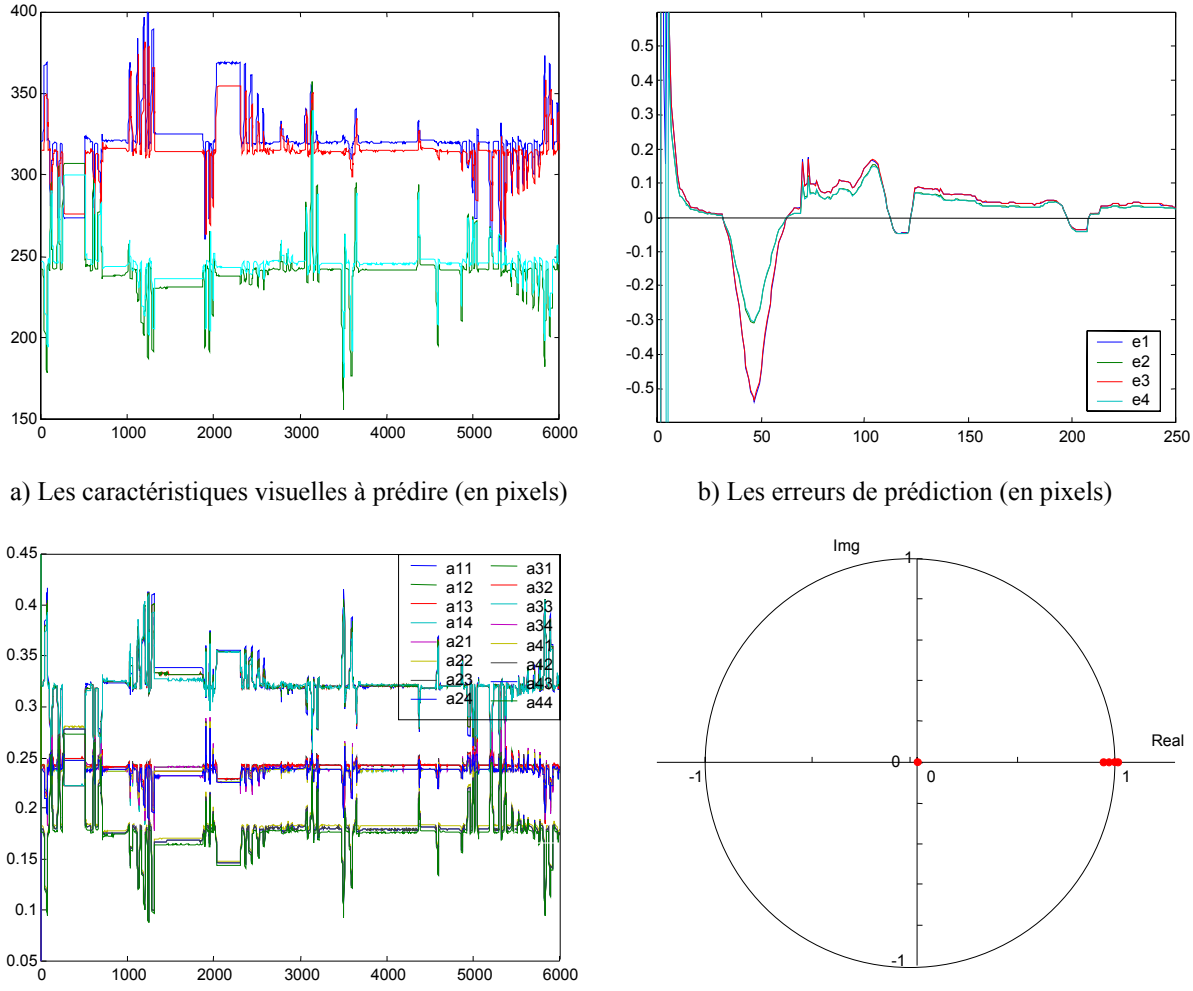
Une première remarque concerne la « corrélation » qui ressort entre l'évolution des signaux à prédire (a) et celle des coefficients a_{ij} de la matrice $\hat{\mathbf{A}}_k$ (c). On constate notamment que lorsque les signaux sont stables, les coefficients de la matrice $\hat{\mathbf{A}}_k$ n'évoluent plus et conservent leur valeur.

Sur la Figure 4-2c, qui montre les coefficients a_{ij}, on voit trois familles de courbes qui varient de la même façon et qui constamment restent proches :

- une première famille d'éléments (celle du haut sur la Figure 4-2c) qui relient les anciennes coordonnées horizontales aux nouvelles coordonnées horizontales, c'est-à-dire les éléments a_{11}, a_{13}, a_{31} et a_{33} ;
- une deuxième famille d'éléments (celle du milieu sur la Figure 4-2c) qui font passer les anciennes coordonnées verticales aux nouvelles coordonnées verticales, en fait les éléments a_{22}, a_{24}, a_{42} et a_{44} ;

- et une dernière famille d'éléments (celle du bas sur la Figure 4-2c) relient les anciennes coordonnées horizontales aux nouvelles coordonnées verticales et relient les anciennes coordonnées verticales aux nouvelles coordonnées horizontales, donc les éléments a_{12}, a_{14}, a_{21}, a_{23}, a_{32}, a_{34}, a_{41} et a_{43}.

Ces constatations démontrent que les coefficients qui ont la même signification évoluent de la même manière, et que cette évolution est représentative des signaux observés.



a) Les caractéristiques visuelles à prédire (en pixels)

b) Les erreurs de prédiction (en pixels)

c) L'évolution des coefficients de la matrice de transition du SMAKF (sans unité)

d) Les valeurs propres de toutes les matrices de transition du SMAKF

Figure 4-2 : Etude du comportement de la matrice de transition adaptative.

Les quatre signaux images issus du capteur stéréoscopique sont représentés en haut à gauche (les unités verticales sont des pixels, le temps est représenté horizontalement). En bas à gauche, on voit que les coefficients de la matrice de transition estimée sont corrélés avec les signaux observés. L'erreur (en pixels) de prédiction faite par le SMAKF sur les signaux observés est montrée en haut à droite pour les 250 premières itérations. Les valeurs propres de cette matrice sont susceptibles d'évoluer : elles sont représentées en bas à droite pour les 100 premières itérations et ne sortent pratiquement jamais du cercle unitaire.

Nous nous sommes également intéressé aux valeurs propres de la matrice $\hat{\mathbf{A}}_k$. Sur la Figure 4-2d, on peut voir que même si les coefficients de la matrice $\hat{\mathbf{A}}_k$ évoluent au cours du temps, les valeurs propres restent quasiment inchangées. Une étude plus approfondie montre que seule la valeur propre qui se trouve le plus vers l'extérieur du cercle unitaire, celui qui donne la référence de stabilité dans l'espace discret, a tendance à se déplacer et peut même se positionner à la limite de la stabilité. Ces cas sont rares et ne durent que le temps d'une itération. Comme ils sont brefs, ils n'altèrent pas la stabilité du filtre et ne sont pas des causes éventuelles de divergence. Ces résultats sont présentés dans (Wira et Urban 2000), où le SMAKF est favorablement comparé à des méthodes classiques.

2.5 Discussion

Incontestablement, le SMAKF est un filtre sous-optimal. Toutefois, par rapport au filtre optimal, l'algorithme SMAKF estime en plus de l'état du système, les paramètres $\hat{\mathbf{A}}_k$, $\hat{\mathbf{R}}_k$, et $\hat{\mathbf{Q}}_k$ à chaque itération.

L'estimation de la matrice de transition par le SMAKF permet de refléter la dynamique du système à un instant donné. La matrice $\hat{\mathbf{A}}_k$ n'est pas représentative d'un système dans son ensemble, mais elle est une bonne représentation de la dynamique d'un système dans un intervalle de temps relativement restreint. Elle est une représentation d'état qui évolue dans le temps et qui est fonction de l'état courant du système. C'est pour cette raison qu'elle doit être mise à jour en permanence.

L'estimation $\hat{\mathbf{A}}_k$ du SMAKF ne dépend pas de la performance du filtre mais uniquement des observations $\mathbf{z}_k$ du système. Les performances du SMAKF dépendent essentiellement des grandeurs choisies pour représenter le système, autrement dit du vecteur d'état. Le système doit également, de façon logique mais non explicite, être un système observable. Le vecteur des observations doit être assez pertinent pour permettre au SMAKF (ou de manière générale au filtre de Kalman classique) d'estimer correctement l'espace d'état.

L'avantage principal de cette méthode, vient du fait qu'il y a très peu de restriction sur le type de signaux à estimer ou à prédire. Le SMAKF est capable de compenser à la fois la non stationnarité des signaux environnementaux, notamment toutes les dérives des propriétés statistiques des bruits, et de compenser les dérives des paramètres déterministes d'un système.

Très peu d'approches prennent en compte le fait que le modèle dynamique du système puisse changer ou évoluer au cours du temps, hormis les approches à modèle multiple qui demandent un critère explicite de changement de modèle. Une des rares approches est celle proposée dans (Turney et al. 1999), où un filtre adaptatif temporel est appliqué à un problème réel.

La méthode d'estimation des covariances des bruits de l'algorithme SMAKF est une technique optimale totalement indépendante de la procédure d'adaptation de la matrice de transition. Elle permet notamment de rendre le filtre apte à travailler dans des environnements non stationnaires. Elle a néanmoins un inconvénient, c'est de n'être réellement performante que lorsque l'évolution des statistiques de l'environnement est fixe et lente.

Pour parer à cette contrainte, nous proposons d'associer l'estimation de la matrice de transition à un schéma général de l'approche à modèle multiple pour l'estimation des paramètres des bruits. Nous privilégions pour cela les structures des Mélanges d'Experts pour les capacités d'apprentissage qu'elles offrent. Ceci nous conduit aux mélanges de filtres de Kalman et plus précisément aux mélanges de filtres de Kalman adaptatifs qui sont développés dans la section suivante.

3 Le mélange de filtres de Kalman adaptatifs

3.1 Autres approches

Les bancs de filtres datent des années soixante et si les concepts étaient déjà jetés, leur application a immédiatement été limitée par des implémentations matérielles beaucoup trop

contraignantes. La technologie numérique, grâce aux microprocesseurs ultrarapides et aux DSP ("Digital Signal Processor") qui traitent un nombre impressionnant d'opérations par secondes, leur donne aujourd'hui un nouveau souffle. Les mélanges de KF ne sont donc pas une nouveauté en soi, un certain nombre de travaux le montre.

Avant de présenter les mélanges de filtres, rappelons que Mehra range le banc de filtres de Magill dans les méthodes Bayésiennes du filtrage adaptatif. Les premiers développements issus du banc de Magill tournent autour du concept de l'IMM ("Interacting Multiple Model"). Ce concept est une version améliorée de l'estimateur de Magill, mais aussi des méthodes générales pseudo-Bayésiennes pour l'estimation à modèle multiple. En fait, l'algorithme IMM exige qu'un nombre fini de réalisations du vecteur des paramètres inconnus soit déduit à partir d'une matrice de probabilité de transition connue (Chaer et al. 1997). Une étude des différentes approches et contributions sont passées en revue dans (Mazor et al. 1998).

Forts de nombreux avantages, les ME ont rapidement été inclus dans les approches à modèle multiple pour finalement constituer une alternative efficace. Dans (Chaer et Bishop 1995) par exemple, les auteurs partent de l'approche des bancs de filtres précédemment mis au point par Magill (Magill 1965). Différentes réalisations de filtres de Kalman forment les experts d'un mélange. Ils se distinguent par des valeurs différentes d'un certain nombre de paramètres, rangés dans un vecteur $\boldsymbol{\theta}$. Un algorithme d'estimation de densité de probabilité pondère leurs sorties pour déterminer la sortie globale du mélange. Les auteurs ont constaté que cet algorithme ne combine correctement les sorties de plusieurs filtres de Kalman que si le filtre optimal (ou un filtre proche de l'optimalité) se trouve dans le banc. Comme les filtres ne sont pas adaptatifs, si le filtre optimal ne se trouve pas parmi les filtres qui composent le banc, alors l'estimateur de la densité de probabilité (une technique Bayésienne) ne peut pas converger vers une solution performante. En fait, quelle que soit la combinaison de filtres utilisés, l'estimateur de densité de probabilité converge et privilégie le filtre qui donne les meilleurs estimations. Si ce filtre n'est pas performant, la structure entière ne peut être efficace.

Dans les mêmes travaux, un algorithme génétique est proposé pour la résolution de ce problème. En cas de non convergence vers une solution satisfaisante, l'algorithme génétique génère des filtres nouveaux avec des paramètres différents de ceux utilisés. Ainsi, après avoir renouvelé une partie de la population des filtres, l'algorithme génétique doit permettre d'approcher correctement les paramètres recherchés. La solution est validée avec une application d'estimation des paramètres des bruits dans le cadre du filtrage adaptatif de Kalman et avec une application de détermination de l'orbite de satellites interplanétaires.

Contrairement au banc de filtres de Magill, les mélanges de filtres ne sont pas obligés de supposer que le filtre optimal se trouve parmi les réalisations (les experts) pour converger. Le fait que les filtres soient adaptatifs (dans le sens général des paramètres) permet de construire des mélanges de tailles moindres mais tout aussi efficaces. De plus, les mélanges de filtres ne nécessitent pas les contraintes qu'imposent les IMM.

Dans la suite de leurs travaux de recherche, les mêmes auteurs modifient leur première approche en proposant une architecture constituée (Chaer et al. 1997) :

- d'un banc de filtres pour constituer les différentes réalisations des filtres avec des paramètres distincts,
- d'un réseau de neurones afin d'apprendre les performances des filtres puis de pondérer leur sortie pour le calcul de la sortie globale (le filtre le plus performant est celui qui se verra attribuer la pondération de plus forte valeur),
- et d'un algorithme de recherche des paramètres pour mettre à jour les filtres du banc à partir des connaissances apprises par le réseau de neurones.

Le banc de filtres et le réseau de neurones sont placés dans la boucle directe, ensemble ils réalisent une méthode d'estimation d'hypothèses multiples ("multiple hypothesis estimation

algorithm"). L'algorithme de recherche se trouve dans la boucle de retour. Les auteurs étudient deux algorithmes de recherche, un algorithme RQP ("Recursive Quadratic Programming"), et un algorithme génétique. Le premier cherche les extremums d'une fonction de vraisemblance modifiée et est compatible avec des applications temps réel, puisqu'uniquement le filtre le plus performant est adapté, et périodiquement seulement. L'algorithme génétique, au contraire, accroît considérablement la complexité du mélange de filtres.

Le principe a été étendu à une approche hiérarchique (Chaer et al. 1998) permettant de tester les hypothèses sur plusieurs niveaux. C'est alors l'algorithme EM qui est chargé de la supervision hiérarchiques des experts.

Une autre approche originale, le MKF ("Mixture Kalman filters"), est présentée dans (Chen et Liu 2000). Il s'agit d'un filtre de Monte Carlo (une technique particulaire) modifié de sorte qu'il porte toute son attention sur un espace restreint de variables indiquées. La conséquence directe de cette modification est une baisse significative de la variance de l'état par rapport à un filtre particulaire classique. Cette approche est validée par des tâches d'estimation et de prédiction lorsque les bruits pris en compte ne sont pas Gaussiens.

3.2 Avec une estimation optimale des paramètres des bruits

L'idée des filtres de Kalman organisés dans une structure modulaire du ME est ici reprise. Dans (Chaer et al. 1997), il est préconisé d'utiliser des filtres adaptatifs pour d'une part, avoir des résultats meilleurs, et d'autre part aboutir à des structures de tailles raisonnables.

Nous allons considérer un ME dont les experts sont des SMAKF et dont le réseau superviseur est un réseau mono-couche avec autant de sortie que d'experts. Comme les SMAKF réalisent une estimation acceptable de la dynamique d'un système inconnu, et que cette estimation est unique à un instant donné, la même loi peut être utilisée pour estimer la matrice $\hat{\mathbf{A}}_k$ de chaque expert, soit (4.10), (4.11), ou (4.12) selon la taille de la matrice de mesure. Dans la pratique, une unique procédure d'estimation de la matrice de transition $\hat{\mathbf{A}}_k$ permet de limiter le coût algorithmique.

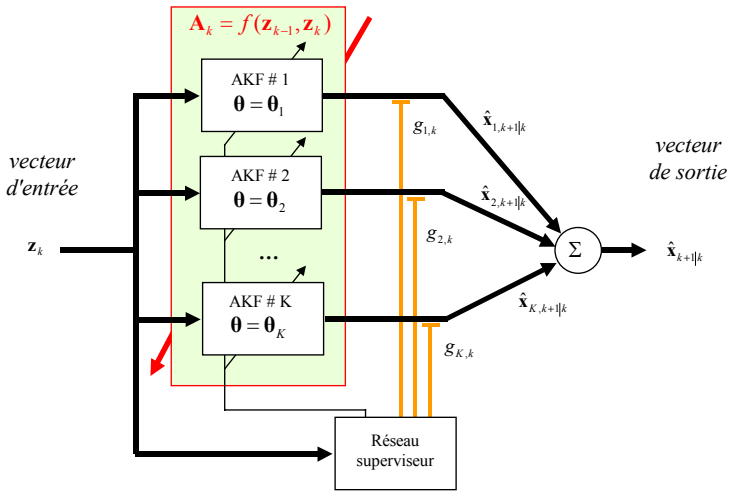


Figure 4-3 : Principe du mélange de filtres adaptatifs.

Le mélange de SMAKF reprend le principe des ME. Chaque expert est un SMAKF avec des paramètres différents. Le réseau superviseur se doit de choisir le filtre le plus performant et, en fonction des connaissances apprises, il influence l'adaptation des paramètres des SMAKF pour les rendre encore plus performants.

A l'opposé, l'estimation les matrices de covariances $\mathbf{Q}_{i,k}$ et $\mathbf{R}_{i,k}$ est propre à chaque expert. Si $\boldsymbol{\theta}_i$ est le vecteur des paramètres qui changent d'un expert à l'autre, les différents SMAKF ne sont différenciés que par des initialisations différentes. A partir d'un vecteur paramètre initial $\boldsymbol{\theta}_{i,0} = [\mathbf{R}_{i,0} \quad \mathbf{Q}_{i,0} \quad \mathbf{P}_{i,1|0} \quad \hat{\mathbf{x}}_{i,1|0}]^T$, seuls les éléments $\mathbf{Q}_{i,k}$ et $\mathbf{R}_{i,k}$ du *i*-ème expert sont corrigés en ligne, avec les règles (4.19) et (4.21), et aboutissent à des estimations significatives pour une interprétation statistique. Le principe énoncé ci-dessus est présenté sur la Figure 4-3.

L'intérêt des matrices de covariance évolutives est de pouvoir traiter la non stationnarité de l'environnement. L'idée développée ici part du principe que, si les propriétés statistiques des signaux environnementaux évoluent de manières constantes, alors un simple SMAKF est suffisant. Par contre, étant donné la difficulté d'estimer les variations brusques des statistiques des signaux de l'environnement, plutôt que de se contenter d'une estimation « retardée », plusieurs estimations sont proposées à travers les différentes réalisations des SMAKF d'un mélange ; et il incombe au réseau superviseur de choisir l'estimation la plus appropriée aux vues des dernières observations. Les estimations des matrices de covariance pourront ainsi être très différentes les unes des autres sans avoir de mauvaises conséquences sur l'estimation globale, puisque le réseau superviseur effectue un choix, et peut le changer de manière très rapide.

Un avantage essentiel de cette structure provient des capacités d'apprentissage du réseau superviseur, comme le montrerons les applications dans le chapitre suivant.

Ce premier mélange de SMAKF comporte cependant un inconvénient. Comme l'algorithme d'estimation des matrices de covariance est issu d'une méthode optimale, les estimations $\mathbf{Q}_{i,k}$ et $\mathbf{R}_{i,k}$ des différents filtres tendent à converger vers des valeurs similaires et uniques. Autrement dit, malgré des valeurs initiales différentes, après un certain nombre d'itérations, les différents filtres qui composent le mélange deviennent similaires. Or, nous avons vu qu'il se créé une indépendance entre les différents modules experts s'ils ne sont pas assez différenciés, et que cette corrélation (positive) dégrade les capacités de l'ensemble du mélange. En fait, pour qu'un ME soit performant il faut une corrélation négative entre les experts, et ceci se traduit par des experts qui se distinguent dans leur comportement.

Dans cette perspective, nous proposons donc un mélange de filtres de Kalman basé sur des règles empiriques d'ajustement des estimations des covariances.

3.3 Avec une estimation empirique des paramètres des bruits

L'estimation empirique des paramètres des bruits que nous allons développer a pour vocation de favoriser le développement d'une corrélation négative entre les experts du mélange. Nous reprenons l'idée de Kelly (Kelly 1994) qui consiste à faire évoluer les covariances de manière décroissante dans le temps pour, à la fois augmenter les performances de l'estimation, et limiter les risques d'apparition des cas de divergence.

La procédure d'apprentissage du ME est la même et le vecteur paramètre est inchangé par rapport à l'approche précédente :

$$\boldsymbol{\theta}_i = \left[\hat{\mathbf{R}}_{i,k} \quad \hat{\mathbf{Q}}_{i,k} \quad \mathbf{P}_{1|0} \quad \hat{\mathbf{x}}_{1|0} \right]^T . \tag{4.25}$$

Si $\mathbf{Q}_{i,k}$ et $\mathbf{R}_{i,k}$ sont les matrices de covariance des bruits du *i*-ème expert, celles-ci sont considérées comme des matrices diagonales et sont initialisées différemment pour chaque expert. Elles évoluent par la suite selon les lois suivantes :

$$\mathbf{Q}_{i,k+1} = \mathbf{Q}_{i,k}(1 - \beta \eta_k h_{i,k}) , \tag{4.26}$$

$$\mathbf{R}_{i,k+1} = \mathbf{R}_{i,k}(1 - \beta' \eta_k h_{i,k}) , \tag{4.27}$$

où η_k est le coefficient d'apprentissage du ME. Ce coefficient peut évoluer dans le temps et avoir, par exemple, l'allure du coefficient d'apprentissage utilisé par l'algorithme SOM. Le terme $h_{i,k}$

est la probabilité a posteriori précédemment introduite dans le principe du ME (cf. équation (3.7)) et, $\beta > 0$ et $\beta' > 0$ sont des coefficients scalaires d'adaptation. On peut se rendre compte qu'avec les lois (4.26) et (4.27), les éléments diagonaux de $\mathbf{Q}_{i,k}$ et $\mathbf{R}_{i,k}$ sont adaptés de façon identique. Lorsque des connaissances préalables sont disponibles sur les bruits, les expressions (4.26) et (4.27) peuvent s'écrire sous les formes matricielles

$$\mathbf{Q}_{i,k+1} = \mathbf{Q}_{i,k}(\mathbf{I} - \boldsymbol{\beta}\eta_k h_{i,k}), \tag{4.28}$$

$$\mathbf{R}_{i,k+1} = \mathbf{R}_{i,k}(\mathbf{I} - \boldsymbol{\beta}'\eta_k h_{i,k}), \tag{4.29}$$

en réglant explicitement les coefficients des matrices $\boldsymbol{\beta} \in \mathbb{R}^{s\times s}$ et $\boldsymbol{\beta}' \in \mathbb{R}^{p\times p}$.

Afin d'augmenter les capacités d'adaptation de chaque expert, un coefficient d'apprentissage adaptatif (qui est un coefficient d'adaptation pour le filtre) est introduit. Son rôle consiste à forcer un expert à s'adapter davantage lorsque l'erreur d'estimation est grande. Si l'erreur d'estimation est faible, l'adaptation de l'expert concerné ne doit pas être grande.

Chaque expert se voit ainsi doté d'un coefficient d'apprentissage propre, qui tient explicitement compte de l'erreur de l'expert par rapport à la sortie désirée :

$$\eta_{i,k} = \eta_{ini}\left(1 - \exp(-\alpha\varepsilon_{i,k})\right)\left(\frac{\eta_{fin}}{\eta_{ini}}\right)^{\frac{k}{k_{fin}}}, \tag{4.30}$$

où

- α est un paramètre scalaire positif, non nul et constant,
- η_i est la valeur initiale du coefficient d'apprentissage,
- η_f est la valeur finale vers laquelle le coefficient doit tendre,
- $\varepsilon_{i,k}$ est l'erreur faite par le *i*-ème expert ; elle est une grandeur scalaire qui est définie selon l'expression suivante :

$$\varepsilon_{i,k} = (\mathbf{d}_k - \mathbf{y}_{i,k})^T(\mathbf{d}_k - \mathbf{y}_{i,k}) \tag{4.31}$$

avec $\mathbf{d}_k$ la sortie désirée et $\mathbf{y}_{i,k}$ la sortie estimée par le *i*-ème expert.

Lorsque le mélange de SMAKF est utilisé comme un prédicteur (c'est le cas lorsqu'il est appliqué à l'asservissement visuel), l'erreur utilisée pour adapter les coefficients d'apprentissage $\eta_{i,k}$ est alors plutôt de la forme :

$$\varepsilon_{i,k|k-1} = (\mathbf{d}_k - \mathbf{y}_{i,k-1})^T(\mathbf{d}_k - \mathbf{y}_{i,k-1}). \tag{4.32}$$

C'est une erreur de prédiction, définie entre la sortie désirée et l'avant dernière estimation $\mathbf{y}_{i,k-1}$ qui est la valeur de $\mathbf{d}_k$ prédite par le *i*-ème expert.

3.4 Remarques

Réaliser une tâche d'apprentissage, d'estimation ou de prédiction dans un environnement non stationnaire n'est pas une tâche aisée, surtout lorsque des changements brusques dans les statistiques des signaux de l'environnement apparaissent.

Les outils statistiques pour estimer les propriétés des signaux sont des techniques qui souffrent d'un handicap majeur : ils requièrent un grand nombre d'observations avant de donner des résultats significatifs. C'est pour cette raison que le banc de Magill est long à passer d'un filtre à un autre dans le processus de sélection d'un modèle parmi tous ceux qui le composent. Le ME, parce qu'il utilise une approche neuronale plutôt qu'un algorithme statistique, ne possède pas cet inconvénient et se montre parfaitement rapide et efficace pour changer de modèle. Le ME est plus qu'un banc de filtres, il a de la mémoire. L'approche des ME, en reprenant l'architecture des bancs de filtres, va au delà du principe des bancs en effectuant un apprentissage, celui qui permet d'associer l'expert le plus performant aux entrées.

Cet apprentissage, réalisé par le réseau superviseur, permet à ce dernier de prédire quel est l'expert le plus judicieux pour un vecteur d'entrée donné. Les connaissances acquises au cours de cet

apprentissage permettent une bonne combinaison des sorties des experts, et permettent surtout de changer rapidement d'expert dès la prise en compte des dernières observations.

Le fait que les experts soient adaptatifs est d'une grande importance dans l'approche à modèle multiple proposée puisque cela permet une spécialisation de chaque expert. Les experts peuvent alors se spécialiser sur une portion de l'espace d'entrée. D'un point de vue pratique, cela permet également de limiter le nombre d'éléments dans la structure et donc de réduire le coût algorithmique.

On aboutit finalement à un banc de filtres adaptatifs capable :

- de sélectionner le filtre le plus performant grâce à un réseau de neurones,
- de répondre à tout changement ou toute modification brusque de l'environnement en modifiant sa sélection grâce aux données les plus récentes,
- d'adapter les filtres individuellement pour mieux les faire correspondre aux grandeurs d'entrée,
- d'éviter les divergences et de limiter les coûts algorithmiques.

Précisons encore une fois que le mélange de SMAKF, pas plus que le SMAKF seul, ne peut apprendre la matrice de transition $\mathbf{A}_k$. Ces modèles permettent d'adapter cette matrice et de faire en sorte qu'elle reflète la dynamique du système étudié de façon transitoire.

4 Apprentissage de la matrice de transition

4.1 Principe

Il peut être intéressant d'établir un modèle de connaissance du système dont on estime l'état, en particulier avec le filtre de Kalman. Pour ce faire, nous proposons d'apprendre les matrices de passage entre les différents espaces de la représentation d'état, notamment la matrice $\mathbf{A}_k$ de transition, et la matrice d'observation $\mathbf{C}_k$. L'estimation de la matrice $\mathbf{A}_k$ peut se faire sur le principe d'une carte SOM à sortie supervisée (un NeuroModule), et celle de la matrice $\mathbf{C}_k$ sur le principe d'une carte ESOM.

Cette nouvelle approche est appelée "Self-Organizing Kalman Filter", ou SOKF, et son principe est expliqué par la Figure 4-4. On peut la comparer à celle qui présente le principe du filtrage de Kalman conventionnel (Figure 2-2).

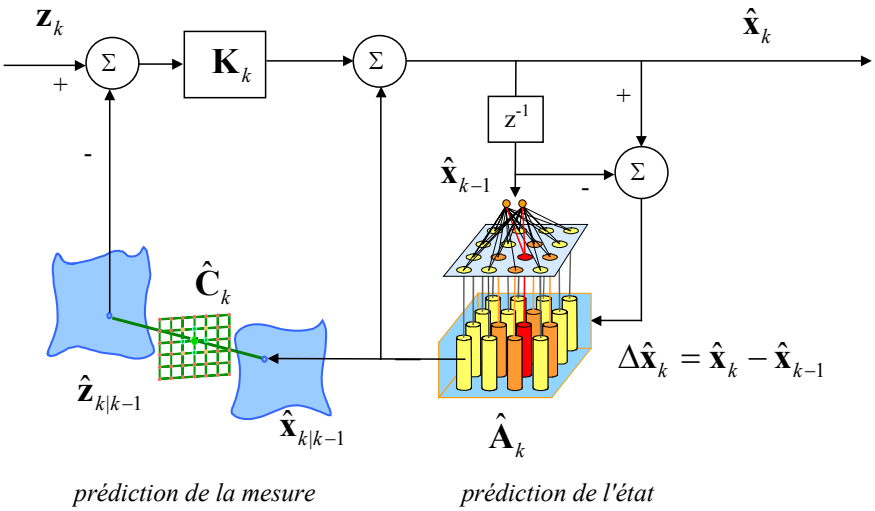


Figure 4-4 : Principe du AKF avec apprentissage des matrices de transition et d'observation.

L'apprentissage de la matrice de transition est réalisé par une carte SOM à sortie supervisée, et l'apprentissage de la matrice d'observation via une carte ESOM.

On peut dire que le SOKF se révèle utile lorsqu'aucune connaissance a priori n'est disponible sur le système, ou que le système est non linéaire et que sa linéarisation induit une approximation trop grossière par rapport à la précision requise. Le SSOM (ou le NeuroModule) réalise également une approximation du premier ordre, mais le fait de poursuivre l'apprentissage au cours de l'utilisation du réseau le rend flexible dans le sens où cette linéarisation n'est plus rigide mais est adaptée à la zone courante de fonctionnement.

L'inconvénient de l'approche est sa lenteur à converger. Ceci peut s'expliquer par le fait qu'il est demandé à ce filtre d'estimer l'état d'un système (c'est son rôle premier), d'adapter le gain de Kalman, et d'apprendre les matrices de passage. Ces différentes tâches sont imbriquées et sont dépendantes les unes des autres. Elles doivent palier le manque de connaissances a priori. Du point de vue de la stabilité, notons que le SOKF est la moins robuste des approches comparées dans les simulations qui sont présentées ci-dessous, dans le sens où les divergences apparaissent plus fréquemment que pour les autres AKF.

Comparé au SMAKF, le SOKF n'aboutit pas au même modèle de connaissance. Le SMAKF est représentatif du système dans un intervalle de temps bien précis, alors que le SOKF représente le système sur l'ensemble de l'évolution de son vecteur d'état. La matrice de transition $\mathbf{A}_k$ et la matrice d'observation $\mathbf{C}_k$ sont apprises, contrairement au principe du SMAKF. Cette dernière approche est illustrée à travers une simulation : l'estimation et la modélisation de la série temporelle de Hénon.

4.2 Simulations

L'apprentissage des matrices de passage entre les différents espaces d'une représentation d'état se justifie essentiellement sur des systèmes qui restent sans modèle ou encore sur les systèmes non linéaires. Nous allons, pour cette série de simulations, nous intéresser à un problème connu dans le domaine de l'estimation et de la prédiction : la série temporelle de Hénon.

La série temporelle de Hénon est un système non linéaire défini par l'équation suivante :

$$x_k = 1 - a\, x_{k-1}^2 + b\, x_{k-2}\,, \tag{4.33}$$

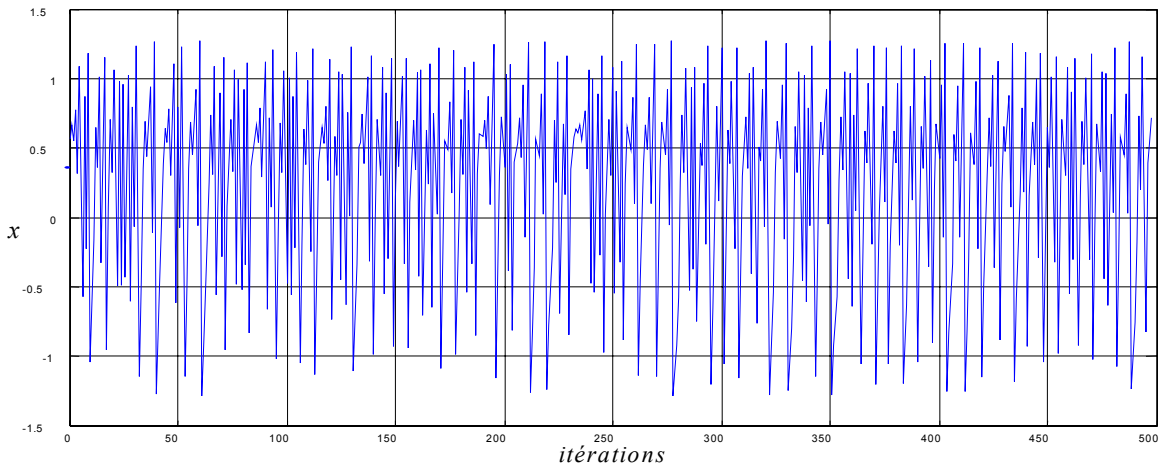
avec les paramètres $a = 1.4$ $b = 0.3$. Les conditions initiales considérées sont : $x_{-1} = 0.3638$ et $x_{-2} = 0.5630$. Précisons que la moyenne de cette série vaut environ $E[x_k] = 0.24$ et sa variance environ $E[x_k^2] = 0.51$ au delà de 100 données ($k > 100$). La série de Hénon est typique de deux caractéristiques propres aux systèmes chaotiques : elle est dépendante des conditions initiales et elle est ergodique. La série temporelle est représentée sur la Figure 4-5.

La série temporelle de Hénon permet de créer un attracteur, uniquement valable autour des valeurs des paramètres $a = 1.4$ et $b = 0.3$. Il se défini par :

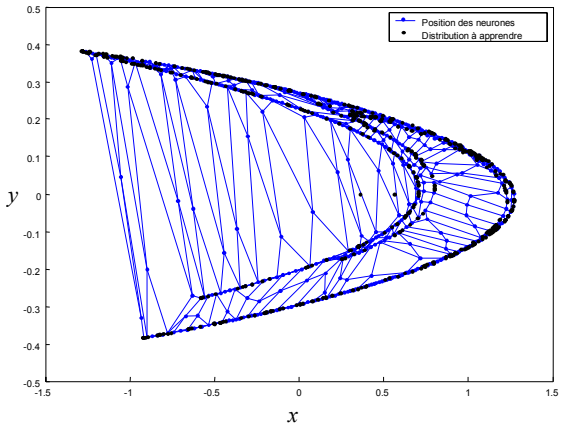
$$\begin{cases} x_k = 1 - 1.4 x_{k-1}^2 + y_{k-1} \\ y_k = 0.3 x_{k-1} \end{cases}. \tag{4.34}$$

En clair, les simulations consistent à modéliser la série temporelle et à prédire $\hat{x}_k = f(x_{k-1}, x_{k-2})$. Elles portent sur les 1500 premiers cycles de la série temporelle pour l'apprentissage et les 500 itérations suivantes pour l'évaluation des performances.

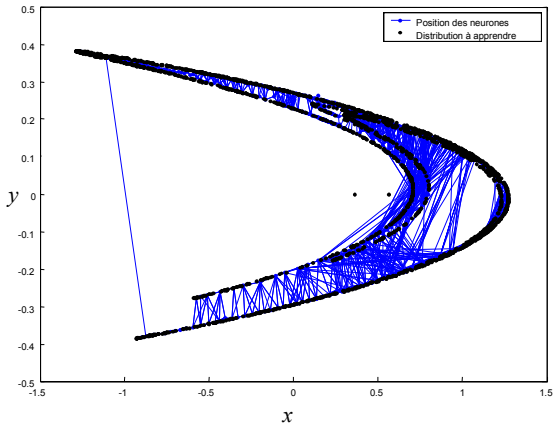
Le vecteur d'état est constitué des grandeurs qui définissent l'attracteur, c'est-à-dire x_k et y_k. L'apprentissage de la matrice de transition $\mathbf{A}_k$ est réalisé par un NeuroModule. La carte auto-organisatrice est une carte bidimensionnelle composée de 25×25 unités neuronales, chacune associée à un réseau Adaline de la taille de l'espace d'état (2×2). Au cours de cette série de simulations, la matrice d'observation est la matrice identité, elle ne sera donc pas estimée. La discrétisation de l'espace d'état par la carte de Kohonen est présentée sur la Figure 4-5. Cette figure montre qu'un certain nombre de neurones se trouvent placés dans des zones dénuées de tout exemple d'apprentissage. Ces neurones, de par leur position isolée par rapport à la distribution se retrouvent inefficaces et inutiles dans le processus d'apprentissage et de quantification.



a) La série temporelle de Hénon



b) La discrétisation de l'attracteur de Hénon par une carte de Kohonen



c) La discrétisation de l'attracteur de Hénon par une carte "neural gas"

Figure 4-5 : Apprentissage et modélisation de la série temporelle de Hénon.

En haut, sur toute la largeur, l'évolution dans le temps de la série temporelle non linéaire de Hénon ; en bas à gauche l'attracteur de Hénon et sa discrétisation par un réseau SOM bidimensionnel (une carte de $25 \times 25 = 625$ *neurones) ; et en bas à droite l'attracteur de Hénon et sa discrétisation par un réseau "neural gas" (625 neurones). De par son voisinage particulier, le réseau "neural gas" réalise une discrétisation de qualité meilleure que celle de la carte de Kohonen, puisque les positions des neurones sont optimisées.*

Les simulations ont alors été répétées avec un apprentissage de l'espace d'état par une carte auto-organisatrice disposant d'une topologie et d'un voisinage différent, le réseau "neural gas". La nature spécifique de la fonction de voisinage intervenant dans les cartes SOM et "neural gas" leur confère des propriétés distinctes qui font aboutir l'apprentissage à une quantification différente de l'espace d'entrée. Les quantifications résultantes sont mises en évidence sur la Figure 4-5 (b et c). La répartition des neurones de la carte SOM préserve les relations de voisinage imposées par la grille topologique alors que le "neural gas" n'est pas basé sur un maillage ou sur une grille. Le réseau "neural gas" optimise le positionnement des neurones lorsque la distribution des exemples est non convexe. A chaque neurone de cette carte est également associé, comme pour le NeuroModule, un réseau Adaline apprenant la transformation entre l'espace d'entrée et de sortie.

Prédicteur	Erreur de prédiction		
	Moyenne	Écart Type	Max.
SOKF avec apprentissage par une carte SOM	0.0097	1.3012	0.0403
SOKF avec apprentissage par un réseau "neural gas"	0.0081	1.0289	0.0412
EKF	0.0141	1.8510	0.0397
SMAKF	0.0074	1.1428	0.0364

Tableau 4.1 : Comparaison de l'erreur absolue de prédiction sur la série temporelle de Hénon.

Les résultats présentés dans ce tableau permettent de comparer le filtre de Kalman avec apprentissage de la matrice de transition par une carte de Kohonen (25×25 neurones) et par un réseau "neural gas" (625 neurones), le filtre de Kalman étendu (EKF) et le filtre de Kalman adaptatif au sens du modèle d'état (SMAKF).

Les résultats qui découlent de ces apprentissages sont confrontés dans le Tableau 4.1. Ils découlent d'une vingtaine de simulations répétées de manière identique. Sont également présentés dans ce même tableau, les erreurs de prédiction d'un filtre de Kalman étendu et celles d'un filtre de Kalman adaptatif au sens du modèle d'état (SMAKF). Ces résultats montrent qu'avec une quantification mieux adaptée avec le "neural gas" qu'avec le réseau de Kohonen, le SOKF est capable de mieux apprendre la distribution des vecteurs dans l'espace d'état et donc de mieux prédire la nouvelle valeur de la série.

L'algorithme EKF est sous-optimal et récapitule ses connaissances passées par une moyenne et une covariance des variables impliquées. L'EKF n'est pas capable de réaliser une prédiction précise car la linéarisation du système n'est pas une approximation correcte sur l'ensemble de l'espace de fonctionnement. Ces imprécisions sont d'autant plus amplifiées qu'elles sont propagées sur plusieurs itérations. La meilleure prédiction est faite par le SMAKF avec des estimations optimales des modèles de bruit.

D'autres algorithmes de prédiction ont été appliqués à cette série, comme ceux dans (Castillo 1998; Castillo et al. 2000). Il est difficile d'effectuer une comparaison directe de nos résultats avec ces travaux. En effet, Castillo utilise une décomposition en fonctions neuronales ("fonction networks") et ses résultats sont présentés dans ce contexte particulier. Son approche est une façon d'introduire plus de connaissances dans les architectures neuromimétiques, puisqu'il effectue des décompositions de fonctions.

4.3 Commentaires sur l'approche

L'approche proposée permet de se passer de tout modèle dans l'utilisation du principe du filtre de Kalman. Les relations entre les espaces impliqués par la représentation d'état sont apprises par des réseaux de neurones artificiels. La loi d'évolution dynamique est apprise par une carte de Kohonen à sortie supervisée (SSOM) et la loi associative de mesure est approchée par une carte de Kohonen étendue (ESOM). L'apprentissage des matrices $\mathbf{A}_k$ et $\mathbf{C}_k$ requière toutefois un coût algorithmique supérieur à la simple adaptation du SMAKF. Le grand avantage que nous voyons à l'utilisation des cartes auto-organisatrices de Kohonen pour les estimations des matrice de transition et d'observation, est leur insensibilité au bruit.

Dans les simulations précédentes, nous avons utilisé le principe de Kohonen et le réseau "neural gas" pour effectuer le pavage de l'espace d'état. Le choix du réseau dépendant de la distribution des vecteurs d'états. On préférera le principe de Kohonen pour un espace d'état « plein » et le réseau "neural gas" dans le cas contraire. La différence entre ces deux réseaux est la topologie qui caractérise l'espace des neurones, chacune est mieux adaptée selon un de ces deux cas.

Les simulations précédentes, mais aussi d'autres qui ne sont pas présentées ici, ont montrées que cette approche faisait aussi bien, voire mieux que l'algorithme EKF. Sur la robustesse et la stabilité, nous pouvons dire que cette approche est lente à converger, c'est-à-dire à donner des

estimations avec une précision suffisante. Il faut un certain temps, un certain nombre d'exemples plus précisément, pour établir un modèle de connaissance sur le système, puisqu'aucun a priori n'est employé.

5 Conclusion

Nous avons proposé et développé dans ce chapitre un filtre de Kalman adaptatif au sens de la représentation d'état. Ce filtre est complètement générique et utilise les dernières observations pour mettre à jour sa matrice de transition. Celle-ci est alors une représentation précise du système à un instant donné. Ce filtre, aux performances supérieures à celles d'un filtre de Kalman étendu, possède l'avantage de ne pas nécessiter de connaissance a priori. Les modèles de bruits sont également adaptés en ligne, soit par une méthode optimale, soit en utilisant des lois plus empiriques.

Le principe est ensuite généralisé à des problèmes plus complexes. En effet, plusieurs réalisations de SMAKF sont organisées au sein d'une approche ME. Ce mélange de SMAKF est une expression de la modularité. Il est une approche à modèle multiple, puisque plusieurs filtres estiment en parallèle les mêmes grandeurs. Un module supplémentaire, un réseau de neurones artificiel estime la probabilité qu'a chaque filtre de délivrer l'estimation optimale. Ces probabilités sont des probabilités conditionnelles, elles sont dépendantes du vecteur d'entrée. Le réseau supervise le rôle de chaque filtre. Les probabilités estimées servent à pondérer le rôle de chaque filtre pour une entrée donnée. Elles servent à gérer la participation des filtres pour le calcul de l'estimation globale. Cette dernière est une somme des estimations de chaque filtre, pondérées par la probabilité qui leur est associée. Les probabilités conditionnelles estimées permettent également de déterminer le degré d'adaptation de chaque filtre pour une entrée donnée. Grâce à ses propriétés d'adaptation et d'apprentissage, cette approche à modèle multiple est capable de compenser la non stationnarité d'un système.

Un filtre de Kalman utilisant des matrices de transition et d'observation apprises est également proposé (le SOKF). Il s'agit d'une approche linéaire locale, basée sur la discrétisation de l'espace d'état par des réseaux de neurones auto-organisateurs. Cette dernière approche peut surtout s'employer pour estimer les états internes non accessibles des systèmes non linéaires et sans modèles connus. Elle ne sera pas utilisée dans la suite de cette thèse.

Nous proposons d'estimer les mouvements d'un objet mobile par le SMAK, et plus généralement par un mélange de SMAKF. L'évaluation du mélange de SMAKF, dans le cadre de l'asservissement visuel, est réalisée dans le chapitre qui suit. Ce chapitre présentera une boucle d'asservissement visuel adaptative et prédictive.

Chapitre 5

Application robotique : le suivi de cible

1 Introduction

Le problème du contrôle robotique par la vision artificielle réside dans l'association intelligente d'un système de perception et d'un système d'action en vue de l'exécution de tâches. Le système sensoriel de vision artificielle fournit au système moteur, le robot, les informations utiles à la réalisation des tâches, mettant en évidence une articulation intelligente de la perception sur l'action.

La modélisation usuelle du système robot-vision est complexe et contraignante. De ce fait, nous la substituons par une approche neuronale, c'est-à-dire par un apprentissage en situation réelle qui ne nécessite aucune connaissance analytique a priori.

Nous nous sommes intéressé aux tâches de suivi de cible. Le mélange de SMAKF pour l'estimation du mouvement 2-D a été intégré dans l'approche neuronale de l'asservissement visuel. Parce qu'il est une approche à modèle multiple, le mélange de SMAKF peut sélectionner le filtre le plus adéquat par rapport au type de mouvements de la cible et à partir des mesures dans les images. Cette sélection est d'autant plus robuste qu'elle est basée sur un processus d'apprentissage capable d'estimer les probabilités associées à chaque filtre qui caractérisent leur chance de fournir l'estimation optimale. Les filtres peuvent ainsi être associés aux différents types de mouvement que peut prendre la cible. La non stationnarité des mouvements est également prise en compte.

Ce chapitre commence par une présentation de la plate-forme robotique, celle qui sert à évaluer nos algorithmes et nos lois de commande. La modélisation des robots du Chapitre 1 a abouti à l'écriture d'un logiciel de simulation. Une partie expérimentale et quelques simulations nous permettent de valider le principe de l'asservissement visuel adaptatif et prédictif. Enfin, ce chapitre se termine par une courte synthèse des méthodes proposées.

2 Présentation de la plate-forme expérimentale

A l'exception des théories mathématiques, une théorie ne sera considérée comme scientifique que si elle est confrontée à la réalité dont elle prétend apporter une connaissance. Elle a donc nécessairement un rapport au réel extérieur, avec des faits. Cela ne signifie pas qu'une expérimentation est nécessairement une preuve absolue, les faits ne confirment ni n'infirment totalement une théorie. Nous voulons, avec les expérimentations présentées dans ce travail de thèse, renforcer un peu plus l'intérêt de notre approche. Cela ne diminue en rien le caractère indispensable des phases de simulations. Elles permettent de se concentrer sur le fond d'un problème et d'occulter les problèmes annexes. Une application réelle est toujours un argument supplémentaire en faveur d'une théorie.

Les expérimentations faites au cours de ces travaux ont été effectuées sur une plate-forme composées de plusieurs unités de calcul, d'un système robotique du type bras manipulateur ainsi que d'une tête robotique portant deux caméras vidéo. La plate-forme est présentée par la Figure 5-1.

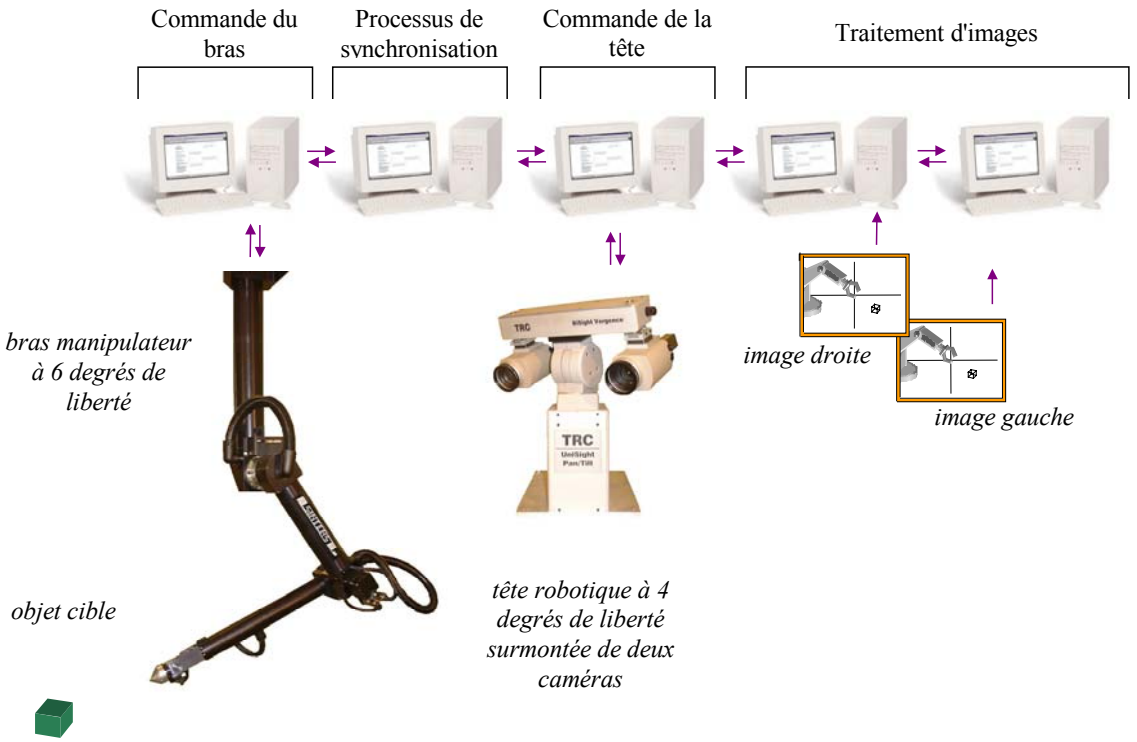


Figure 5-1 : La plate-forme robotique expérimentale.

L'architecture de la plate-forme est une architecture distribuée ; on peut y distinguer :

- un bloc d'acquisition et de traitement des paires d'images à chaque instant d'échantillonnage ;
- un bloc de commande et une tête robotique pour réaliser une rétroaction sur les paramètres du capteur à travers une stratégie de vision active, la rétroaction porte sur les quatre degrés de liberté de la tête (de la société américaine TRC) pour orienter les caméras seulement sur les parties importantes de la scène ;
- un bloc de commande et un bras manipulateur à six degrés de liberté (de la société Sinters).

Le traitement d'images est composé d'un seuillage à deux niveaux afin de sélectionner efficacement les pixels actifs des images. Cette façon de faire aboutit à des images binaires contenant uniquement les taches résultantes des amers visuels placés sur les objets importants de la scène (l'effecteur du robot et l'objet cible), et permet de s'affranchir des effets de bruit dus à l'éclairage. Les caractéristiques visuelle sont alors les positions des taches dans les images ; elles peuvent être calculées en temps réel.

Les différentes tâches sont distribuées sur plusieurs micro-ordinateurs de type PC avec le système d'exploitation Windows 2000. Les commandes des systèmes robotiques se font à l'aide de cartes PMAC ("Peripheral Module Access Controller") compatibles avec des langages de programmation évolués. Les processus communiquent entre eux et s'échangent des données via un réseau local en utilisant le protocole standard TCP/IP. Les temps de communication (environs 1 à 2

ms) sont compensés par la puissance répartie des micro-ordinateurs. Les actions qui interviennent dans la boucle de commande sont synchronisées par des processus logiciels. Une synchronisation matérielle par des cartes SCI ("Scalable Coherent Interface") est en cours de développement et permettra par la suite de réduire les temps de latence des échanges d'informations entre deux machines.

Un robot supplémentaire (à cinq degrés de liberté) est ajouté à la configuration de base présentée sur la Figure 5-1. Ce dernier sert à réaliser dans la pratique des mouvements dans l'environnement de travail : il constitue un excellent moyen de mouvoir un point cible dans un espace tridimensionnel tout en laissant la maîtrise de sa position. La plate-forme expérimentale permet alors de tester la robustesse réelle des algorithmes proposés.

3 Simulation d'une scène robotique

Réaliser des expérimentations avec des systèmes robotiques réels est, dans la pratique, contraignant. De plus, toutes les grandeurs ne peuvent pas toujours être maîtrisées. Il est impossible de s'affranchir des forces de frottements, de négliger certaines imperfections matérielles, ou d'omettre les délais, les temps de réponses et les temps de calculs. D'autres grandeurs sont inaccessibles. Pour diverses raisons, elles peuvent être non mesurables, cachées, peu fiables ou bruitées…

L'intérêt de la simulation réside alors dans la souplesse et la rapidité de mise en œuvre d'une scène robotique par rapport au système expérimental. Une scène robotique simulée est reconfigurable à loisir. Une seule instruction permet de positionner un robot, de le déplacer ou de le faire bouger sans poser des problèmes de sécurité et sans danger de collision.

Mais surtout, la scène robotique devient alors idéale : toutes les grandeurs sont disponibles, les signaux ne sont pas bruités, les temps de réponse sont maîtrisés et les robots répondent quasi-instantanément. Cet environnement d'évolution idéal n'est pas forcément une bonne chose pour le chercheur puisque les problèmes se retrouvent simplifiés et « nettoyés » de toutes imperfections. La commande d'un système est vouée à un système réel. Les résultats obtenus en simulations doivent de ce fait être relativisés, et doivent être complétés par des expérimentations réelles.

Vue notre problématique, la simulation de la scène robotique est un environnement idéal pour réaliser des systèmes d'apprentissage, évaluer leurs performances et analyser leurs comportements. La scène simulée permet justement d'utiliser dans un premier lieu des données représentatives du problème simplifié et complètement maîtrisé.

3.1 Autres environnements de simulations

Parmi les efforts consentis dans le domaine de la recherche en robotique, les efforts de simulations sont indispensables. Plusieurs chercheurs ont développés des environnements de simulation de scènes robotiques. On peut citer, par ordre chronologique, les environnements[1] :

- Robotica pour Mathematica de Nethery et de Spong (Nethery et Spong 1994) qui est réalisé en calcul symbolique ;
- Simderella de van der Smagt (van der Smagt 1994) qui requiert une interface graphique particulière ;
- la Robotics Toolbox de Corke (Corke 1996) pour Matlab, écrit dans un langage interprété qui peut s'avérer onéreux en terme de calculs (le même auteur a mis au point un package pour Maple) ;
- Roboot de Gourdeau (Gourdeau 1997) qui est une modélisation objet en langage C (le même auteur a développé une Mobile Robotics Toolbox pour Matlab) ;
- et Spacelib de Legnani (Legnani 1998) qui est écrit en langage C et en Matlab.

[1] Tous ces environnements peuvent être téléchargés sur Internet, les liens sont donnés dans la partie bibliographique.

La majorité des environnements de simulation de systèmes mécaniques articulés ne possède pas d'interface de visualisation ou possède des interfaces graphiques particulières ou peu courantes. Les outils développés en calcul symbolique, ou ceux développés en langage interprété, sont ralentis par les temps de calculs. Les outils mis au point (en langage C le plus souvent) ne permettent pas de maîtriser toutes les opérations matricielles. Dans certains environnements, il manque la modélisation d'un capteur de vision indispensable pour réaliser des tâches d'asservissement visuel.

Un environnement de simulation a également été mis au point au laboratoire TROP par Breton (Breton 1996). Ce dernier utilise le modèle lentille épaisse pour le capteur de vision (qui ne se justifie sans doute pas pour des tâches d'asservissement visuel). Cet environnement est générique, il permet de construire tout type de système mécanique articulé. Il n'est cependant pas portable sur l'environnement de la plate-forme réelle.

Ces différentes raisons nous ont conduit à élaborer notre propre environnement de simulation, Simulo.

3.2 Le logiciel Simulo

Simuler une scène robotique consiste à gérer une collection d'objets, des corps rigides, des systèmes mécaniques articulés et des capteurs de vision. Le logiciel Simulo gère et simule une scène robotique comprenant plusieurs systèmes robotiques, caméras et objets rigides.

La convention de Denavit-Hartenberg modifiée nous a permis de modéliser un robot à six degrés de liberté, un robot à cinq degré de liberté et une tête robotique à quatre degrés de liberté. Les détails de cette modélisation, ainsi que tous les paramètres utilisés, se trouvent dans (Perello 1999). Les objets rigides sont des ensembles de points solidaires les uns des autres pouvant être déplacés et orientés pour simuler soit des cibles fixes, soit des cibles mobiles. Les tâches d'asservissement visuel nécessitent un système sensoriel de vision. La modélisation du système de vision, deux caméras, utilise le principe de la projection perspective. D'autres projections peuvent être utilisées, telles les projections orthographique, affine ou para-perspective. Le modèle le plus détaillé de caméra est le modèle lentille épaisse. Le modèle sténopé (la projection perspective) est généralement compatible avec la précision requise par les tâches d'asservissement visuel même si certains travaux cherchent à tenir compte de la distorsion des lentilles (Breton 1999; Lange et al. 1999).

Le logiciel est basé sur une conception orientée objet et a été développé en langage C++. Si la partie calcul numérique est écrite en langage C++, les affichages sont réalisés dans l'environnement Matlab. La Figure 5-2 donne un aperçu de la scène simulée, avec notamment les deux robots, la tête robotique portant deux caméras et un corps rigide évoluant dans l'espace.

La modélisation de l'ensemble de la scène passe par l'implémentation de la représentation matricielle et du formalisme des matrices homogènes. Une bibliothèque d'outils mathématiques a spécialement été conçue et développée pour cela. Cette bibliothèque contient également différents types de réseaux de neurones, comme le réseau Adaline, le réseau de Miyamoto-Kawato-Setoyama-Suzuki (Miyamoto et al. 1988; Soucek 1989), le Mélange d'Expert, la carte auto-organisatrice de Kohonen et ses extensions (NeuroModule et ESOM), mais aussi des outils du traitement du signal comme des filtres numériques, des filtres blanchisseurs, le filtre de Kalman, le SMAKF, le SOKF, etc.

Le logiciel Simulo, avec sa bibliothèque d'outils permet de considérer divers aspects de la commande robotique : chaînes cinématiques, contraintes, contrôleurs associés aux robots, simulations de perception et boucles perception-action. L'avantage de Simulo réside dans le fait que les contrôleurs sont compatibles, aussi bien pour la scène simulée, que pour la plate-forme expérimentale.

Les boucles simulées de perception-action sont gérées en « différentiel ». Elles utilisent un modèle dynamique de caméra, c'est-à-dire un modèle qui retourne les grandeurs projetées sur les images avec un temps de retard. D'autres travaux utilisent également ce modèle dynamique de caméra (Gangloff 1999) pour des applications d'asservissement visuel rapides.

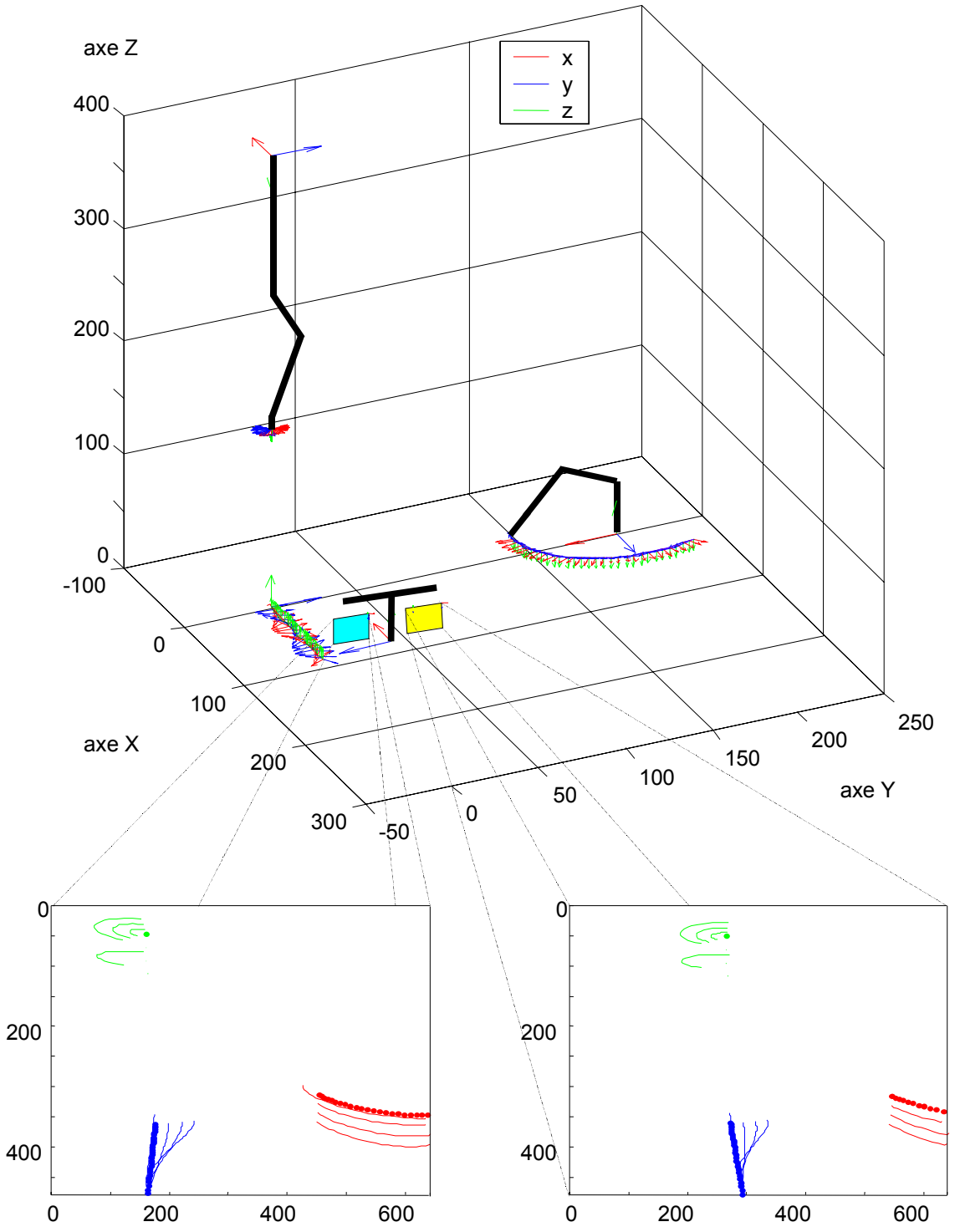


Figure 5-2 : La scène simulée avec deux bras manipulateurs, un objet cible et un capteur stéréoscopique.

Dans la scène, les deux bras manipulateurs sont des robots à 6 et à 5 degrés de liberté. Le capteur stéréoscopique est constitué d'une tête robotique à 4 degrés de liberté et de deux capteurs. Chaque position et chaque orientation des effecteurs des robots est représentée par un repère dans l'espace tridimensionnel. Les trajectoires projetées sur les images gauche et droite sont représentées en bas (en gras). Des points supplémentaires on été rajoutés, avec un lien fixe par rapport à l'effecteur, pour mettre en évidence l'évolution des trajectoires (en trait fin dans les images).

L'originalité des approches développées réside dans la recherche de compromis, modulables entre le respect des lois de la physique et la rapidité des calculs.

4 Réalisations de contrôleurs prédictifs

Cette dernière partie présente des résultats de simulation et des résultats expérimentaux obtenus sur la plate-forme robotique.

Nous validons dans un premier temps l'approche des ME en estimant des fonctions, notamment de fonctions robotiques inverses. L'apprentissage est ensuite étendu à l'estimation de la transformation sensori-motrice, pour être finalement appliqué à une tâche de positionnement par asservissement visuel.

D'autres expérimentations concernent l'estimation du mouvement par le filtre de Kalman adaptatif SMAKF et par le mélange de SMAKF. Cette dernière approche est validée en prédisant la trajectoire d'un objet dans l'espace des images. L'estimation du mouvement est faite à partir des mesures de position, vitesse et accélération.

Pour finir, nous présentons un schéma complet d'asservissement visuel adaptatif et prédictif. Un neuro-contrôleur en approchant la transformation sensori-motrice convertie les informations visuelles en consignes articulaires. Un mélange de SMAKF estime les mouvements de l'objet à suivre et permet d'introduire les compensations nécessaires pour réduire les erreurs de traînage. Cette loi de commande, qui intègre un prédicteur dans la boucle de retour d'informations visuelles, est évaluée sur des tâches de suivi de cible et des tâches de vision active à la fois par des simulations et par des expérimentations réelles.

4.1 Application du mélange d'experts à la commande

4.1.1 Apprentissage de la transformation entre deux espaces multidimensionnels

a Objectif

Une première application servira d'appui pour valider l'approche du ME. Elle consiste à réaliser l'approximation de la transformation reliant deux espaces à plusieurs dimensions. L'estimation faite est du type linéaire locale. Nous montrerons que l'apprentissage et le découpage en régions de l'espace des entrées est bénéfique au processus d'apprentissage. Cette simulation permet essentiellement de comprendre et de valider l'association intelligente d'un réseau expert à une portion de l'espace d'entrée faite par le réseau superviseur softmax. Des comparaisons avec des réseaux superviseurs plus basiques sont présentées.

b Description

Considérons deux espaces distincts $X \in \mathbb{R}^4$ et $Y \in \mathbb{R}^2$. Soit F(.) la fonction vectorielle permettant d'associer à chaque point $\mathbf{x}$ de X un point $\mathbf{y}$ dans Y. La transformation de l'espace X vers l'espace Y est considérée comme non connue. Afin de donner une idée de la complexité de la fonction F(.) à apprendre, nous dirons qu'elle est constituée de fonctions trigonométriques, des fonctions du type :

$$y = \alpha \frac{\cos(x^2)}{x}. \tag{5.1}$$

Si $\mathcal{X}$ est une trajectoire dans X et que l'image de cette trajectoire dans Y soit $\mathcal{Y}$= F($\mathcal{X}$), nous nous proposons d'apprendre la matrice Jacobienne reliant un petit déplacement $\Delta\mathbf{x} \in \mathcal{X}$ à un petit déplacement de la trajectoire $\mathcal{Y}$, $\Delta\mathbf{y} \in \mathcal{Y}$. L'apprentissage de cette matrice a été effectué par un mélange

d'experts supervisés par plusieurs algorithmes (réseaux superviseurs). Le but affiché de ces simulations est la validation du processus d'apprentissage.

Les réseaux experts sont des réseaux Adalines et le réseau superviseur est un réseau mono-couche dont les poids sont régis par un apprentissage basé sur la règle du softmax. Un espace tridimensionnel des indices des réseaux experts est considéré pour définir la position d'un réseau expert (l'indice dans l'expression (3.4) est remplacé par un vecteur). Chaque réseau expert est indexé par des valeurs i, j et k. Pour réaliser l'approximation de la matrice Jacobienne introduite précédemment, un ME comportant $2 \times 4 \times 5$ experts est utilisé.

L'apprentissage du réseau superviseur permet de réaliser la sélection d'un neurone de sortie de la même façon que le principe WTA. Pour chaque entrée présentée, le réseau superviseur choisit un expert vainqueur.

Un certain nombre de tests comparatifs a été réalisé, plus précisément avec des ME identiques avec le même type de réseaux experts, avec le même nombre d'experts, mais avec des réseaux superviseurs différents. C'est dans cette optique qu'un processus aléatoire a été utilisé en tant que réseau superviseur. Ce réseau superviseur sélectionne un expert aléatoirement parmi tous, lui affecte une pondération unité et met une pondération nulle à tous les autres réseaux experts. Un réseau superviseur dit « à mémoire courte » a été défini et utilisé dans les comparaisons. Ce dernier sélectionne le dernier réseau expert le plus performant (à l'itération k-1) pour répondre à l'itération courante k. Les réponses des autres réseaux experts sont alors pondérées par des coefficients nuls. D'autres réseaux superviseurs ont été utilisés dans les tests comparatifs. Ils reprennent le principe de la mémoire courte et associent dans la réponse globale du ME, le dernier réseau le plus performant ainsi que des experts voisins, soit les 6, soit les 18 plus proches experts voisins dans l'espace des indices. Le réseau expert vainqueur se voit attribuer 50% de la somme des pondérations et les réseaux experts voisins se partagent alors les 50% qui restent. La motivation principale que nous voyons à cette dernière approche est de vérifier la théorie et de valider l'apprentissage en cherchant à voir si plusieurs experts ne répondent pas mieux qu'un seul à la fois.

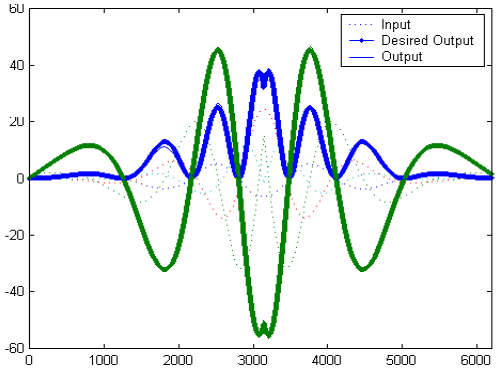
c Résultats

Les résultats des tests comparatifs sont résumés dans le Tableau 5.1. Les chiffres contenus dans ce tableau reflètent le comportement de chaque algorithme sur une vingtaine de simulations identiques. Les performances des ME, avec les réseaux superviseurs qui viennent d'être définis sont comparées aux performances d'un réseau Adaline unique.

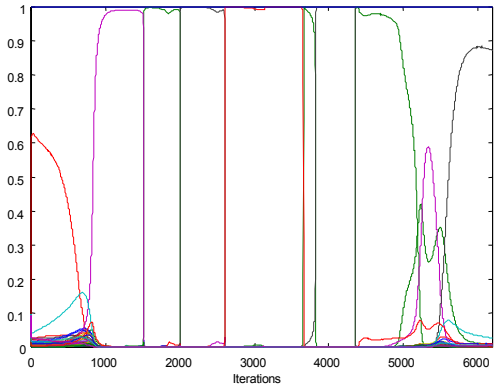
Réseau superviseur	Erreur moyenne absolue		Écart type		Erreur maximale absolue	
	e_1	e_2	e_1	e_2	e_1	e_2
Réseau Adaline unique	3.484	3.825	4.206	4.229	21.981	18.469
Processus aléatoire	2.078	3.271	3.262	3.593	17.615	17.816
Mémoire courte avec 18 voisins	1.183	2.352	1.955	2.449	8.796	10.862
Mémoire courte avec 6 voisins	1.183	2.352	1.955	2.449	8.796	10.862
Mémoire courte sans voisinage	1.183	2.352	1.955	2.449	8.796	10.862
Réseau superviseur softmax	0.345	0.310	0.381	0.2682	3.150	1.632

Tableau 5.1 : Erreurs d'approximation du ME avec différents réseaux superviseurs.

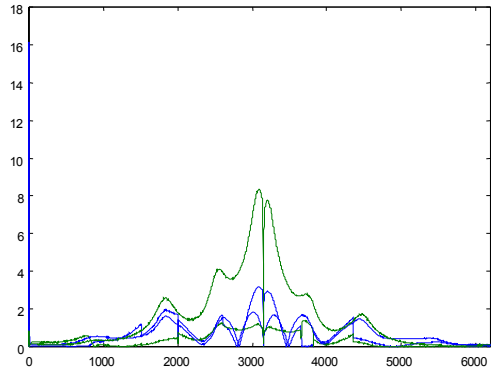
Les coefficients d'apprentissage des experts organisés au sein de mélanges basés sur les réseaux superviseurs aléatoire et à mémoire courte ne sont pas pondérés par les coefficients g_i. Tous les experts utilisent un coefficient d'apprentissage identique ; seules les sorties des experts sont pondérées (les erreurs sont exprimées sans unité).



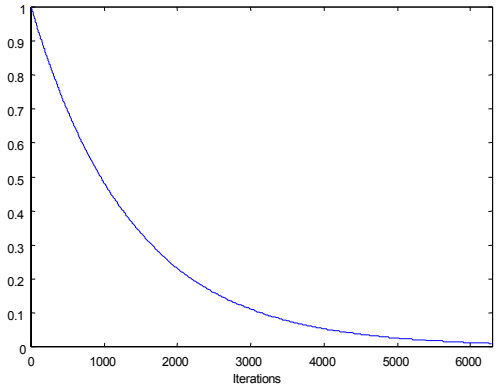
a) Les signaux d'entrée, les sorties désirées et les sorties estimées



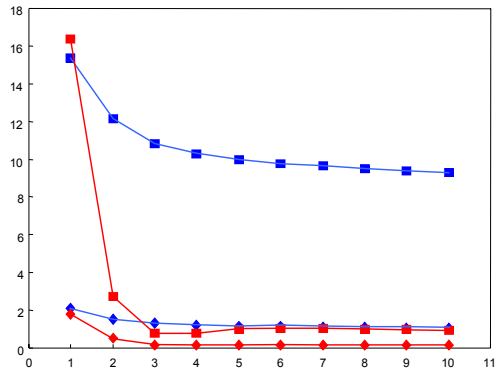
b) Les coefficients de pondération g_i calculés par le réseau superviseur



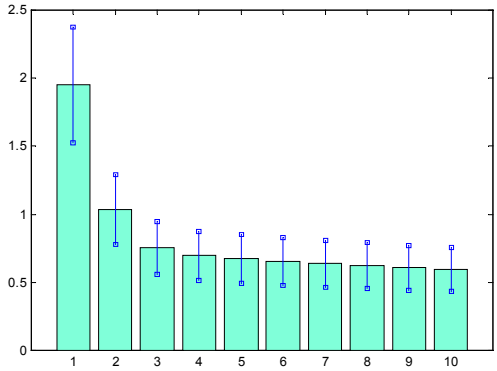
c) Les erreurs absolues d'estimation (sans signe)



d) L'évolution du coefficient d'apprentissage η



e) Les erreurs, moyennes (◆) et maximales (■) après plusieurs passages sur la trajectoire $\mathcal{Y}$



f) Les erreurs moyennes après plusieurs passages sur la trajectoire $\mathcal{Y}$

Figure 5-3 : Apprentissage de la Jacobienne de la relation entre deux espaces multidimensionnels.

La figure en haut à gauche montre les trajectoires dans les espaces de départ et d'arrivée, ainsi que la trajectoire estimée dans l'espace d'arrivée. L'erreur absolue entre la trajectoire idéale et l'estimation est représentée à gauche sur la figure du milieu. La figure en bas à gauche montre l'évolution de l'erreur moyenne et maximale (chaque point est une valeur moyenne sur une séquence de 100 données). Les pondérations des réseaux experts sont représentées en haut à droite, et l'évolution du coefficient d'apprentissage du ME au milieu sur la figure de droite. La figure en bas à droite montre l'évolution du vecteur d'erreur et son écart type après plusieurs passages sur la trajectoire $\mathcal{Y}$ (chaque barre représente une valeur moyenne sur une séquence de 100 données).

Le réseau Adaline unique est configuré de telle sorte à ne pas mémoriser les données explicatives de la transformation. Ceci est réalisé en maintenant élevé le coefficient d'apprentissage durant toute la phase d'apprentissage et durant toute la phase de restitution (≈ 0.8). Ce réseau Adaline est donc adaptatif et avec uniquement une mémoire à court-terme.

L'erreur d'approximation servant de base de test est l'erreur quadratique moyenne $\mathbf{e}$ définie dans l'espace Y entre la valeur estimée et la valeur désirée (donc théorique) :

$$\mathbf{e} = [e_1 \quad e_2]^T = [(\Delta\hat{y}_1 - \Delta y_1)^2 \quad (\Delta\hat{y}_2 - \Delta y_2)^2]^T \tag{5.2}$$

Les trajectoires $\mathcal{X}$ et $\mathcal{Y}$ sont composées de 6000 points chacune. Les courbes de la Figure 5-3 permettent d'illustrer les simulations, ainsi que les résultats obtenus. Le Tableau 5.1 montre clairement l'inaptitude du réseau Adaline à apprendre la transformation. Les résultats du ME avec un réseau superviseur aléatoire montrent son inefficacité. Le réseau superviseur aléatoire ne permet pas d'associer correctement une région de l'espace d'entrée à un réseau expert. La conséquence directe d'un tel comportement se traduit par le fait que chaque réseau expert se trouve obligé d'apprendre à partir des points distribués sur l'ensemble de l'espace d'entrée et est alors incapable d'apprendre correctement la transformation.

Les ME qui utilisent un réseau superviseur à mémoire courte réalisent un apprentissage meilleur que celui qui utilise un réseau superviseur aléatoire. Notons cependant que plus le voisinage est grand, plus les résultats sont mauvais. Ceci s'explique par le fait qu'il n'existe aucun lien entre les experts, hormis celui du voisinage dans l'espace des indices. En d'autres termes, une approche géométrique ne fonctionne que si elle a une signification concrète par rapport à la tâche à apprendre. Une approche géométrique, avec un voisinage défini dans l'espace des entrées, est correct puisqu'il permet de préserver les propriétés topologiques de ce dernier. C'est l'idée qui est reprise dans les cartes auto-organisatrices.

Le ME le plus performant est celui qui utilise la règle d'apprentissage softmax. Ce ME est capable d'effectuer une sélection cohérente pour un vecteur présenté de l'espace d'entrée. Dans cette configuration, un seul réseau expert domine très largement les autres pour un vecteur d'entrée donné. Ceci permet à plusieurs experts de se relayer pendant l'apprentissage de la trajectoire. Chacun peut se spécialiser uniquement sur une portion de l'espace des entrées. Les changements de réseaux experts peuvent intervenir très rapidement. On voit, toujours sur les courbes de la Figure 5-3, que le ME avec le softmax continue à apprendre et à affiner son approximation de la Jacobienne lorsqu'on réalise plusieurs passages sur la même trajectoire.

4.1.2 Apprentissage du modèle géométrique inverse d'un robot trois axes

L'approche modulaire du ME est essentiellement utilisée pour des tâches de classification (Waterhouse 1997). L'implication des ME dans le domaine de la robotique est rare. Très peu de travaux se sont intéressés à l'application de l'architecture des ME au contrôle robotique. Parmi les rares travaux, citons ceux de Jacobs et Jordan (Jacobs et Jordan 1993). Ils se sont intéressés à commander un robot comportant deux degrés de liberté dans un plan, lorsqu'une masse qui modélise une charge est placée à l'extrémité du robot et varie pendant un déplacement. Une approche du type "gain scheduling" est proposée. Elle permet essentiellement d'éviter de concevoir un contrôleur global et unique. Dans l'application proposée, l'espace des paramètres du système est partitionné en autant de régions que d'experts. Pour chaque région, les auteurs introduisent des connaissances supplémentaires via des transformations, des fonctions $f_i(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})$ dépendant des vecteurs positions, vitesses et accélérations angulaires ($\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$ et $\ddot{\boldsymbol{\theta}}$). Ces transformations permettent de compenser le caractère fortement non linéaire de la dynamique du système robotique.

Une approche différente mais néanmoins proche des ME est développée par Bishop (Bishop 1994). Son objectif est identique, à savoir contrôler et commander un bras robotique (à deux degrés de liberté) par l'apprentissage du modèle inverse en approximant la densité de probabilité conditionnelle des sorties théoriques par rapport aux entrées. L'architecture du réseau proposé diffère quelque peu de l'approche ME. Bishop utilise une approche MDN ("Mixture Density Network") qui est une combinaison séquentielle de deux réseaux de neurones. Les vecteurs d'entrée sont présentés à un premier réseau, un réseau "feed-forward" qui fournit en réponse les paramètres du second réseau, un mélange de modèles. Ce dernier réseau représente la densité de probabilité conditionnelle des variables désirées pour les vecteurs d'entrée connus. Il est essentiellement réalisé par des fonctions à noyaux. Divers types de fonctions peuvent être utilisés, les fonctions Gaussiennes restent cependant les plus courantes.

Jacobs et Jordan prennent comme entrée du contrôleur les variables angulaires $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$ et $\ddot{\boldsymbol{\theta}}$ pour commander le robot par des consignes en couple. Bishop apprend le modèle cinématique inverse du robot.

Dans la section qui suit, nous nous sommes arrêté à l'apprentissage de fonctions robotiques par une approche du type ME, l'objectif final étant la commande robotique et plus particulièrement la réalisation de tâches de positionnement. Pour cela, une première série de simulations a été effectuée. L'originalité de ces travaux réside dans le fait que l'approche ME réalise un apprentissage complet du modèle inverse du système sans connaissance préalable. Ces simulations montrent qu'il est possible d'apprendre le modèle géométrique ou différentiel inverse d'un robot à deux degrés le liberté. Nous avons utilisés des réseaux Adalines comme réseaux experts, et des réseaux mono ou multicouches pour le réseau superviseur. Précisons que la commande n'est pas du type "gain scheduling" comme dans (Jacobs et Jordan 1993), mais est un contrôle adaptatif inverse classique comme celui illustré par la Figure 5-4.

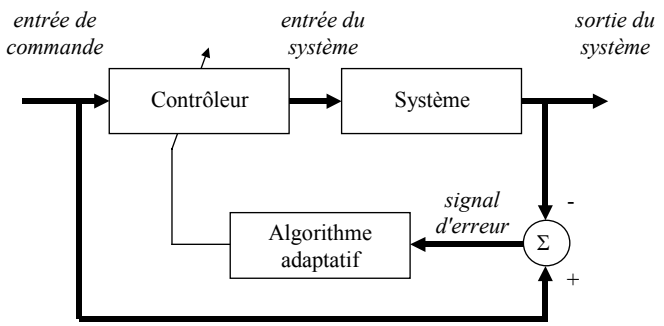
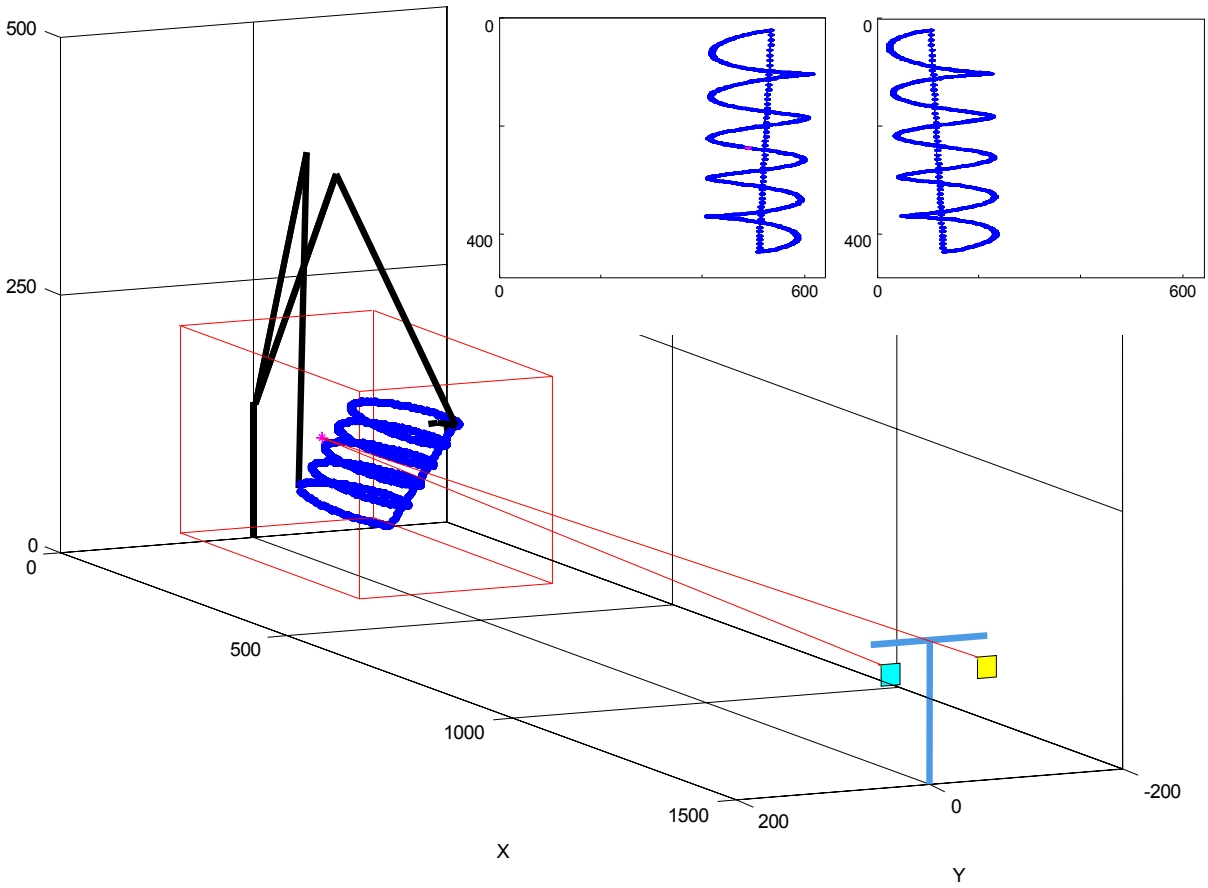
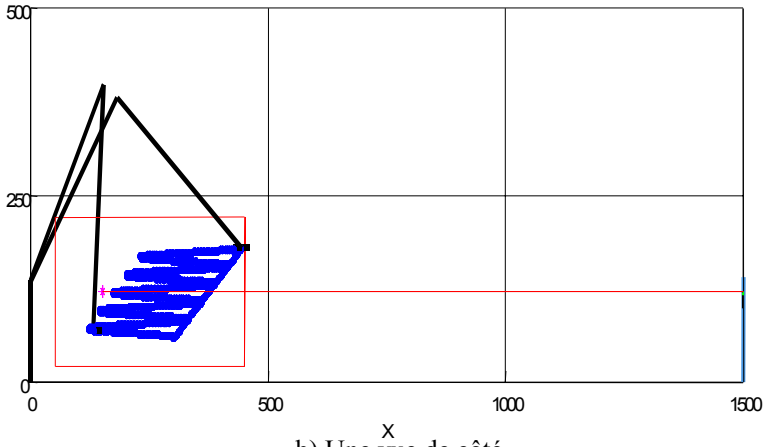


Figure 5-4 : Concept général du contrôle adaptatif inverse.

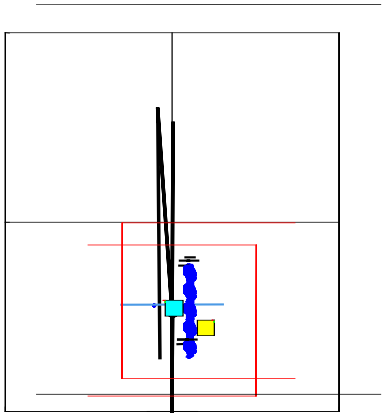
Dans un premier temps, l'apprentissage porte sur le modèle géométrique inverse et sur le modèle différentiel inverse d'un robot à deux degrés de liberté. Ces cas d'école nous ont servi à évaluer les capacités d'apprentissage de l'architecture ME dans le contexte particulier du contrôle robotique. Les propriétés d'apprentissage et les performances des ME ont dans un premier temps été évaluées dans (Hrycenko 2000). Ces résultats concernent l'estimation et l'apprentissage de fonctions multidimensionelles. D'autres résultats sont présenté dans (Cexus 2001), ils concernent plus particulièrement les applications robotiques. Plutôt que de présenter ces premiers résultats, nous allons nous intéresser directement à l'apprentissage d'un système robotique associé à un système de vision pour une tâche de positionnement dans un espace tridimensionnel.



a) Une vue en perspective de la scène robotique



b) Une vue de côté



c) Une vue de derrière

Figure 5-5 : Plusieurs vues d'une scène simulée.

En haut, une vue en perspective, en bas une vue du côté et une vue de la scène lorsque l'observateur se trouve derrière les caméras. Le robot est représenté avec deux positions différentes. L'étoile représente le point test projeté dans les plans des caméras. Le parallélépipède montre la partie de l'espace de travail dans laquelle sont tirés les points pour l'apprentissage des réseaux neuromimétiques. L'effecteur du robot suit une trajectoire hélicoïdale.

4.1.3 Tâche de positionnement

a Objectif

La tâche de positionnement peut être décrite comme l'asservissement d'un bras pour atteindre un point cible statique dans son espace de travail. Dans cette application, l'asservissement est réalisé par une approche dynamique "look and move". Les consignes pour contrôler un robot à trois degrés de liberté sont calculées à partir des informations visuelles délivrées à la cadence vidéo par deux caméras fixes et déportées. Le contrôleur n'utilise aucune connaissance a priori sur le système robot-vision. Il est basé sur un apprentissage de la relation entre variation dans l'espace des caractéristiques visuelles et variation dans l'espace articulaire, autrement dit sur l'estimation du modèle différentiel inverse du système robotique et du système sensoriel : la matrice Jacobienne de la transformation inverse de l'espace de commande à l'espace capteur.

L'apprentissage du neurocontrôleur est effectué via l'architecture ME. Le réseau superviseur utilise un apprentissage du type softmax et les réseaux experts sont des réseaux Adalines. Afin d'évaluer plus précisément les performances de ce contrôleur, trois contrôleurs neuronaux vont être comparés. Pour la même tâche, le ME est confronté à un NeuroModule avec une sélection effectuée dans l'espace des images, et un NeuroModule avec une sélection réalisée dans l'espace articulaire du robot. Cette application est simulée et est décrite ci-dessous.

b Description de la scène simulée

La scène simulée dans cette application ressemble à celle présentée sur la Figure 5-5. Nous avons voulu que cette scène soit aussi proche que possible de celle de Ritter, Martinetz et Schulten et qui est décrite dans (Ritter et al. 1992, pages 182-183). La même configuration sera utilisée dans des simulations présentées plus loin dans ce document, on peut donc se permettre de donner quelques détails. Les unités que nous adoptons dorénavant sont les millimètres (mm) pour les distances et les degrés (°) pour les angles.

La scène est définie par un repère orthonormé qui lui est associé. Cette scène comprend un robot du type bras manipulateur à trois degrés de liberté permettant à son organe terminal d'effectuer des déplacements dans l'espace tridimensionnel. Les mouvements du bras robotique sont observés par deux caméras déportées et fixes. Les longueurs des éléments du robot sont $l_1 = 130$, $l_2 = 310$ et $l_3 = 330$. Sa base est située à l'origine du repère associé à la scène, c'est-à-dire en $[0 \quad 0 \quad 0]^T$. Les capteurs des caméras sont constitués de 640×480 pixels de dimensions 11×13 µm ; leur distance focale est la même, f = 50. La caméra gauche est placée en $[1500 \quad -50 \quad 120]^T$, la caméra droite en $[1500 \quad 50 \quad 120]^T$. Les deux caméras sont orientées de la même façon, cette orientation est définie par les angles d'Euler $[90 \quad -90 \quad -90]^T$. Les plans de projection des deux caméras se trouvent ainsi inclus dans un même plan. En considérant tous les paramètres précédents, le point $[150 \quad 0 \quad 120]^T$ de l'espace tridimensionnel se projette en $[488.35 \quad 240]^T$ dans l'image gauche et en $[151.16 \quad 240]^T$ dans l'image droite.

L'espace articulaire du bras manipulateur est défini par, $-75 \le \theta_1 \le 75$, $0 \le \theta_2 \le 90$ et $-135 \le \theta_3 \le 45$, soit un espace qui autorise les redondances de positionnement du robot pour un point de l'espace 3-D. La zone de travail est également délimitée dans l'espace tridimensionnel par : $x \in [50,\ 450]$, $y \in [-100,\ 100]$ et $z \in [20,\ 200]$. Les points utilisés pour l'apprentissage ne sont tirés que dans cette zone. De cette façon, un pixel de l'espace des images représente environ une distance de 0.25 mm au milieu de cette zone.

c Apprentissage et résultats

L'approche neuronale qui consiste à relier une erreur de position dans l'espace des images (l'espace capteur) à la correction angulaire à appliquer au robot, revient à estimer le modèle différentiel inverse du système robot-vision. Dans l'application qui suit, seule la position de l'organe terminal du

robot est gérée. Le vecteur d'entrée du réseau est alors l'erreur de position mesurée dans l'espace image entre l'effecteur du robot et le point cible. Pour la phase d'apprentissage, la sortie désirée est la consigne angulaire idéale pour corriger la position du robot vers le point cible.

Le réseau ME proposé est comparé à deux NeuroModules fonctionnant de manière différente. L'opération de sélection des réseaux Adalines par la carte de Kohonen se fait, soit dans l'espace des caractéristiques image, soit dans l'espace articulaire du robot.

L'approche ME a été évaluée avec deux réseaux de tailles différentes, avec $5 \times 12 \times 12 = 720$ et avec $12 \times 14 \times 18 = 3024$ réseaux experts (des Adalines). Le nombre d'experts répartis suivant chaque dimension de l'espace des indices n'est pas un critère influent dans le fonctionnement et l'apprentissage du ME. C'est le réseau superviseur qui réalise le découpage de l'espace des entrées ; il se base uniquement vecteurs d'entrée et sur les performances des experts par rapport aux entrées.

La répartition des neurones dans les dimensions de l'espace des indices est par contre cruciale pour les réseaux de Kohonen. La taille de la grille de neurones est un facteur déterminant dans les performances de la carte SOM et du NeuroModule. Comme les neurones doivent se répartir et discrétiser l'espace des entrées, la « forme » de la grille permet dans un sens, de définir le pas de quantification. Ses considérations nous ont amené à choisir des NeuroModules de tailles différentes selon le choix de l'espace de sélection. Ainsi, pour des NeuroModules qui effectuent la sélection dans l'espace des images, des grilles de $5 \times 12 \times 12 = 720$ et de $16 \times 12 \times 16 = 3072$ neurones (donc autant de réseaux Adalines) ont été choisies. Pour les NeuroModules qui effectuent l'opération de sélection dans l'espace articulaire, les grilles ont été fixées à $7 \times 9 \times 12 = 756$ et à $12 \times 14 \times 18 = 3024$ neurones (donc 3024 réseaux Adalines).

Pour les comparaisons, on tiendra compte de la taille des réseaux, donc du nombre d'Adalines dans chaque réseau. Les « petits » réseaux comprennent environs 700 Adalines et les « grands » réseaux environs 3000 Adalines. Les Adalines pour tous les réseaux présentés ont les mêmes dimensions, fixées par le contexte de l'application. Ils possèdent 3 sorties, les corrections angulaires pour chacun des 3 axes du bras robotique, et 3 entrées, qui sont 3 vitesses observées dans l'espace des caractéristiques images. Comme les plans des caméras sont parallèles, les projections d'un point dans l'espace tridimensionnel se projettent selon cette configuration sur les deux images avec la même ordonnée.

Réseau superviseur	Erreur moyenne absolue (°)			Erreur maximale absolue (°)		
	$\Delta\theta_1$	$\Delta\theta_2$	$\Delta\theta_3$	$\Delta\theta_1$	$\Delta\theta_2$	$\Delta\theta_3$
NeuroModules avec sélection articulaire de $12 \times 14 \times 18 = 3024$ Adalines	1.751	1.573	1.511	5.123	6.574	6.676
NeuroModules avec sélection image de $16 \times 12 \times 16 = 3072$ Adalines	1.871	1.719	1.714	5.511	5.524	5.568
ME avec $12 \times 14 \times 18 = 3024$ Adalines	1.900	1.782	1.725	5.739	5.901	5.770
NeuroModules avec sélection articulaire de $7 \times 9 \times 12 = 756$ Adalines	7.151	6.847	6.925	12.358	11.087	13.215
NeuroModules avec sélection image de $5 \times 12 \times 12 = 720$ Adalines	7.245	7.417	7.295	13.047	12.483	17.581
ME avec $5 \times 12 \times 12 = 720$ Adalines	5.157	4.961	4.980	14.021	13.247	13.351

Tableau 5.2 : Erreurs d'apprentissage des réseaux ME comparées à celles de NeuroModules.

Dans ce tableau, les trois premières lignes présentent les résultats des réseaux de tailles plus importantes et les trois suivantes les résultats des réseaux de tailles plus petites.

Une phase d'apprentissage préliminaire précède une manipulation de contrôle robotique. Dans une moindre mesure, les réseaux neuronaux poursuivent leur apprentissage durant les tâches de positionnement. En fait, lors de leur utilisation subséquente, l'apprentissage des poids synaptiques est très peu autorisé (en fixant des coefficients d'apprentissage faible) n'autorisant qu'une certaine plasticité.

La phase d'apprentissage permet à chaque réseau d'apprendre correctement la totalité de l'espace de travail du robot. La base d'apprentissage est constituée d'un jeu de 140 000 points. Les résultats sont présentés dans le Tableau 5.2 après 5 époques d'apprentissage, ils représentent la restitution sur une trajectoire hélicoïdale qui est présentée sur la figure précédente (Figure 5-5). Les chiffres de ce tableau sont des valeurs moyennes obtenues sur une centaines de simulations, avec des conditions initiales différentes concernant les réseaux de neurones. La vitesse de cette trajectoire dans l'espace tridimensionnel est en moyenne de 0.15 m/s (max. 0.05 m/s pour une période d'échantillonnage de 33 ms).

L'erreur utilisée est l'erreur moyenne absolue entre la trajectoire suivie par le robot et celle, idéale, définie dans l'espace articulaire.

Les réseaux sont toujours comparés avec des tailles équivalentes, et les comparaisons avec les NeuroModules nous ont permis de constater que l'approche ME permet une meilleure estimation de la matrice d'interaction pour un nombre de neurones moindre (simulation avec les « petits » réseaux). L'apprentissage du NeuroModule est plus rapide lorsque la taille des réseaux est plus grande. Il profite de la notion du voisinage et permet une estimation légèrement plus précise. De manière générale il faut une grande quantité d'exemples pour réaliser un apprentissage correct avec l'approche ME.

Le fait d'augmenter le nombre de neurones dans les cartes auto-organisatrices des NeuroModules permet d'augmenter la précision de l'approximation. La précision est directement liée au nombre de neurones dans l'approche de Kohonen. Cette vérité n'est pas transposable à l'approche ME. Il est impossible, de prédéterminer la taille idéale du réseau ME, même en connaissant la taille des espaces à apprendre. L'approche des ME nécessite un certain nombre de réseaux experts pour une complexité donnée d'un problème. En deçà de ce nombre les performances sont faibles, au delà elles stagnent.

4.1.4 Expérimentation de vision active

a Objectif

La vision active consiste à adapter les paramètres du capteur de vision en fonction des données mesurées afin d'augmenter son acuité visuelle. Le principe général de la vision active est présenté sur la Figure 5-6.

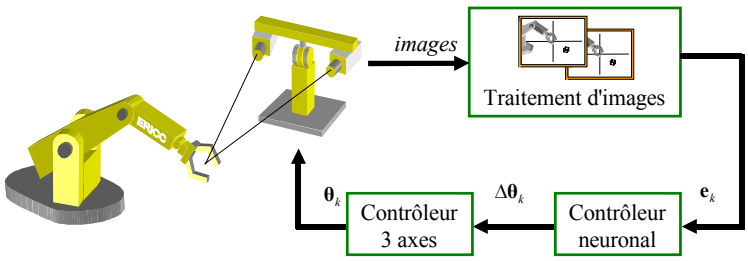


Figure 5-6 : Principe de la vision active.

Un objet cible de la scène, l'organe terminal d'un bras robotique par exemple, se doit d'être centré dans les deux images d'un capteur stéréoscopique. La boucle de retour d'informations visuelles est au service du capteur de vision. La tête doit en permanence orienter les caméras pour que l'objet soit conservé au centre de chaque image.

Des expérimentations ont été menées sur la plate forme robotique présentée ci-dessus. Dans un premier temps, l'objectif consiste à centrer des points fixes dans les deux images du capteur stéréoscopique. Celui-ci comporte quatre degrés de liberté. Le neuro-contrôleur réalise l'asservissement des axes panoramique et azimutal. Les degrés de liberté supplémentaires, les angles de vergence de chaque caméra, sont contraints géométriquement.

La commande est réalisée par un contrôleur ME et ses performances sont comparées à celles réalisées par un réseau Adaline unique et à celles d'un contrôleur proportionnel.

b Conditions expérimentales et résultats

La tâche de vision active est réalisée à l'aide de la tête robotique représentée sur la Figure 1-3. Sur cette tête, sont montées deux caméras CCD ("Coupled Charged Device"). Des images d'une résolution de 640×480 pixels et avec un niveau d'intensité défini sur 8 bits sont transférées en DMA ("Direct Memory Access") dans la mémoire vive du processeur pour être traitées.

L'asservissement se fait sur des points cible fixes situés à une distance de 1 m du socle de la tête. A cette distance, un pixel dans les images recueillies représente une longueur de 10 mm dans l'espace tridimensionnel.

Les résultats sont présentés dans (Cexus 2001) et montrent que dans la pratique 7 à 8 itérations permettent de centrer un objet cible dans les deux images du capteur stéréoscopique avec un mélange de 12 Adalines. En théorie (en simulations), deux itérations sont suffisantes pour la même configuration de la scène. A titre comparatif, la même expérimentation a été effectuée sur la plate-forme avec un Adaline unique et avec une simple correction proportionnelle par axe. Un Adaline unique nécessite 10 itérations et la correction proportionnelle requiert environ 12 itérations pour centrer la cible dans les images. Chaque manipulation a été répétée une trentaine de fois sur la plate-forme expérimentale.

4.2 Estimation du mouvement

4.2.1 Introduction et objectif

Les capteurs de vision permettent de mesurer l'évolution de la scène de manière distante et chaque image numérique contient une quantité d'informations considérable. Les progrès de la technologie des capteurs de vision et ceux du traitement des images ont contribués à rendre effectif l'utilisation des données visuelles dans la boucle de commande des robots. Si le traitement d'images réalise une interprétation des images, encore faut-il pouvoir tenir compte de ces informations, représentatives de l'état de la scène dans le contrôle robotique. Il en découle une conception de nouveaux types de contrôleur.

Nous allons dorénavant nous intéresser à des tâches de poursuite de cibles mobiles. Il s'agit toujours d'une tâche d'asservissement visuel, mais la cible peut être considérée comme nouvelle à chaque paire d'images acquise. Un contrôleur permettant d'atteindre des cibles statiques (c'est une tâche de positionnement) n'est dès lors plus satisfaisant. Le problème devient dynamique lorsque la cible se déplace. Améliorer les performances, mais aussi assurer la robustesse de la boucle de commande visuelle, se fait en tenant compte explicitement des incertitudes et des changements survenants dans l'environnement du robot.

C'est dans ce contexte que nous proposons d'introduire dans la boucle de commande de haut niveau, un algorithme de prédiction capable de stocker une forme de connaissance, et capable d'intégrer des incertitudes. Cet algorithme a pour vocation de tenir compte des incertitudes de la scène elle-même, des mouvements d'une cible associés à aucun modèle et qui relèvent souvent d'un caractère non stationnaire, ou de tout autre événement inattendu. Cet algorithme doit également compenser les erreurs de modélisation (le modèle est une approximation issue d'un processus d'apprentissage) et les

imperfections du système robot-vision. Le robot est un système mécanique articulé avec des défauts, il n'est pas calibré, pas plus que le capteur de vision. Il est d'ailleurs extrêmement difficile de parfaitement contrôler les paramètres d'une caméra.

Nous allons donc estimer les déplacements des objets dans l'espace de vision par l'approche développée dans le Chapitre 4, le SMAKF ou plus précisément le Mélange de SMAKF. Les informations visuelles prédites sont alors utilisées pour la commande robotique pour suivre et saisir des objets en mouvement. Estimer le mouvement des objets cibles permet d'assurer un suivi correct, c'est-à-dire avec une erreur de traînage minimale.

4.2.2 Principe

Chaque acquisition faite par le système de vision donne la dernière observation de la scène. Lorsque les caractéristiques images sont extraites, elles sont données en entrée du mélange de SMAKF. Ces caractéristiques, les coordonnées de l'objet cible et celles de l'effecteur du robot dans les images gauche et droite, sont alors prédites. Le principe d'adaptation et d'apprentissage du mélange de SMAKF est schématisé sur la Figure 5-7. L'apprentissage et l'adaptation sont réalisés en ligne après chaque acquisition d'une nouvelle paire d'images. Rappelons que l'adaptation concerne la matrice de transition de tous les réseaux experts du mélange et que l'apprentissage concerne le réseau superviseur du mélange.

Dans toutes les simulations et les expérimentations qui vont suivre, aucune estimation n'est faite pendant les deux premières itérations, puisque la détermination de la matrice de transition des SMAKF nécessite deux observations. Les grandeurs observées sont également les grandeurs prédites ; on peut dans ce cas choisir une matrice d'observation unitaire, $\mathbf{C} = \mathbf{I}$. Les matrices de covariance des bruits des SMAKF sont initialisées aléatoirement, toutes avec des valeurs différentes pour les différents filtres qui composent le mélange. Les poids des réseaux neuronaux utilisés en tant que réseau superviseur sont initialisés par des valeurs aléatoires.

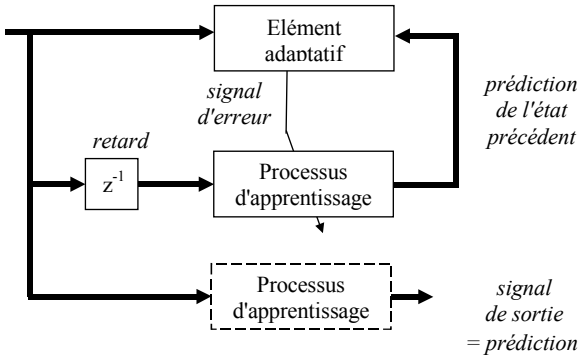


Figure 5-7 : Application de l'AKF en tant que prédicteur.

L'estimation du mouvement a été testée par des séries de simulations, mais également par des expérimentations faites sur la plate-forme robotique. Ensembles, les simulations et les expérimentations permettent d'évaluer très précisément la précision de l'estimation des mouvements dans le contexte de l'asservissement visuel. Les résultats expérimentaux qui sont présentés dans la section suivante ont été obtenus lors d'une manipulation de vision active où il s'agit de centrer une cible dans les deux images d'un capteur de stéréovision.

4.2.3 Résultats expérimentaux

La tâche de vision active qui est considérée est l'asservissement visuel dynamique d'un capteur de stéréovision sur un point se déplaçant dans un espace tridimensionnel. La tête robotique déjà présentée est utilisée en tant que capteur de vision. Les corrections (les consignes de commande) sont produites par une technique de saccades oculaires. Le centre de chaque image comporte une zone d'insensibilité, ce qui se traduit par le fait que la tête n'effectue une correction de sa position angulaire que lorsque la cible sort de cette zone.

Les techniques utilisant des saccades sont généralement associées au controlleurs neuronaux. Les mouvements du capteur ne sont de ce fait pas permanents ; il est alors possible de mesurer à un instant précis, les coordonnées articulaires du robot manipulateur, les caractéristiques visuelles, et les paramètres extrinsèques des caméras qui leur sont associés. Cette opération est nécessaire pour enrichir la base d'apprentissage des réseaux neuronaux avec des exemples parfaitement représentatifs du système robotique.

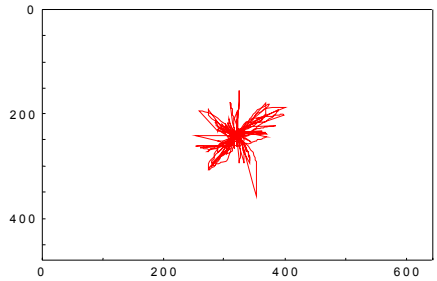
La cible est l'effecteur du bras Sylvia et ses déplacements rapides expliquent les trajectoires excentrées dans les images gauche et droite de la Figure 5-8. Cette même figure montre l'évolution de la trajectoire dans les images au cours du temps, et lui fait correspondre l'évolution des coefficients de la matrice de transition du SMAKF.

Les erreurs de prédictions sont présentées dans le Tableau 5.3. Ce sont des résultats représentatifs d'une vingtaine de manipulations, avec des filtres initialisés de manière différente à chaque fois, pour prédire des trajectoires du même type que celle montrée sur la Figure 5-8. Les performances de l'approche SMAKF et celles du mélange de SMAKF sont comparées aux performances de prédicteurs simples qui servent uniquement de références pour les comparaisons : la moyenne itérative, et les interpolations linéaires d'ordre 0 et 1. L'interpolation linéaire d'ordre 0 consiste à utiliser pour la prédiction la dernière valeur observée, et l'interpolation linéaire d'ordre 1 consiste à utiliser pour la prédiction le prolongement d'une droite définie par les deux dernières observations. Ces prédicteurs ne sont efficaces que sur des signaux spécifiques, comme des signaux aléatoires à variance faible autour de leur valeur moyenne, des signaux monotones, ou des signaux évoluant à vitesse réduite.

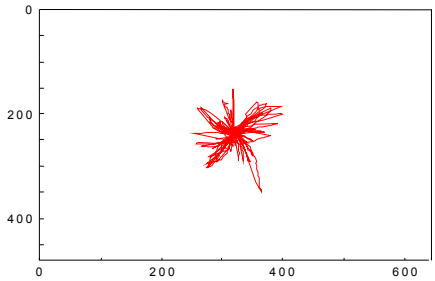
Méthode de prédiction	Coord.	Erreur moyenne absolue		Erreur maximale absolue		Ecart type	
		x	y	x	y	x	y
Moyenne itérative	gauche	12.590	13.061	86.377	112.685	15.345	15.391
	droite	11.711	13.024	88.083	109.465	15.694	15.181
Interpolation linéaire d'ordre 0	gauche	1.240	1.349	224.900	316.330	3.680	4.511
	droite	1.154	1.375	97.160	320.910	2.907	4.586
Interpolation linéaire d'ordre 1	gauche	1.601	1.589	261.020	318.150	5.444	6.145
	droite	1.439	1.628	97.590	322.480	3.926	6.248
SMAKF avec la dernière mesure	gauche	1.037	1.026	165.514	187.348	9.433	11.454
	droite	1.087	1.003	197.873	147.186	9.254	10.726
SMAKF avec la dernière mesure et la dernière vitesse	gauche	0.018	0.045	141.823	143.777	2.039	1.925
	droite	0.050	0.055	143.507	209.614	1.930	2.736
Mélange de $3 \times 2 \times 2 = 12$ SMAKF	gauche	0.011	0.031	74.255	83.510	1.684	1.802
	droite	0.027	0.036	86.248	88.391	1.751	1.770

Tableau 5.3 : Comparaisons des erreurs de prédiction de différents prédicteurs.

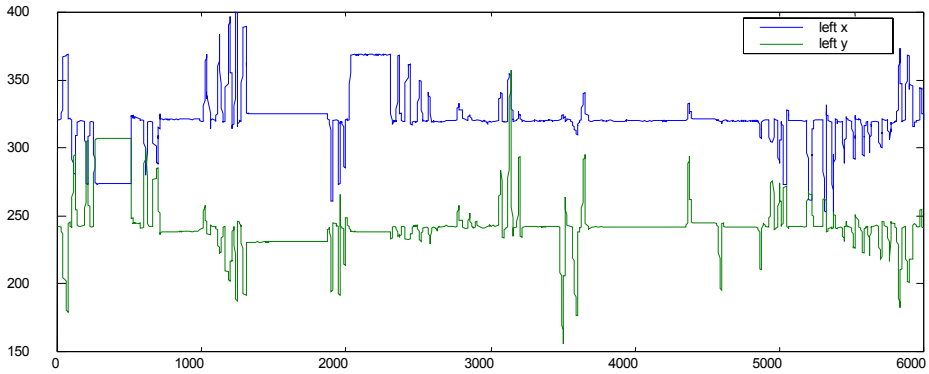
Les erreurs sont des erreurs absolues qui sont définies dans l'espace des caractéristiques visuelles, donc dans les images gauche et droite, et sont de ce fait exprimées en pixels.



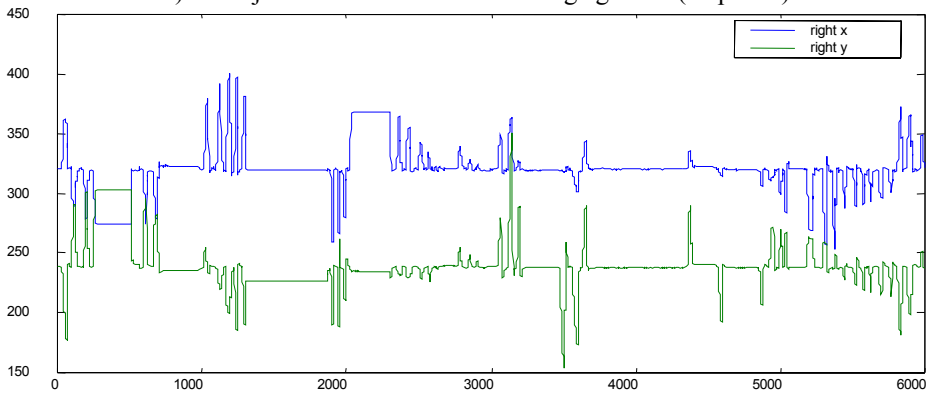
a) La projection de la cible dans l'image gauche



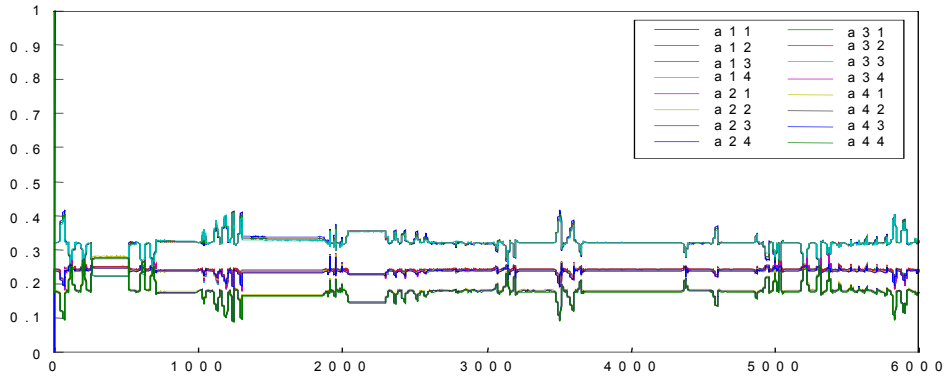
b) La projection de la cible dans l'image droite



c) La trajectoire de la cible dans l'image gauche (en pixels)



d) La trajectoire de la cible dans l'image droite (en pixels)

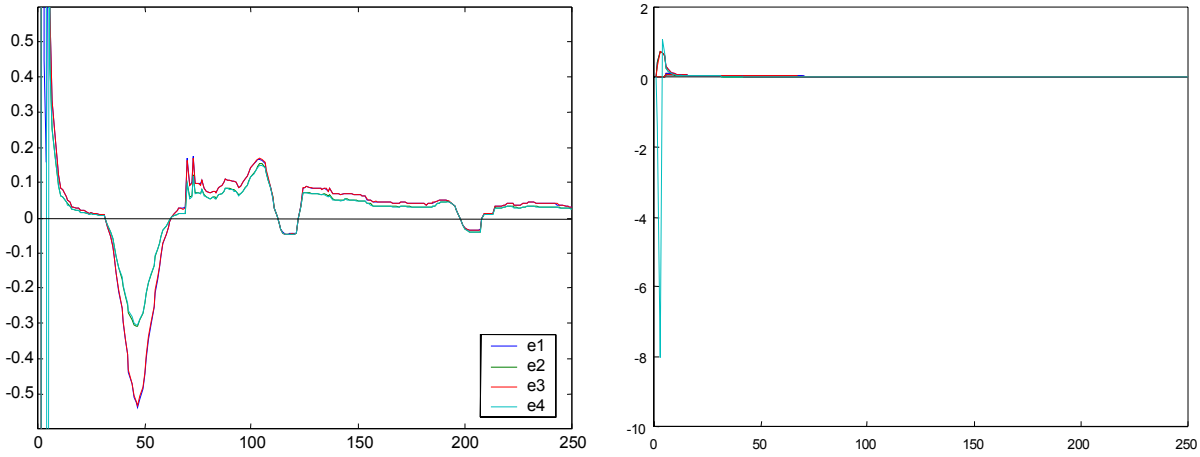


e) Les coefficients de la matrice de transition du SMAKF (sans unité)

Figure 5-8 : Principe général de l'asservissement visuel direct d'un robot.

Du haut vers le bas, sont représentés sur cette figure, la trajectoire du point cible projetée dans les images de gauche et de droite du capteur stéréoscopique, l'évolution du point projeté dans l'image gauche au cours du temps discret, l'évolution du même point projeté dans l'image droite, et enfin le comportement des coefficients de la matrice de transition adaptée par le SMAKF. Il faut remarquer que les coefficients de la matrice de transition sont, d'une certaine manière, corrélés avec les informations visuelles.

Le Tableau 5.3 compare un SMAKF avec uniquement la prise en compte de la dernière observation, un SMAKF qui estime la position et la vitesse de la cible, et un mélange de SMAKF qui estime la position et la vitesse de la cible. On remarque que la prise en compte explicite de la vitesse permet de réduire l'erreur de prédiction, et que le mélange de SMAKF permet d'obtenir les erreurs maximales les plus faibles. La Figure 5-9 montre l'évolution pour les 250 premières itérations de l'erreur de prédiction obtenue avec le SMAKF qui tient compte de la vitesse, ainsi que le comportement du gain de Kalman durant cette même période. Une partie des résultats a déjà fait l'objet d'une présentation (Wira et Urban 2000).



a) Les erreurs de prédiction du SMAKF (en pixels) b) Le gain de Kalman

Figure 5-9 : Comportement du prédicteur SMAKF.

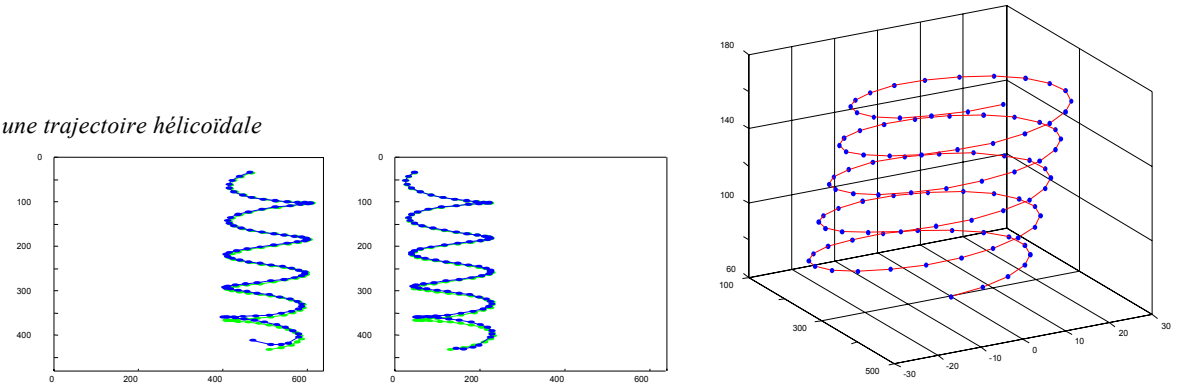
La courbe de gauche représente l'évolution de l'erreur de prédiction d'une trajectoire d'un objet cible dans l'espace des images pour les 250 premières itérations. La courbe de droite, pour le même intervalle, montre l'évolution du gain de Kalman.

Les expérimentations précédentes ont été complétées par d'autres séries de simulations. Plusieurs types bien distincts de trajectoires ont été utilisés, entre autres ceux montrés sur la Figure 5-10. Ces trajectoires comportent notamment des vitesses et des accélérations non constantes. Hormis ces caractéristiques, en considérant la Figure 5-10 on se rend compte qu'il entre une part d'aléatoire dans certains comportements de la cible.

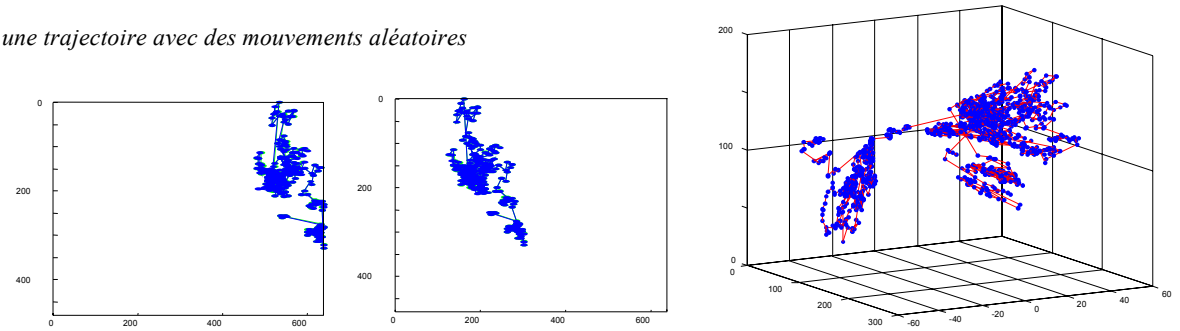
Concernant le SMAKF, la théorie, confirmée par des expérimentations, montre clairement que les propriétés des bruits, ou les paramètres pour les faire évoluer ou pour les estimer, ne sont valables que pour un seul type de comportement de la cible. Par contre, et parce qu'il a été conçu dans cet objectif, seul le mélange de SMAKF est capable d'estimer et de prédire correctement tous les types de trajectoires. Après une phase initiale d'une centaine d'itérations, le mélange de SMAKF est capable de prédire en permanence le mouvement de la cible avec une erreur de l'ordre du pixel, même si son comportement change. L'existence de plusieurs SMAKF (réseaux experts), et le fait que le réseau superviseur associe par apprentissage un SMAKF à un comportement de la cible, autorise le mélange à être capable de tenir compte de la non stationnarité des signaux, donc à prédire correctement les mouvements de la cible.

Nous constatons que lorsqu'il n'y a pas lieu, le réseau superviseur ne fait pas de changement de SMAKF. Ceci à été validé sur une trajectoire droite dans l'espace tridimensionnel (Wira et Urban 2001a).

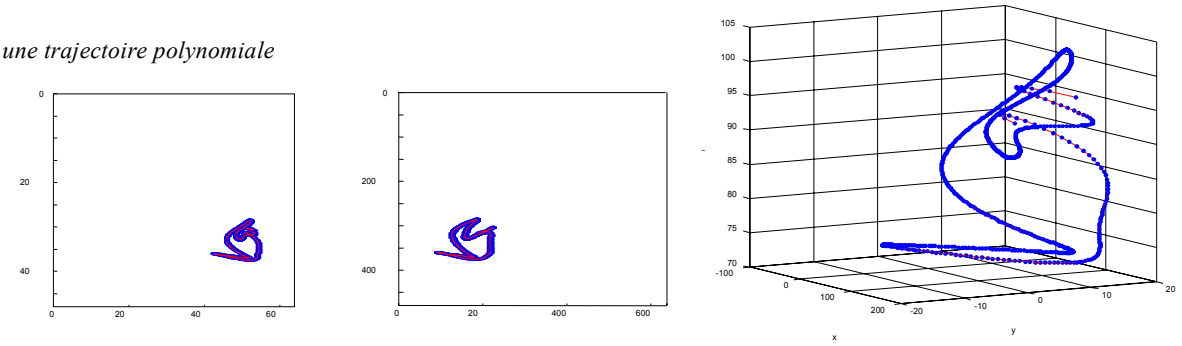
a) une trajectoire hélicoïdale



b) une trajectoire avec des mouvements aléatoires



c) une trajectoire polynomiale



d) une trajectoire linéaire par morceaux

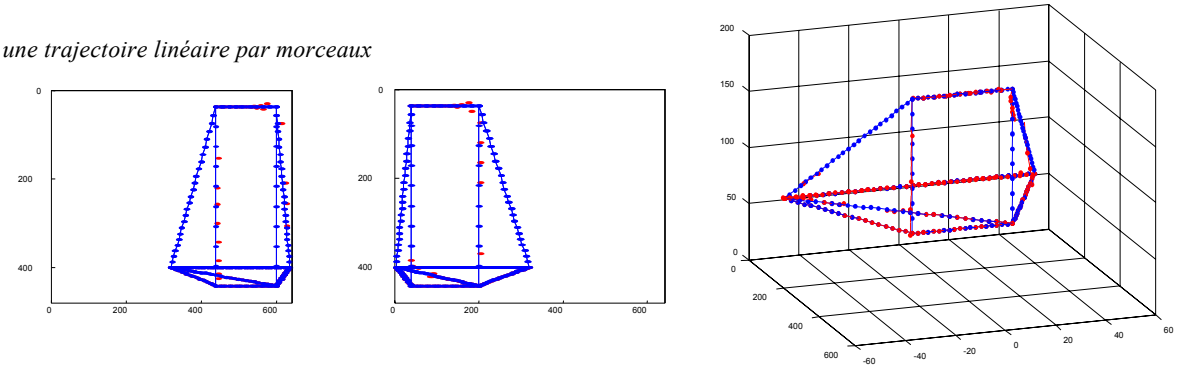


Figure 5-10 : Des exemples de trajectoires prédites.

Trois trajectoires différentes et définies dans l'espace tridimensionnel sont représentées à droite, avec à chaque fois (à gauche) leurs projections dans les plans des images. Les prédictions dans les images sont tracées en pointillées, mais sont à peine visibles puisque l'erreur de prédiction est extrêmement faible (de l'ordre du pixels).

4.3 Contrôle prédictif

4.3.1 Tâches de suivi de cibles par un bras manipulateur

a Objectif

Nous avons dans la section précédente validé les capacités de prédictions de l'approche du mélange de SMAKF dans le contexte particulier de l'estimation du mouvement. Nous allons montrer que cette estimation permet de réaliser un asservissement visuel rapide, robuste, et performant lorsqu'elle est mise au service de la régulation dans les images et lorsqu'elle est appliquée à la poursuite de cible.

La tâche de poursuite de cible par asservissement visuel dynamique est réalisée par un neuro-contrôleur. Un traitement d'images permet d'extraire des primitives visuelles représentatives de la position de la cible et de l'effecteur du robot dans l'espace tridimensionnel. Entre le traitement d'images et le neuro-contrôleur, est placé un prédicteur du type mélange de SMAKF. Le principe est illustré par la Figure 5-11.

Réaliser un apprentissage, une estimation, ou une prédiction dans un environnement non stationnaire n'est pas une tâche aisée, qui plus est lorsqu'apparaissent des changements brusques dans l'environnement. Certains auteurs contournent ces difficultés en proposant des approches basées sur les modèles. Dans (Nagahama et al. 2000) par exemple, un KF est utilisé pour réaliser une prédiction à long terme (plusieurs itérations à l'avance). Cette approche requiert un modèle du capteur de vision, ainsi qu'un modèle de trajectoire (des trajectoires rectilignes ou circulaires sont uniquement utilisées). Les difficultés ne sont que partiellement résolues puisque les événements imprévus ne sont pas pris en compte.

Notre solution pour répondre au problème se veut indépendante des modèles. Le neuro-contrôleur construit lui-même son propre modèle de connaissance de la transformation entre l'espace de perception et l'espace de commande par un processus d'apprentissage. Le mélange de SMAKF construit son propre modèle de connaissance des incertitudes qui persistent, des erreurs d'apprentissage, et des comportements de la cible.

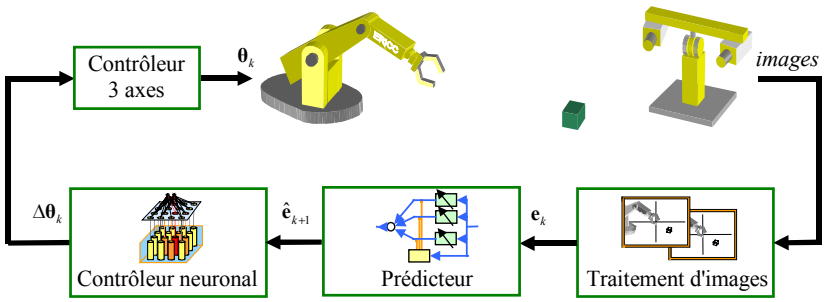


Figure 5-11 : Principe de l'asservissement visuel avec prédiction.
Lorsque les primitives visuelles sont extraites des images, elles sont prédites par le mélange des SMAKF. La régulation se fait dans l'espace image mais avec des grandeurs prédites. L'erreur de positionnement prédite dans les images est convertie en consigne angulaire par un contrôleur neuromimétique de sorte qu'aucune connaissance a priori n'est utilisée par la boucle de retour visuelle.

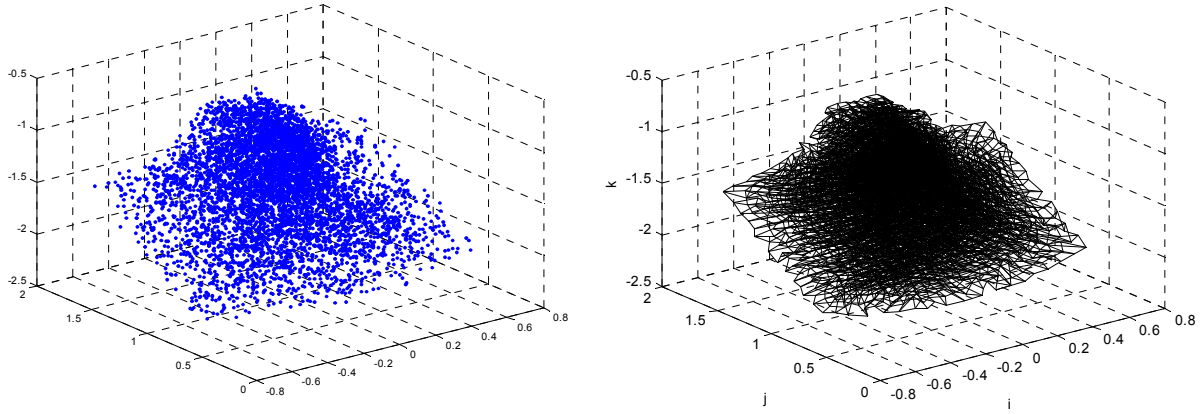
Les processus d'apprentissage, d'adaptation, et de prédiction que comporte la boucle de retour visuelle doivent être effectués en ligne et doivent de ce fait être compatibles avec la contrainte temps réel vidéo pour être parfaitement efficace. Ainsi, à chaque itération, une vue de la scène est acquise, les caractéristiques visuelles sont extraites, elles sont ensuite prédites par le mélange de SMAKF, ce

dernier apprend et s'adapte à partir des nouvelles données, et le neuro-contrôleur calcule une nouvelle consigne puis adapte ses poids pour affiner son modèle de connaissance.

La tâche qui nous intéresse est le suivi d'une cible dans un espace à trois dimensions par un bras manipulateur. Au départ, l'effecteur du robot et l'objet cible partent d'un même point dans l'espace 3-D. Les informations visuelles nécessaires à la boucle de régulation de haut niveau sont fournies par deux caméras fixes. Cette tâche de suivi est réalisée par des simulations ; les paramètres de la scène et les caractéristiques du robot et du capteur de vision sont ceux de la scène décrite précédemment.

b Architecture du contrôleur

Les simulations suivantes utilisent comme contrôleur un NeuroModule. La carte auto-organisatrice est une grille tridimensionnelle comportant $15 \times 15 \times 20$ neurones dans chaque dimension. La taille du réseau neuromimétique a été fixée de telle sorte à ne pas être un facteur limitatif dans la boucle de régulation, pour évaluer pleinement le gain de la prédiction par le mélange de SMAKF. L'espace de sélection du NeuroModule est l'espace articulaire du robot. Les neurones de la grille discrétisent le volume articulaire du bras robotique. A chaque neurone est associé une matrice Jacobienne estimée par un réseau Adaline ayant trois entrées et trois sorties. Des simulations supplémentaires ont montré qu'un NeuroModule effectuant l'opération de sélection dans l'espace des images donne des résultats similaires en terme de précision.



a) La distribution des points à apprendre (radians) b) La carte de Kohonen déployée après apprentissage

Figure 5-12 : Distribution d'entrée et ordonnancement résultant des neurones de la carte de Kohonen.

Les exemples d'apprentissage pour la sélection du NeuroModule sont distribués dans l'espace articulaire du robot (à gauche, en radians). Ils servent de base d'apprentissage pour que la carte topologique de Kohonen puisse réaliser un pavage aussi précis que possible (à droite).

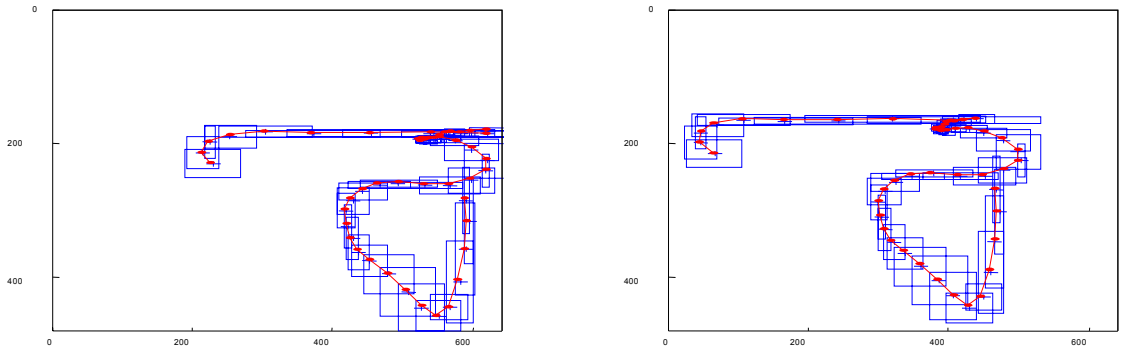
Un apprentissage préliminaire est effectué sur une base d'apprentissage avant l'utilisation du neuro-contrôleur dans la boucle de commande. L'ordonnancement topologique des neurones dans l'espace articulaire est montré sur la Figure 5-12. Les coefficients d'apprentissage et les fonctions de voisinage utilisés sont du même type que ceux préconisés dans le Chapitre 1. Le coefficient d'apprentissage diminue au cours de l'apprentissage et le rayon de voisinage est de plus en plus restreint.

c Estimation du mouvement et traitement d'images

L'étape de prédiction est au service de la boucle de régulation dans le but d'en améliorer les performances. La prédiction des caractéristiques visuelles peut, dans la pratique, contribuer à accroître

la robustesse de la loi de commande. En effet, face à des phénomènes d'occlusion de certains éléments, les informations prédites se révèlent alors indispensables. Les informations prédites peuvent également contribuer à faciliter le traitement d'images.

Pour réduire les temps de traitement des images, seules quelques fenêtres de l'images, appelées des *régions d'intérêts* (et en anglais ROI, "region of interest") sont prises en considération. Ces régions sont déterminées par les anciennes positions des objets suivis. Elles ne sont souvent pas à une position optimale pour permettre la détermination de la nouvelle position d'un objet dans l'image. Cette constatation est d'autant plus vraie lorsque les objets se déplacent rapidement. La taille de ces fenêtres est de plus, soit réglée empiriquement en procédant par essai/erreur, ou est alors fixée suffisamment grande pour ne jamais « perdre » l'objet. Dans ce dernier cas, le temps du traitement de l'image est alors fortement augmenté.



a) Projection dans l'image gauche b) Projection dans l'image droite

Figure 5-13 : Evolution des régions d'intérêt en fonction de la trajectoire.

Les positions des régions d'intérêts sont déterminées par les prédictions (moments d'ordre un) des positions de l'objet cible, et leurs tailles sont inversement proportionnelles aux éléments de la matrice de covariance prédite (moments d'ordre deux) à chaque itération par le SMAKF.

Le filtre de Kalman permet d'apporter une solution par le fait qu'il estime et prédit à la fois le moment d'ordre un et le moment d'ordre deux d'une grandeur. Appliqué à l'estimation du mouvement dans les images, le KF prédit la position future et la covariance associée. La prédiction est donc associée à une indication sur sa propre qualité. Les positions des régions d'intérêt, dès lors sont fixées par les positions prédites des objets, et leurs dimensions sont fixées comme étant inversement proportionnelles aux éléments de la matrice de covariance prédite $\mathbf{P}_{k+1|k}$. Un exemple d'une trajectoire avec les positions des régions d'intérêts avec des tailles différentes est montrée par la Figure 5-13.

4.3.2 Résultats

La boucle complète de régulation dans les images, incluant une étape de prédiction avec un mélange de 40 SMAKF, est évaluée sur une trajectoire simulée. Le début de la trajectoire de la cible est un polynôme quelconque d'ordre 17, référencé par rapport au repère orthonormé associé à la scène robotique. Après environ 500 itérations, il s'opère un changement brusque de comportement, puisque la trajectoire se continue par des petits mouvements aléatoires. La trajectoire complète est représentée dans l'espace de travail du robot sur la Figure 5-14. Du début à la fin, la vitesse et l'accélération de la cible sont non constantes. La vitesse varie entre 0.01 et 1.5 m/s et l'accélération entre 0 et 10.0 m/s^2 dans l'espace tridimensionnel.

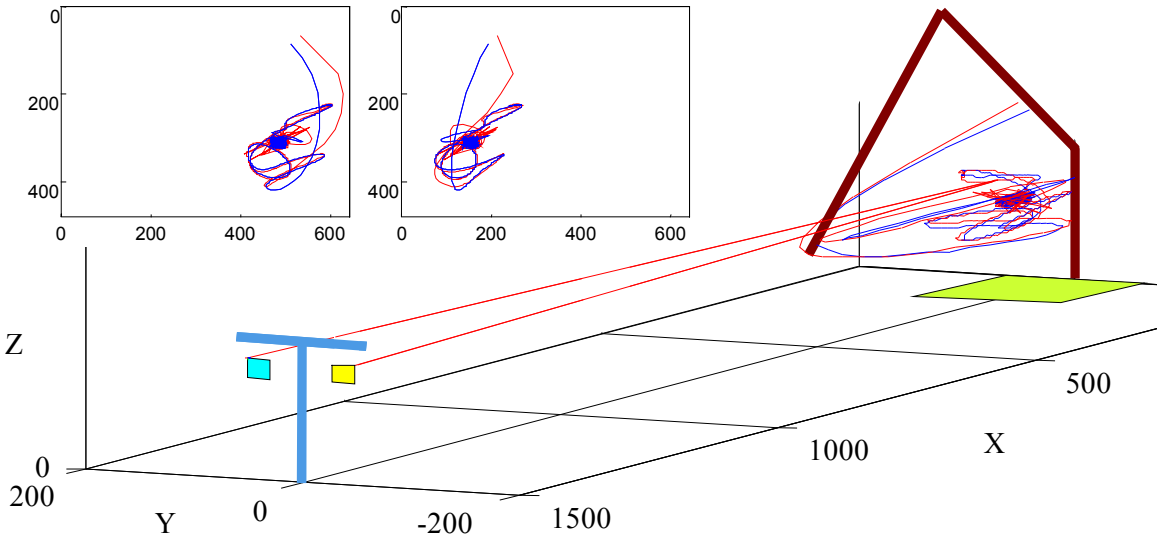


Figure 5-14 : Vue de la scène simulée avec une trajectoire ayant deux types de comportement.

Le début de la trajectoire est un polynôme d'ordre 17 caractérisé par des mouvements amples à vitesse et à accélération non constantes. La fin de la trajectoire est composée par une série de petits mouvements aléatoires dans toutes les directions. La trajectoire de la cible est tracée en bleue et la trajectoire de l'effecteur du bras qui la suit est en rouge. Ces deux tracés apparaissent dans la scène tridimensionnelle, ainsi que dans les projections images.

Différents réseaux superviseurs sont utilisés par le mélange de SMAKF dans le but de montrer le côté déterminant de l'apprentissage. Un mélange de SMAKF avec un réseau superviseur basé sur l'apprentissage softmax est comparé aux mélanges utilisant des réseaux superviseurs aléatoire et à mémoire courte. Les réseaux superviseurs aléatoire et à mémoire courte ont déjà été introduits dans l'apprentissage d'une fonction vectorielle. Les performances de prédiction de ces trois approches du mélange sont comparées à celles d'un SMAKF unique. Ces comparaisons sont présentées dans le Tableau 5.4 ; elles sont basées sur une vingtaine de manipulations, répétées avec des conditions initiales différentes à chaque fois.

Mélange de SMAKF	Erreur sur la coordonnée X			Erreur sur la coordonnée Y		
	Moyenne	Écart Type	Max.	Moyenne	Écart Type	Max.
SMAKF unique	0.5162	9.431	178.481	0.432	7.532	161.599
Avec un superviseur aléatoire	31.270	70.296	148.361	23.35	52.269	195.607
Avec un superviseur à mémoire courte	1.025	11.824	144.459	1.022	9.726	137.948
Avec un superviseur avec apprentissage	0.375	2.716	22.110	0.2938	2.632	13.057

Tableau 5.4 : Erreur absolue de prédiction par le mélange de SMAKF.

Les erreurs absolues de prédiction présentées dans ce tableau correspondent à celles d'un mélange d'experts avec un réseau superviseur doté de l'apprentissage softmax, d'un réseau superviseur à mémoire courte, et d'un réseau superviseur aléatoire. Les erreurs sont exprimées dans l'espace des caractéristiques visuelles (images gauche et droite, en pixels) et sont comparées à l'erreur de prédiction d'un SMAKF unique.

L'analyse des résultats montre que le mélange de SMAKF est incapable de prédire correctement les mouvements avec un réseau superviseur aléatoire et rend de ce fait la boucle d'asservissement très peu performante. Le mélange doté d'un réseau superviseur à mémoire courte est capable d'estimer les mouvements de la cible uniquement lorsque ces mouvements ne sont pas très rapides et pas très variés. Il opère très peu de changement de réseaux experts, et lorsque ce changement a lieu, il s'effectue toujours avec du retard. Ce type de réseau superviseur n'est pas capable d'anticiper ni d'associer correctement un réseau expert à un comportement de la cible.

Le SMAKF seul est capable d'estimer les mouvements de la cible et de l'effecteur avec une précision relativement correcte sur le début de la trajectoire. Sa principale limitation survient lors du changement brusque de comportement. Les estimations de ses matrices de covariances $\mathbf{Q}_k$ et $\mathbf{R}_k$ sont spécifiques à un type de trajectoire. Après un certain temps, elles sont inadaptées au nouveau comportement de la cible et induisent des erreurs d'estimations importantes (surtout en valeur maximale).

Le mélange de SMAKF avec un réseau superviseur du type softmax donne les meilleurs estimations des mouvements de la cible. Il permet surtout d'avoir des valeurs maximales qui ne dépassent jamais une vingtaine de pixels. Comme il a été conçu pour faire face à différents types d'incertitude, le changement brusque dans la trajectoire de la cible est pris en compte grâce aux différentes réalisations des réseaux d'experts. Le comportement du mélange est détaillé à travers la Figure 5-15. Cette figure montre l'évolution de la trajectoire de la cible et celle suivie par l'effecteur du robot asservi (a). Cette figure montre également le comportement des poids de sortie g_i du réseau superviseur (b). Ces coefficients de pondération témoignent de l'importance d'un réseau expert à chaque instant, et les courbes (a et b) montrent comment les experts sont répartis sur les différentes parties de la trajectoire. Les capacités d'apprentissage et de mémorisation du réseau superviseur font qu'à chaque type de comportement est associé un réseau expert. Lorsque le comportement de la cible change, le réseau expert choisi comme étant le plus apte à prédire change également. Lorsque la cible répète un comportement, le même réseau expert qui a déjà appris ce comportement est à nouveau sollicité. Ce principe permet à chaque réseau expert de se spécialiser uniquement sur une portion de l'espace d'entrée. Cette spécialisation est aussi mise en œuvre par la capacité d'adaptation de chaque expert, qui est renforcée lorsqu'il est dominant. L'adaptation des experts est montrée sur la Figure 5-15 avec les paramètres $\mathbf{Q}_{i,k}$ (d) et $\mathbf{R}_{i,k}$ (e). Seuls, les experts ne sont représentatifs que d'une partie de la tâche ; ensembles, ils sont capables de parfaitement apprendre et reproduire toute la tâche apprise.

Le mélange de SMAKF utilisé comporte 40 réseaux experts, mais tous ne sont pas utilisés. Seule une vingtaine participe aux prédictions des mouvements de la cible sur l'ensemble de la trajectoire proposée. Il est difficile de déterminer au départ le nombre de réseaux experts nécessaires pour résoudre un problème. Ceci est inhérent à l'approche ME. Si un nombre insuffisant de réseaux experts peut limiter les performances du mélange, en revanche, ajouter plus d'experts que nécessaire ne les améliore pas. Nous procédons par essai/erreur, de sorte qu'il y a toujours plus de réseaux d'experts disponibles que de types de comportements. Ceux qui ne sont pas utilisés, sont disponibles pour d'autres types de comportements.

Concernant l'asservissement lui-même, les performances sont évaluées grâce à l'erreur absolue de traînage dans l'espace tridimensionnel. Comme les deux trajectoires débutent à la même position, cette erreur est l'écart moyen (sans signe) qui persiste entre la position de la cible et celle de l'effecteur du bras au cours de la trajectoire. Par rapport à un asservissement identique, avec les mêmes paramètres et le même neuro-contrôleur, les prédictions du mélange de SMAKF permettent de diviser l'erreur de traînage par un facteur 3. Avec l'estimation des mouvements, l'erreur de traînage est évaluée à 9.15 mm (max. 46.25) dans l'espace 3-D en norme. Elle est de 8.58 mm (max. 44.96) selon l'axe des x, de 1.65 mm (max. 20.62) selon l'axe des y et de 1.59 mm (max. 17.13) selon l'axe des z. Ces résultats sont présentés dans (Wira et Urban 2001b) et dans (Wira et al. 2001).

a) Les coordonnées tridimensionnelles de la cible et de l'effecteur (en mm)

b) Les valeurs des coefficients de pondérations g_i (sans unité)

c) Le coefficient d'apprentissage adaptatif (sans unité)

d) Les coefficients diagonaux de la matrice $\mathbf{Q}_{i,k}$

e) Les coefficients diagonaux de la matrice $\mathbf{R}_{i,k}$

f) L'erreur de prédiction dans les images (en pixels)

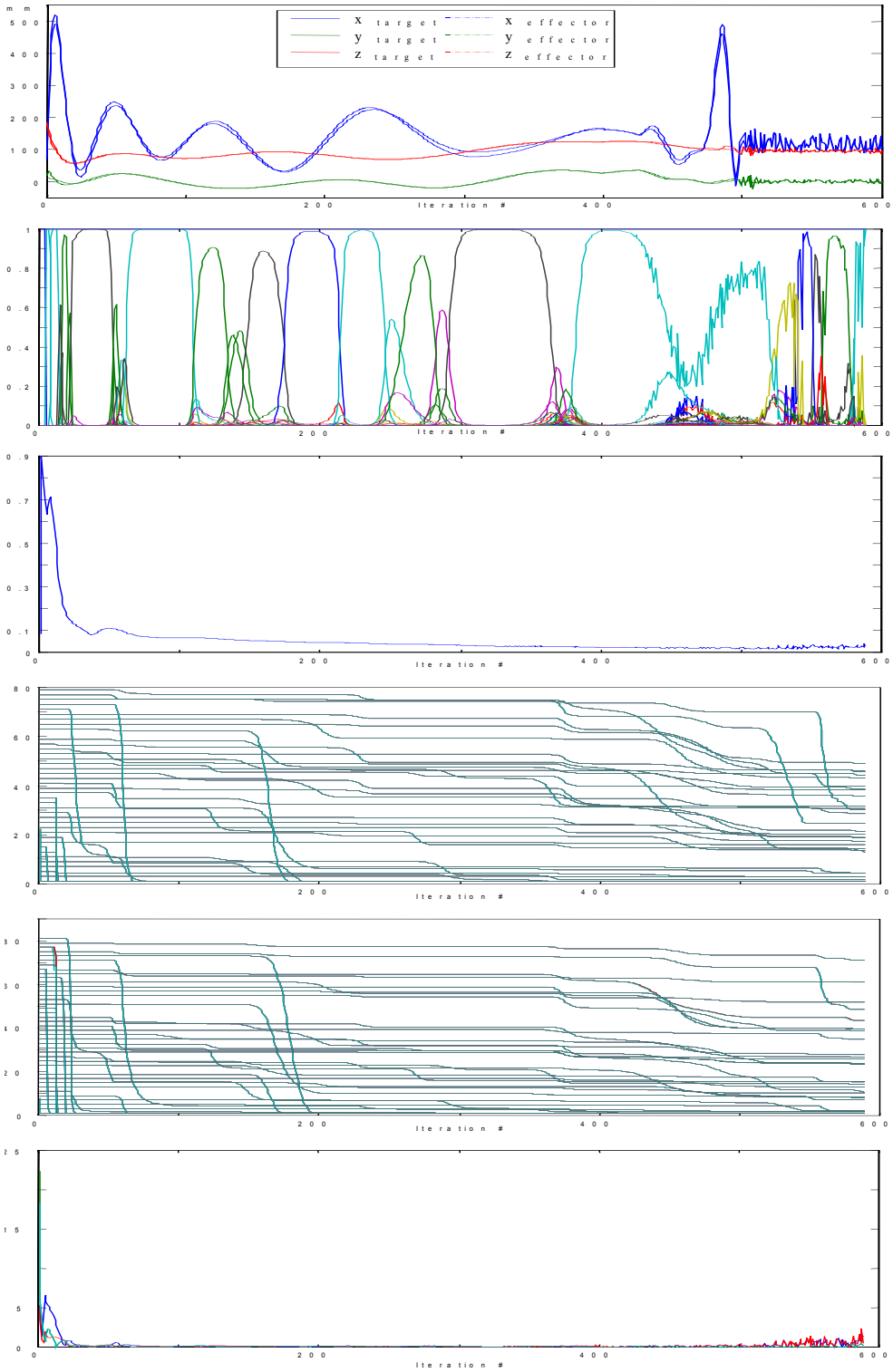
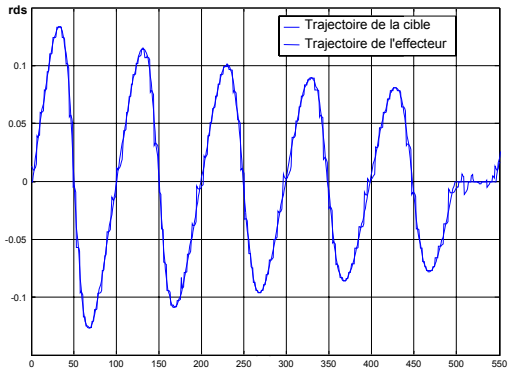
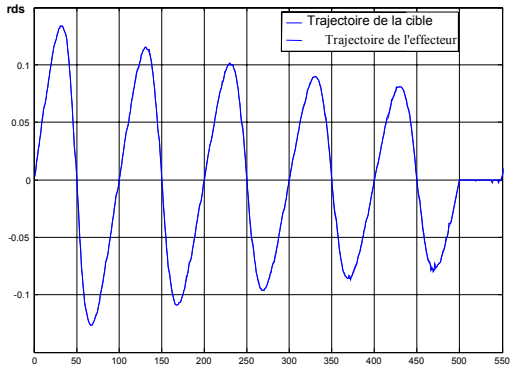


Figure 5-15 : Illustration du comportement du mélange de SMAKF.

De haut en bas, ces courbes représentent l'évolution dans le temps des coordonnées tridimensionnelles de la cible et de l'effecteur du robot qui la suit, des coefficients de pondération délivrés par le réseau superviseur pour chaque réseau expert, du coefficient d'apprentissage adaptatif, des coefficients diagonaux des matrices $\mathbf{Q}_{i,k}$ *et* $\mathbf{R}_{i,k}$*, et de l'erreur de prédiction dans les images (en pixels). Il est intéressant de remarquer de quelle manière le coefficient d'apprentissage et les éléments des matrices* $\mathbf{Q}_{i,k}$ *et* $\mathbf{R}_{i,k}$ *sont influencés par les pondérations* g_i *; et comment ces derniers sont influencés par la trajectoire de la cible.*



a) La consigne angulaire θ_1 (radians) sans estimation



b) La consigne angulaire θ_1 (radians) avec estimation

Figure 5-16 : Les trajectoires de la cible et celle suivie par l'effecteur avec et sans prédiction.

La courbe à gauche montre les trajectoires de la cible et de l'effecteur dans l'espace articulaire (en radians) sans estimation du mouvement, et celle à droite avec estimation du mouvement.

D'autres simulations, avec des trajectoires différentes montrent que cette erreur peut être diminuée d'un facteur variant entre 2.5 et 3.5. La trajectoire hélicoïdale déjà présentée a également servit de support de simulation. Elle est constituée de plusieurs comportements, puisqu'elle évolue du bas vers le haut selon une hélice inclinée, et revient selon une ligne droite après cinq révolutions sur son point initial. Une dizaine de SMAKF suffise pour estimer les mouvements de la cible pour cette trajectoire particulière. Les résultats sont du même gabarit que les résultats précédents. Ils peuvent également être interprétés dans l'espace articulaire, comme montré sur la Figure 5-16 et le Tableau 5.5. Les résultats du Tableau 5.5 découlent d'une simulation répétée 100 fois, avec des initialisations différentes de tous les filtres à chaque fois.

Le prédicteur le plus efficace pour ce type de trajectoire serait basé sur un modèle plus judicieux : un polynôme dont les coefficients seraient estimés en ligne (par une méthode des moindres carrés par exemple). Il a été montré que ceci n'est efficace qu'avec des trajectoires extrêmement simples, telles que des droites, ou des trajectoires très lentes. L'avantage du mélange de SMAKF est d'être capable d'estimer toutes sortes de trajectoires, mais aussi de pouvoir effectuer un changement de modèle (filtre) lorsque cela s'avère nécessaire, de mesurer ses propres performances, et de les améliorer par la suite grâce à un processus d'apprentissage.

Axe	Sans prédiction				Avec prédiction par le mélange de AKF			
	Moyenne	Écart Type	Max.	Relative (%)	Moyenne	Écart Type	Max.	Relative (%)
θ_1	1.11	2.57	96.0	0.07	0.74	1.30	22.0	0.04
θ_2	0.86	1.73	29.0	0.04	0.64	1.17	15.0	0.03
θ_3	1.07	1.92	48.0	0.04	0.82	1.14	17.0	0.03

Tableau 5.5 : Comparaison de l'erreur absolue de traînage dans l'espace articulaire (10^{-3} radians).

L'erreur est ici définie dans l'espace articulaire du robot comme la différence (sans signe) entre la trajectoire idéale et théorique de la cible, et celle effectivement suivie par l'effecteur du bras.

Quelle que soit la trajectoire, les performances de l'estimation des mouvements sont sensiblement les mêmes que ceux obtenus dans cette partie. Ceci a été vérifié par de nombreures simulations et à permit de conclure que l'approche est générique.

L'approche du mélange de SMAKF a été expérimentée et a été validée avec un contrôleur neuromimétique différent, une extension du CMAC (Kara et al. 2000a; Kara et al. 2000b).

Des simulations supplémentaires ont par ailleurs été conduites avec une loi de commande basée sur une matrice d'interaction calculée explicitement à partir des modèles. Là encore, on peut constater de la généricité de l'estimateur, et les performances sont améliorées d'un même facteur, avec et sans estimation des mouvements.

5 Remarques de conclusion

Nous avons évalué dans ce chapitre un schéma d'asservissement visuel utilisant l'approche fonction de tâche. Ce schéma est spécifié en terme de régulation dans l'image et utilise une loi de commande adaptative et prédictive. L'objectif de cette boucle d'asservissement visuel est de poursuivre un objet mobile possédant un mouvement inconnu.

Nous avons d'abord présenté un logiciel de simulation des robots, des capteurs de vision, et d'une scène dans sa globalité, permettant de simuler des tâches robotiques et d'évaluer des lois de contrôle et des algorithmes de commande.

Nous avons adopté une approche adaptative de l'asservissement visuel. Utiliser un réseau de neurones artificiels permet d'apprendre la transformation de l'espace image vers l'espace articulaire du robot à commander. L'estimation obtenue est un modèle de connaissance de la transformation sensori-motrice, et permet de ce fait de convertir une erreur mesurée dans les images en consignes robotiques. Nous utilisons dans nos applications une extension à sortie supervisée d'une carte auto-organisatrice de Kohonen.

Le schéma d'estimation et de prédiction du mouvement de la cible dans l'image a été intégré dans la loi de commande. L'asservissement devient donc une boucle de contrôle adaptative et prédictive. Les simulations tout comme les expérimentations, montrent que les SMAKF organisés selon une approche de type ME permettent de détecter et de compenser les mouvements de la cible, aussi bien que les changements brusques, que les mouvements aléatoires, que les changements de comportement, ou encore que la non stationnarité.

L'asservissement visuel prédictif utilisant le mélange de SMAKF, comparé à des asservissements utilisant d'autres techniques de prédiction comme l'EKF, donne des résultats légèrement meilleurs en règle générale. Ceci est d'autant plus encourageant que le mélange de SMAKF n'utilise pas de connaissance préalable. Très peu de méthodes peuvent être employées lorsque les mouvements de l'objet à suivre sont non stationnaires. Le mélange de SMAKF est alors le seul capable de prédire et de compenser les mouvements de la cible dans la loi de commande.

Plus la vitesse est grande, plus l'asservissement en poursuite est mis à l'épreuve. Les comparaisons que nous proposons sont effectuées à vitesses égales et à trajectoires identiques, pour la même loi de commande. La reproduction d'une configuration est parfaitement possible en simulation.

L'approche de contrôle prédictif par le SMAKF a été validée avec d'autres contrôleurs neuromimétiques, tel que le CMAC (Kara et al. 2000a; Kara et al. 2000b). Une synthèse des travaux présentés dans ce chapitre est disponible dans (Wira et Urban 2001a).

La généricité de la méthode d'estimation proposées a non seulement été montrée avec de nombreuses applications robotiques, mais a également été confirmée par de nombreuses simulations sur des systèmes différents, linéaires et non linéaires, voire même sur des séries temporelles comme la série de Hénon, celle de van de Pol, ou encore les variations temporelles de la luminosité du soleil.

Conclusion

1 Contributions principales

Nous avons abordé dans cette thèse les aspects de suivi et d'estimation du mouvement d'un objet. Plus précisément, nous avons développé une méthode permettant l'estimation du mouvement dans les images d'un objet mobile pour l'asservissement visuel d'un système robotique.

La première étape de ce travail introduit des outils de modélisation des systèmes mécaniques articulés qui permettent d'en approfondir la connaissance, et présente les différentes approches d'asservissement visuel. Une vue générale sur les réseaux de neurones, leurs architectures, les paradigmes d'apprentissage et la description d'un réseau particulier, les cartes auto-organisatrices de Kohonen, montre qu'ils constituent une approche intéressante pour la réalisation d'une loi de commande. L'apprentissage et le connexionisme sont ensuite présentés sous l'angle des statistiques. Cette étude démontre que la combinaison des deux présente de sérieux avantages dans bon nombre d'applications modernes. Les théories statistiques font maintenant partie intégrante de l'étude des réseaux neuronaux, elles permettent de faire d'un apprentissage autre chose qu'un art approximatif. La combinaison des deux peut être optimisée grâce à une approche modulaire de l'apprentissage.

Nous distinguons deux formes de modularité dans la conception des architectures neuronales : la modularité à modèle multiple et la modularité vraie. La première utilise plusieurs modules pour traiter un même problème. Les sorties de tous les modules sont combinées pour calculer la sortie globale. La seconde est une composition de plusieurs modules qui apprennent séparément tous les sous-problèmes d'un problème complexe.

Nous présentons plus en détails l'approche modulaire des ME. Elle sert de modèle pour la conception d'un estimateur à modèle multiple basé sur des filtres de Kalman adaptatifs. Le principe nouveau de ce filtre adaptatif réside dans l'ajustement de sa représentation d'état au cours du temps. Cette adaptation utilise les dernières observations et permet de se passer de toute connaissance a priori relative au système dont on souhaite estimer l'état interne. Un module de supervision, un réseau de neurones, associe et estime la probabilité pour chaque filtre que celui-ci produise l'estimation optimale pour une entrée donnée. Ces probabilités permettent d'assigner les filtres à des zones de l'espace des entrées. L'adaptation des filtres est renforcée dans leur zone et est bloquée ailleurs, de façon à produire un spécialiste par zone. Les deux approches (SMAKF et ME) se complètent efficacement, notamment pour compenser les propriétés non stationnaires de certains environnements.

Appliqué à l'asservissement visuel et à l'estimation du mouvement d'un objet, ce principe permet de caractériser des types de comportement de l'objet, et d'y associer à chaque fois un filtre spécialisé pour réaliser une estimation meilleure. Le mouvement de l'objet est estimé dans l'image à partir de la mesure de sa position, de sa vitesse, et de son accélération. L'algorithme d'estimation permet de compenser dans la loi de commande l'éventuel caractère non stationnaire du comportement de la cible. Il en résulte une commande précise et robuste malgré une connaissance analytique minimale du système robot-vision et malgré certains événements imprévus dans la scène.

Nous avons présenté un schéma d'asservissement visuel par apprentissage qui spécifie la tâche de vision en terme de régulation dans l'image. Nous avons proposé une loi de commande adaptative et prédictive basée sur cette approche qui permet de poursuivre un objet possédant un mouvement

inconnu. Pour ce faire, un schéma robuste d'estimation et de prédiction du mouvement de la cible dans l'image a été présenté et introduit dans la loi de commande. Les résultats expérimentaux soulignent que l'emploi de plusieurs filtres de Kalman combinés pour estimer le mouvement de la cible, permet de détecter et de compenser des changements brusques du mouvement de la cible.

Il ressort des expérimentations réalisées, l'importance de l'approche adaptative retenue du filtre de Kalman. Par conséquent, les résultats se trouvent améliorés même lorsqu'on ne dispose d'aucun a priori sur le type de mouvement de l'objet poursuivi. Le mélange de SMAKF se prête bien aux tâches d'asservissement visuel ou de vision active. Ces dernières se caractérisent par des mouvements saccadés.

2 Perspectives

Nos perspectives en asservissement visuel concernent de manière générale la prise en compte d'objets plus complexes, afin d'en accroître les domaines d'application et la facilité de mise en œuvre des techniques d'asservissement visuel. Se passer des amers visuels placés sur les éléments de la scène, comme l'indique d'autres travaux (Marchand et al. 1999), permet de tenir compte implicitement de toute évolution et tout changement inopinés, même de prendre en compte dans la loi de commande des objets non prévus par avance.

D'un point de vue plus théorique, des études méritent également d'être menées sur la stabilité du SMAKF. Ce filtre est robuste dans le sens où le nombre de divergences n'est pas supérieur à celui d'un filtre classique de Kalman.

Sous la forme introduite dans ce travail, le mélange comprend un nombre fixe de SMAKF prédéterminé par essai/erreur. Son architecture et son apprentissage permettent d'envisager des extensions évolutives. L'objectif qui se dessine est alors de ne plus travailler avec un mélange volontairement surdimensionné qui alourdirait inutilement les calculateurs. On peut penser utiliser le réseau évolutif MILSYS (Rouzier 1998) avec des SMAKF, ou une version évolutive de l'approche des HME (Waterhouse 1997).

Une suite intéressante à ce travail serait aussi d'utiliser l'approche du SOKF, qui réalise un apprentissage de la représentation d'état, tout en estimant l'état interne du système pour des applications concrètes telles que celles de la robotique.

Les expérimentations qui ont été conduites et les simulations qui ont été menées, l'ont toutes été avec un système de vision fixe. Une évolution naturelle est l'introduction d'une stratégie de vision active dans la tâche de suivi de cible par asservissement visuel d'un bras. Un travail dans ce sens a déjà été commencé dans (Urban et al. 1997). L'association de la vision active à l'asservissement visuel permettrait d'évaluer les performances du mélange de SMAKF en présence de degrés de liberté supplémentaires. Les paramètres non constants du capteurs de vision ont une influence directe sur la représentation des caractéristiques visuelles.

Commander des axes supplémentaires nécessite une capacité de mémorisation accrue de la part des réseaux neuromimétiques qui réalisent la ou les lois de commande. Une dernière perspective serait donc d'arriver à réduire la taille des réseaux de neurones. Les tailles requises restent un souci permanent de l'asservissement visuel par apprentissage. La modularité semble être une piste prometteuse (Buessler et al. 1999). Nos connaissances dans ce sens nous font pencher plus vers une modularité vraie que vers qu'une modularité à modèle multiple.

Annexe

Une vue générale du connexionisme

1 Introduction

A la base de l'étude des réseaux de neurones artificiels, il y a la curiosité et la volonté de la part de l'homme de comprendre et d'imiter le fonctionnement du cerveau. A cette fin, une caractérisation biologique très simplifiée du cerveau a été formalisée avec la description de ses éléments constitutifs de base (les neurones) et avec son architecture (la façon dont les neurones sont reliés entre eux). Si l'on sait décrire ainsi les éléments de base et la structure du cerveau, on pourra les modéliser, et le fonctionnement d'un tel modèle présentera alors des propriétés identiques à celles du cerveau.

Si reproduire le fonctionnement du cerveau restait l'unique objectif des réseaux de neurones artificiels ("Artificial Neural Networks", ANN), il serait très loin d'être atteint et bon nombre de chercheurs le qualifierait d'utopique. Le cerveau est un système infiniment complexe.

A ce jour, les réseaux neuronaux continuent à être développés et sont utilisés principalement en tant qu'outils pour l'ingénieur et comme modèles biologiques plausibles. Leurs objectifs sont donc distincts : les premiers servent en tant que solutions alternatives là où les méthodes traditionnelles sont vouées à l'échec, alors que les seconds visent à comprendre, ou à faire comprendre, le fonctionnement des systèmes nerveux. Si les objectifs ne sont pas les mêmes, on peut aisément comprendre que les moyens et les contraintes peuvent également diverger. Nous allons, tout au cours de cette étude, plus considérer les premiers et n'utiliser les seconds qu'en tant qu'éventuelle source d'inspiration.

Quoiqu'il en soit, les réseaux neuromimétiques sont des modèles mathématiques et informatiques. Ils sont des assemblages d'unités de calculs appelés neurones formels dont l'inspiration originelle était un modèle extrêmement simplifié de la cellule nerveuse humaine décrit par McCulloch et Pitts en 1943. Le neurone formel effectue une sommation pondérée des influx qui lui parviennent. Si cette somme dépasse un seuil, le neurone s'active et transmet aux autres neurones un influx égal à son activation. Sinon, il reste inactif et ne transmet rien.

En 1949, Hebb a complété le modèle du neurone en proposant un modèle simple des phénomènes d'apprentissage dans le cerveau, basé sur le renforcement ou l'affaiblissement des efficacités synaptiques, c'est-à-dire des taux de transmission de l'influx entre deux neurones. Le postulat de Hebb veut que l'activité des synapses soit modifiée par leur propre fonctionnement.

Le Perceptron, proposé par Rosenblatt en 1962, est le premier modèle de réseau intégrant le neurone de Mc Cullogh et Pitts et la règle d'apprentissage de Hebb. Les capacités d'apprentissage du Perceptron susciteront de grands espoirs, mais c'est surtout la renaissance des ANN au cours des années 1980 qui ravivera les espérances des chercheurs pour ces algorithmes. Parmi d'autres, la théorie

des approximateurs universels (Rumelhart et McClelland 1986) et la carte auto-organisatrice de Kohonen (Kohonen 1982b), ont été des plus prometteuses.

2 Modèles connexionnistes : paradigmes et architectures

Depuis une vingtaine d'années, beaucoup de réseaux nouveaux, ou des variantes élaborées à partir des réseaux historiques et des réseaux de bases ont été mis au point. Tous, se définissent principalement par leur architecture et leur mode d'apprentissage.

Nous allons, dans un premier temps, définir un processus d'apprentissage et notifier la nuance que nous voyons entre un processus d'apprentissage et un processus adaptatif. Nous tenterons également de dresser une situation parcimonieuse des principales architectures des réseaux.

2.1 Apprentissage

La tâche assignée à un réseau neuromimétique est l'élaboration, par apprentissage, d'un modèle de connaissance de l'environnement dans lequel il opère. Le terme connaissance désigne au sens large, une information mémorisée, utilisée par un processus d'apprentissage pour interpréter, prédire et répondre aux stimulations de l'environnement.

En conséquence de quoi, l'apprentissage revient à changer les paramètres internes d'un réseaux de neurones pour augmenter et améliorer ses performances vis-à-vis de la tâche à résoudre. Apprendre signifie aussi mémoriser.

Nous allons d'abord définir l'apprentissage dans le cadre des ANN et, après un tour d'horizon des différents paradigmes existants, nous étudierons les propriétés d'un processus d'apprentissage.

2.1.1 Définition d'un processus d'apprentissage

L'apprentissage est le processus de détermination des poids synaptiques d'un réseau neuromimétique. En clair, il désigne un processus d'adaptation des paramètres internes (les connexions) d'un réseau de neurones formels avec comme finalité l'obtention d'un état stable. Cet état stable correspond alors à un modèle de connaissance dans un environnement dont une base d'exemples fournit une description. Toute la connaissance d'un réseau est représentée – mémorisée – par les poids synaptiques.

D'un point de vue pratique, l'apprentissage d'un réseau consiste à modifier ses poids de manière à optimiser un critère de performance (par exemple l'erreur quadratique sur la base d'apprentissage). Il peut être vu comme la recherche d'une solution (la meilleure) dans un espace de dimension le nombre de poids du réseau. Cette recherche ne peut aboutir à une solution satisfaisante que si la complexité (le nombre de paramètres) est suffisante pour apprendre une tâche. Si cette complexité est trop faible, le réseau n'a pas assez de flexibilité pour modéliser correctement les données de la base d'exemples. Un réseau trop complexe a tendance à entraîner un phénomène de sur-apprentissage. Dans les deux cas, les performances de généralisation du réseau sont amoindries.

L'apprentissage neuromimétique nécessite en général une grande quantité de données, que l'on regroupe dans un corpus d'apprentissage, plus couramment appelée base d'apprentissage. Selon le type d'apprentissage, la phase de restitution se fait soit sur une portion de la base d'apprentissage, soit sur une nouvelle base. Ces bases supplémentaires sont appelées corpus de test et corpus de généralisation.

Idéalement, le réseau doit acquérir davantage de connaissance à chaque itération des trois étapes suivantes du mécanisme de l'apprentissage :

- le réseau est stimulé par l'environnement ;
- en réponse à cette stimulation, et selon des règles précises, le réseau adapte et modifie ses paramètres latents dans l'unique but d'améliorer ses performances ;
- en conséquence, et suite aux changements intervenus dans sa structure interne, le réseau répond différemment à l'environnement.

Concrètement, les règles d'apprentissage sont destinées à chercher une solution au problème de l'apprentissage que constitue l'assimilation de la connaissance à partir d'une base d'exemples. Ces règles sont liées à l'architecture du réseau, mais sont plus ou moins efficaces selon l'application qui est considérée. Dans (Haykin 1994) une liste des règles et des principes d'apprentissage les plus courants est dressée ; elle comporte notamment l'apprentissage par correction d'erreur ("error-correction learning"), l'apprentissage basé sur la mémorisation ("memory-based learning"), l'apprentissage de Hebb ("Hebbian learning"), l'apprentissage compétitif ("competitive learning") et l'apprentissage de Boltzmann ("Boltzmann learning").

Il n'existe pas d'unique processus d'apprentissage. Les apprentissages diffèrent principalement selon leur manière d'ajuster les poids synaptiques d'apprentissage – on parle alors de règle ou loi d'apprentissage – mais aussi selon leur façon d'interagir avec leur environnement. Ce dernier point fait référence à des paradigmes d'apprentissage.

2.1.2 Paradigmes d'apprentissage

Les paradigmes d'apprentissage des réseaux neuromimétiques opèrent selon le contexte, la disponibilité des informations et les contraintes induites par la tâche à résoudre. Ainsi, les informations disponibles pour entraîner un réseau ne sont pas de même nature lorsqu'il s'agit d'approximer une fonction continue, de séparer et distinguer des classes, de prédire une série temporelle ou de contrôler un robot.

La disponibilité de l'information relative à l'environnement permet de séparer trois paradigmes différents, l'apprentissage supervisé, l'apprentissage par assignation de crédit et l'apprentissage non supervisé.

a L'apprentissage supervisé

L'apprentissage supervisé, en anglais "Supervised Learning", doit disposer d'un comportement de référence précis pour pouvoir l'inculquer au réseau neuronal. Le réseau doit mesurer la différence entre son propre comportement et le comportement de référence, il doit alors corriger ses poids de façon à réduire cette erreur (la rétropropagation du gradient est un exemple).

Sur la Figure 6-1, le système à apprendre constitue un *éducateur* pour le réseau. Le réseau prend comme stimulus le même vecteur des variables explicatives que le système. Ce dernier, en réponse à l'entrée présentée, fournit une sortie désirée qui représente pour le réseau le comportement de référence.

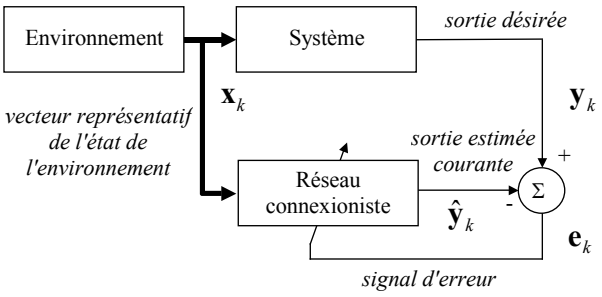


Figure 6-1 : Principe de l'apprentissage supervisé.

Dans ce mode d'apprentissage, les réponses désirées fournies par l'éducateur décrivent la fonction du réseau. Le réseau doit remplir la même fonction que le système. L'apprentissage doit faire

en sorte que la fonction du système puisse être reproduite seulement à partir du corpus d'apprentissage, c'est-à-dire des stimuli et des sorties désirées.

L'ajustement des poids d'un réseau Adaline suit un schéma supervisé. Une des règles les plus utilisées est la rétropropagation du gradient.

Pour ce type d'apprentissage, le comportement de référence est nécessaire. Il peut être une contrainte dans la mesure où, selon l'application, aucun comportement de référence n'est disponible. Un apprentissage qui ne nécessite pas de comportement de référence explicite mais seulement d'informations grossières, tel qu'un encouragement ou une pénalisation, est appelé apprentissage par renforcement. Ce type d'apprentissage fait partie d'un schéma plus large d'apprentissage, le "Credit-Assignment Problem" ou CAP.

b L'apprentissage par assignation de crédit

Il est intéressant de considérer l'apprentissage par assignation de crédit dans l'étude des systèmes distribués. On en parle alors également en terme d'apprentissage par pénalité /récompense ou en anglais "learning activity coordination". Le problème du CAP est celui de l'élaboration et l'assimilation d'un encouragement ou d'une pénalisation à l'issue de chaque décision interne prise par la machine d'apprentissage. En d'autres termes, on parle d'apprentissage par assignation lorsqu'une action particulière entraîne un ajustement heuristique des paramètres internes du réseau qui lui ont donné naissance (Haykin 1994). Un algorithme de critique est chargé d'établir un crédit ou un blâme pour renforcer ou réduire des poids synaptiques.

Ce type particulier de méthodes d'apprentissage est situé à mi-chemin entre l'apprentissage supervisé et l'apprentissage non supervisé. On peut le considérer comme étant soit supervisé, soit non supervisé, selon la source des informations utilisées pour déterminer le signal de renforcement $\mathbf{r}_k$ de la Figure 6-2. Si ce signal est uniquement fonction des paramètres internes au réseau, l'apprentissage sera non supervisé, alors que s'il est déterminé à partir d'informations externes on parlera plutôt d'apprentissage supervisé ou d'apprentissage semi-supervisé.

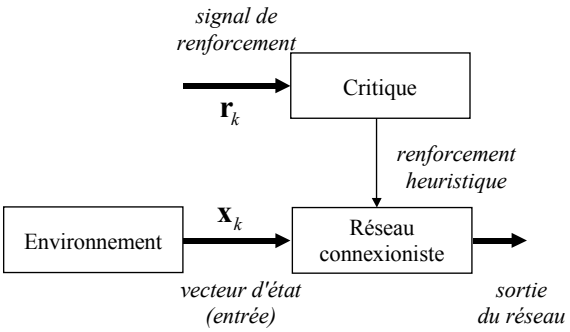


Figure 6-2 : Principe de l'apprentissage par assignation de crédit.

Comme types d'apprentissage CAP, il existe entre autre la méthode du "heuristic dynamic programming" présentée par exemple dans (Venayagamoorthy et al. 2000), l'algorithme "temporal difference" TD(λ) (Sutton 1988), le "Q-learning" (Watkins et Dayan 1992) (très bien résumé page 622 dans (Haykin 1999)) ou l'apprentissage renforcé (Sutton et Barto 1998).

L'apprentissage renforcé reste sans doute le plus courant des apprentissages par assignation de crédit. C'est un apprentissage semi-supervisé. En effet, l'algorithme de critique utilise des indications imprécises sur le comportement final souhaité (telles que « chaud ou froid », « échec ou succès » du réseau…) à travers le vecteur des variables explicatives de la Figure 6-3. L'ajustement des poids du réseau se fait uniquement à partir d'une simple évaluation de la qualité de sa réponse.

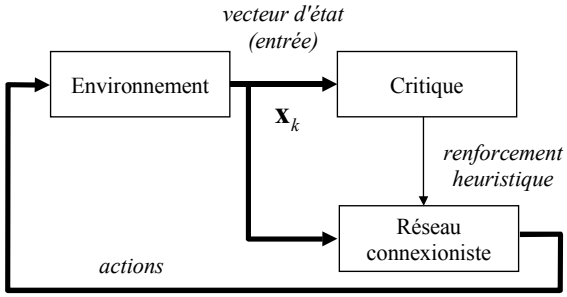


Figure 6-3 : Principe de l'apprentissage par renforcement.

Ces méthodes peuvent se révéler très utiles dans les domaines où l'on a peu d'information sur la réponse désirée. C'est en particulier le cas des problèmes de conduite de trajectoire en robotique où l'on se trouve souvent dans le cas où, pour une situation donnée, on a plusieurs comportements possibles du robot, sans que l'on puisse dire lequel est le meilleur. Dans un cadre plus général, l'apprentissage renforcé revient à maximiser la récompense du comportement d'un agent interagissant avec son environnement.

c L'apprentissage non supervisé

L'apprentissage supervisé s'effectue sous le contrôle d'un expert, alors que l'apprentissage non supervisé (appelé en anglais "unsupervised learning") est autodidacte. Les paramètres internes du réseau ne sont modifiés qu'avec les seuls stimuli, aucune réponse désirée n'est prise en considération. La Figure 6-4 montre que la sortie du réseau n'est pas utilisée par la procédure d'apprentissage.

Par nature, ce type d'apprentissage construit une représentation interne de la connaissance issue de l'environnement. Cet apprentissage est basé sur une mesure de la qualité de la représentation de la connaissance pour ajuster en conséquence les paramètres internes du réseau de neurones. Un critère interne souvent utilisé pour modifier les poids des neurones est la co-activation des neurones.

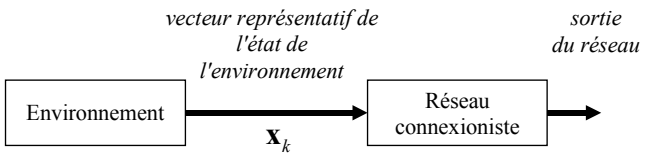


Figure 6-4 : Principe de l'apprentissage non supervisé.

L'apprentissage non supervisé s'applique aux réseaux à bases de prototypes. Dans ce cas, le réseau fonctionne comme un système de catégorisation (de "clustering") dans lequel les différents prototypes sont adaptés au fur et à mesure que sont présentées les données. Cette adaptation des prototypes peut être réalisée de trois façons différentes :

- par assimilation : si l'exemple présenté en entrée n'est proche d'aucun prototype connu, un nouveau prototype est crée en recopiant l'exemple nouveau ;
- par accommodation : si l'exemple présenté est reconnu comme proche d'un prototype, il est ajouté à l'ensemble des exemples représenté par ce prototype, et ce dernier est modifié de façon à rester le centre de gravité du nouvel ensemble ;
- par différentiation : si l'exemple présenté active un mauvais prototype, il faut réduire la région d'influence du prototype de façon à exclure l'exemple.

Dans tous les cas le but est d'effectuer le meilleur pavage possible du domaine d'entrée à l'aide des prototypes. On parle d'apprentissage non supervisé parce que la réponse attendue n'est pas donnée au réseau. Le réseau apprend d'une façon qui est prédéterminée par les lois d'évolution fixées par le concepteur, en particulier à travers le choix de la distance d'un exemple avec les différents prototypes. L'exemple le plus clair est le principe de compétition des cartes d'auto-organisation.

2.1.3 Propriétés de l'apprentissage

Il est difficile de mesurer la qualité d'un apprentissage. Si les outils statistiques tentent de façon très récente d'apporter des réponses, il n'existe pas d'outil universel pour qualifier ou quantifier un bon apprentissage. Le terme bon apprentissage est lui-même relatif, il dépend de l'objectif qui a guidé la conception du réseau, des informations disponibles, de l'application et de son environnement, du type d'apprentissage, de l'architecture du réseau, etc.

Un vocabulaire particulier s'est mis en place dans la littérature depuis un certain nombre d'années. On parle ainsi, à propos d'un réseau de neurones ayant appris, de sa capacité de généralisation et de sa sensibilité. Ces deux propriétés d'un apprentissage sont également connues sous les termes de biais et de variance.

Un réseau de neurones peut être considéré comme une forme de connaissances empiriques d'un phénomène physique dans son environnement. Ces connaissances ont été acquises ou mises au point durant la phase d'apprentissage. Le terme empirique utilisé pour qualifier les connaissances a été délibérément choisi pour signifier qu'elles sont issues d'un jeu de mesures représentatives de ce phénomène. Le réseau neuronal mémorise les connaissances sous la forme d'une matrice poids $\mathbf{w}$. Pour alléger les écritures nous n'allons pas indexer cette matrice dans le temps : nous allons simplement considérer que le réseau a appris, et que ses poids permettent une bonne estimation du phénomène (nous ne sommes pas intéressé par le processus d'apprentissage en lui même, mais plus par l'estimation qu'il a permis d'obtenir).

Supposons que le réseau de neurones réalise une estimation d'un phénomène représenté par la fonction $f(\mathbf{x})$. Cette estimation peut être exprimée par une fonction $F(\mathbf{x},\mathbf{w})$ dépendant d'un signal d'entrée $\mathbf{x}$ et de la matrice poids synaptique. N réalisations du vecteur d'entrée, et autant de réalisations d'un comportement désiré, des scalaire d, constituent la base d'apprentissage $\mathcal{T}$ du réseau :

$$\mathcal{T} = \{\mathbf{X}, D\} = \{\mathbf{x}_i, d_i\}_{i=1}^{N}. \tag{6.1}$$

Ne disposant pas de la connaissance exacte de la relation entre $\mathbf{X}$ et D, nous utilisons le modèle :

$$D = f(\mathbf{X}) + \epsilon, \tag{6.2}$$

où $f(.)$ est une fonction scalaire déterministe et ϵ est une erreur aléatoire attendue représentant l'ignorance de la dépendance de $\mathbf{X}$ et D.

Le modèle de régression donné par l'expression (6.2) possède deux propriétés intéressantes :

1. La valeur moyenne de l'erreur ϵ est nulle :

$$E[\epsilon|\mathbf{x}] = 0. \tag{6.3}$$

On peut directement en déduire (Haykin 1994) que la fonction de régression $f(\mathbf{x})$ est la moyenne conditionnelle du modèle de sortie D. Ainsi, si $\mathbf{X} = \mathbf{x}$:

$$f(\mathbf{x}) = E[D|\mathbf{x}]. \tag{6.4}$$

2. L'erreur ϵ n'est pas dépendante de la fonction de régression $f(\mathbf{x})$, donc :

$$E[\epsilon . f(\mathbf{x})] = 0. \tag{6.5}$$

Cette dernière propriété est connue sous le nom de principe d'orthogonalité. Ce principe affirme que toute information disponible sur D à travers l'entrée $\mathbf{X}$ est encodée dans la fonction de régression $f(\mathbf{X})$ (Haykin 1994).

Le modèle régressif est une description mathématique utilisant le formalisme des variables aléatoires appliqué à un processus physique. Le réseau entraîné sur la base $\mathcal{T}$ constitue alors également une approximation de cette fonction.

En considérant la base $\mathcal{T}$, les poids $\mathbf{w}$ du réseau peuvent être obtenus en minimisant une fonction de coût du type :

$$\mathcal{E}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}\left(d_i - F(\mathbf{x}_i, \mathbf{w})\right)^2 . \tag{6.6}$$

Sachant que toutes les connaissances empiriques de $\mathcal{T}$ sont contenues après apprentissage dans les poids $\mathbf{w}$, $F(\mathbf{x}, \mathbf{w})$ est équivalent à $F(\mathbf{x}, \mathcal{T})$. L'expression (6.6) peut alors s'écrire plus convenablement avec $E_{\mathcal{T}}[.]$, l'opérateur moyenne sur l'ensemble de la base d'apprentissage $\mathcal{T}$:

$$\mathcal{E}(\mathbf{w}) = \frac{1}{2}E_{\mathcal{T}}\left[\left(d_i - F(\mathbf{x}_i, \mathcal{T})\right)^2\right]. \tag{6.7}$$

Sachant que :

$$\begin{aligned} d - F(\mathbf{x}, \mathcal{T}) &= \left(d - f(\mathbf{x})\right) + \left(f(\mathbf{x}) - F(\mathbf{x}, \mathcal{T})\right) \\ &= \epsilon + \left(f(\mathbf{x}) - F(\mathbf{x}, \mathcal{T})\right) \end{aligned} \tag{6.8}$$

$\mathcal{E}(\mathbf{w})$ peut s'écrire :

$$\mathcal{E}(\mathbf{w}) = \frac{1}{2}E_{\mathcal{T}}\left[\epsilon^2\right] + \frac{1}{2}E_{\mathcal{T}}\left[\left(f(\mathbf{x}) - F(\mathbf{x}, \mathcal{T})\right)^2\right] + E_{\mathcal{T}}\left[\epsilon.\left(f(\mathbf{x}) - F(\mathbf{x}, \mathcal{T})\right)\right], \tag{6.9}$$

qui se simplifie en :

$$\mathcal{E}(\mathbf{w}) = \frac{1}{2}E_{\mathcal{T}}\left[\epsilon^2\right] + \frac{1}{2}E_{\mathcal{T}}\left[\left(f(\mathbf{x}) - F(\mathbf{x}, \mathcal{T})\right)^2\right] \tag{6.10}$$

Dans l'expression (6.10), l'énergie $\mathcal{E}(\mathbf{w})$ est la somme de deux termes : la variance de l'erreur du modèle de régression calculée sur la base $\mathcal{T}$ et la variance de la différence entre le modèle de régression et l'estimation effective du réseau. Le premier terme est propre au modèle utilisé, il ne dépendant pas des poids $\mathbf{w}$ du réseau et n'entre pas en compte lors de la minimisation de $\mathcal{E}(\mathbf{w})$ par rapport à $\mathbf{w}$. Ainsi, la solution optimale $\mathbf{w}^*$ des poids qui minimise l'énergie $\mathcal{E}(\mathbf{w})$ est uniquement tributaire de la grandeur (Geman et al. 1992) :

$$L_{av}\left(f(\mathbf{x}), F(\mathbf{x}, \mathcal{T})\right) = E_{\mathcal{T}}\left[\left(f(\mathbf{x}) - F(\mathbf{x}, \mathcal{T})\right)^2\right]. \tag{6.11}$$

La qualité de l'estimée $F(\mathbf{x}, \mathcal{T})$ de d peut donc directement être évaluée avec $L_{av}\left(f(\mathbf{x}), F(\mathbf{x}, \mathcal{T})\right)$. Cette grandeur est le fondement mathématique sur lequel repose le compromis entre le biais et la variance qu'il faut faire en prenant $F(\mathbf{x}, \mathbf{w})$ pour approximer $f(\mathbf{x})$.

L'expression (6.11) s'écrit aussi[2] :

$$L_{av}\left(f(\mathbf{x}), F(\mathbf{x}, \mathcal{T})\right) = E_{\mathcal{T}}\left[\left(E[D|\mathbf{X} = \mathbf{x}] - F(\mathbf{x}, \mathcal{T})\right)^2\right]. \tag{6.12}$$

La différence utilisée dans (6.12) peut s'écrire :

$$E[D|\mathbf{X} = \mathbf{x}] - F(\mathbf{x}, \mathcal{T}) = \left(E[D|\mathbf{X} = \mathbf{x}] - E_{\mathcal{T}}\left[F(\mathbf{x}, \mathcal{T})\right]\right) + \left(E_{\mathcal{T}}\left[F(\mathbf{x}, \mathcal{T})\right] - F(\mathbf{x}, \mathcal{T})\right). \tag{6.13}$$

Après simplification (toutes les étapes sont détaillées dans (Haykin 1994)), l'expression (6.12) devient :

$$\begin{aligned} &L_{av}\left(f(\mathbf{x}), F(\mathbf{x}, \mathcal{T})\right) \\ &= E_{\mathcal{T}}^2\left[F(\mathbf{x}, \mathcal{T}) - E[D|\mathbf{X} = \mathbf{x}]\right] + E_{\mathcal{T}}\left[\left(F(\mathbf{x}, \mathcal{T}) - E_{\mathcal{T}}\left[F(\mathbf{x}, \mathcal{T})\right]\right)^2\right] \end{aligned} \tag{6.14}$$

[2] C'est une approche tout à fait « statistique » puisque $f(\mathbf{x})$ est considérée dans cette étape comme une espérance conditionnelle (qui est constante sur la base $\mathcal{T}$).

Dans cette dernière expression (6.14), il ressort deux grandeurs :

- le biais

$$E_{\mathcal{T}}\left[F(\mathbf{x},\mathcal{T})-E[D|\mathbf{X}=\mathbf{x}]\right], \tag{6.15}$$

- et la variance

$$E_{\mathcal{T}}\left[\left(F(\mathbf{x},\mathcal{T})-E_{\mathcal{T}}\left[F(\mathbf{x},\mathcal{T})\right]\right)^2\right]. \tag{6.16}$$

a Le biais

Le biais d'un réseau se définit intuitivement comme une mesure de sa capacité à généraliser correctement sur un ensemble de tests (reflétant la sortie moyenne sur l'ensemble des bases d'apprentissage).

On appelle capacité de généralisation la capacité du réseau à réagir correctement lors de son utilisation à des entrées qui n'ont pas été vues lors de l'apprentissage. C'est cette capacité à généraliser qui rend les réseaux de neurones attractifs par rapports à d'autres approches.

Lorsque les capacités de généralisation d'un réseau sont très faibles, on emploie l'expression « apprentissage par cœur » pour désigner le fait que ce réseau ne soit capable de répondre que pour des entrées présentées dans la phase d'apprentissage et ne puisse pas trouver de réponse adéquate autrement.

b La variance

La variance d'un réseau se définit de façon similaire, comme une mesure de la sensibilité de la sortie d'un réseau à partir des exemples sur lesquels il a été entraîné ; autrement dit, la variance indique dans quelle mesure des résultats identiques auraient été obtenus si une base d'apprentissage différente avait été utilisée.

c Le dilemme biais/variance

L'apprentissage d'un réseau, sur la base d'une taille fixe et s'il est optimisé pour réduire le biais, aura pour conséquence une valeur élevée de la variance. Pour un réseau, l'unique façon de réduire à la fois le biais et la variance, serait de procéder à un apprentissage sur un corpus d'apprentissage de taille infinie !

Une éventuelle solution pour contourner ce dilemme consiste à volontairement introduire un biais dans l'estimation faite par le réseau qui aura pour conséquence la réduction de la variance. Ce terme de biais ne doit pas « casser » l'apprentissage. Pour cela, il doit être mis au point spécifiquement pour chaque application en contraignant l'architecture du réseau. La meilleure façon de procéder consiste dès lors à utiliser des connaissances a priori sur la tâche à apprendre.

Pour finir, remarquons que Geman, Bienenstock et Doursat (Geman et al. 1992) utilisent une simple moyenne pour analyser le dilemme biais/variance. L'erreur MSE d'un estimateur peut être exprimée comme la somme de deux termes, tous les deux fonctions des poids $\mathbf{w}$:

$$L_{av}\left(f(\mathbf{x}),F(\mathbf{x},\mathcal{T})\right)=biais^2+variance. \tag{6.17}$$

Des travaux de recherche concernant le biais et la variance des réseaux modulaires sont en cours (Breiman 1996b; Heskes 1998; Kohavi et Wolpert 1996; Tibshirani 1996; Wolpert 1997).

Le biais et la variance des Mélanges d'Experts ont été étudiés dans (Jacobs 1997). Auparavant, Clemen et Winkler (Clemen et Winkler 1985) quantifient l'effet de la corrélation positive et négative, ainsi que l'effet de la non corrélation des experts, lorsque leurs estimations sont combinées par la règle Bayésienne. Perrone (Perrone 1993), dans son mémoire de thèse tente également d'apporter quelques éléments de réponse.

En fait, nombreux sont les auteurs qui se servent aujourd'hui des travaux de Geman, de Bienenstock et de Doursa, et nombreux sont ceux qui reprennent leurs mesures de performances pour

valider ou non l'apprentissage de réseaux qui sont bien plus complexes que le réseau multicouche (Breiman 1996b ; Heskes 1998 ; Jacobs 1997; Kohavi et Wolpert 1996 ; Tibshirani 1996 ; Wolpert 1997). Tous ces auteurs utilisent leurs mesures de performances pour clairement mettre en évidence comment le biais et la variance dépendent de la complexité du réseau. Le bon compromis doit être trouvé en sachant que lorsque la complexité augmente, le biais décroît et la variance croît ; inversement, lorsque la complexité baisse, la biais augmente et la variance décroît.

2.2 Apprentissage et adaptation

2.2.1 Adaptation

Dans cette étude, le sens accordé au terme adaptation est celui de l'ajustement des paramètres internes d'une structure en général. L'adaptation concerne des structures telles que les algorithmes, les filtres ou encore les réseaux de neurones artificiels. La Figure 6-5 montre le bloc-diagramme de l'adaptation d'un filtre adaptatif, mais il est valable pour les autres structures (on reconnaît d'ailleurs le principe de l'apprentissage supervisé d'un réseau de neurones, Figure 6-1).

Si l'adaptation d'un filtre est quelque chose de parfaitement bien connu, on peut se demander, concernant les réseaux de neurones, ce qui distingue l'adaptation de l'apprentissage.

L'apprentissage sous-entend pour un réseau à la fois l'assimilation, et la mémorisation des connaissances. Si un système d'apprentissage est basé sur la mémoire, donc sur sa capacité de se rappeler et d'exploiter les expériences passées, l'adaptation est un ajustement des paramètres internes limité aux toutes dernières expériences. La nuance vient précisément de la prise en considération du facteur temps dans le processus d'apprentissage.

Un processus d'apprentissage adapte ses paramètres internes en fonction des entrées (et des sorties, s'il s'agit d'un apprentissage supervisé) qui lui sont présentées, et garde en mémoire sous une certaine forme toutes ces expériences. Un processus d'adaptation ajuste les paramètres de sa structure, mais ne mémorise pas pour pouvoir tirer profit des expériences passées. Ainsi, l'adaptation peut être vue comme un cas particulier de l'apprentissage (ou inversement) : c'est une mémoire à très *court terme*, qui se « rappelle » uniquement des derniers instants et qui « oublie » toutes les expériences plus anciennes.

La limite entre les deux est subtile. En jouant sur le coefficient d'apprentissage d'un réseau et en le gardant élevé, on peut altérer les capacités de mémoire à plus *long terme*. En effet, les informations les plus récentes, fortement pondérées, au fur et à mesure qu'elles sont assimilées, vont « obnubiler » la mémoire du réseau. Un tel réseau présente alors un comportement adaptatif, au sens fort du terme, c'est-à-dire lié à la nature spatio-temporelle implicite de l'apprentissage.

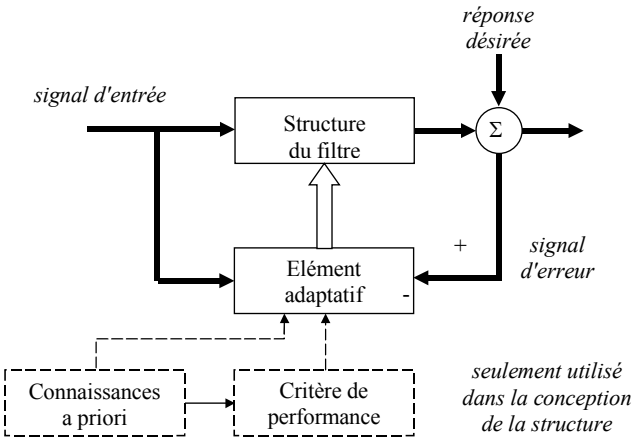


Figure 6-5 : Block diagramme d'un filtre adaptatif.

Comme exemples d'algorithmes adaptatifs, on peut citer un prédicteur linéaire qui utilise les deux dernières mesures pour estimer une valeur future. Entrent également dans la catégorie des algorithmes adaptatifs, l'estimateur des moindres carrés récursifs, ainsi que la grande majorité des filtres numériques itératifs.

2.2.2 Environnements non stationnaires, apprentissage et adaptation

Lorsqu'un réseau neuronal est placé dans un environnement stationnaire (i.e., dont les propriétés statistiques ne changent pas au cours du temps), les propriétés statistiques peuvent en théorie être apprises par le réseau de neurones. Si les poids synaptiques des neurones ont été déterminés à partir d'une base d'apprentissage représentative de l'environnement, ils capturent les statistiques sous-jacentes de l'environnement (on peut dès lors geler la valeur de ces poids). Le système d'apprentissage est de cette façon basé sur la mémoire : se rappeler pour exploiter les expériences passées apprises.

Cependant, les environnements sont généralement non stationnaires. Cela signifie que les paramètres statistiques des informations portées par les signaux issus de l'environnement varient au cours du temps. Le traditionnel apprentissage (supervisé) n'est pas, dans ce cas, une méthode adéquate pour apprendre le phénomène. Le réseau neuronal (unidirectionnel) n'est pas conçu pour tenir compte des incontournables variations des statistiques de l'environnement.

Les différentes façons de procéder pour tenir compte de ce phénomène consistent, soit à poursuivre la phase d'apprentissage, soit à explicitement le compenser à l'aide d'algorithmes adaptatifs. Le premier cas revient en quelque sorte à rendre le réseau adaptatif et à trouver un compromis entre apprentissage et adaptation. On choisit souvent dans ce cas un coefficient d'apprentissage empirique, mais des méthodes plus rigoureuses commencent à voir le jour.

a Choix empirique du coefficient d'apprentissage

Plongé dans un environnement non stationnaire, un réseau de neurones artificiels pour fournir des performances correctes ne peut pas occulter la dérive temporelle des signaux. Une façon assez évidente de procéder consiste à perpétuellement ajuster la connaissance du réseau à l'aide des plus récentes données représentatives de l'environnement. Haykin (Haykin 1994) pense d'ailleurs que dans un environnement non stationnaire, plus un réseau est d'une « certaine manière » adaptatif (au sens fort du terme), meilleures seront ses performances. Le coefficient d'apprentissage se doit d'être adéquat, mais on procède généralement par essai/erreur.

Il faut savoir qu'un coefficient d'apprentissage trop petit[3] a pour unique conséquence l'impossibilité de compenser les dérives des paramètres de l'environnement. Par contre, un coefficient d'apprentissage trop grand, a tendance à faire changer les valeurs des poids synaptiques du réseau à partir des nouvelles mesures, elles, influencées par des statistiques changeantes, au point de remplacer la connaissance déjà acquise sur le système par ces nouvelles mesures. Lorsqu'un réseau perd toute son expérience passée au profit d'une connaissance tributaire de paramètres supplémentaires (ici les propriétés statistiques changeantes de l'environnement), on parle de *désapprentissage*. La connaissance du réseau ne sera alors valable et utilisable que si ces paramètres supplémentaires utilisés pour l'apprentissage sont connus, ou si on se retrouve dans les mêmes conditions que celles de la phase d'apprentissage. Un réseau ne peut répondre correctement que pour des stimuli déjà appris, ou pour des stimuli proches de ceux déjà appris.

[3] Les termes trop petit et trop grand, en soi ne veulent rien dire. C'est par rapport à la tâche à apprendre, mais aussi par rapport aux variations des propriétés statistiques de l'environnement, donc à la vitesse de dérive des paramètres, et sur l'influence qu'ils ont sur les signaux issus du système qu'il faut les comprendre.

b Optimisation du coefficient d'apprentissage

Une façon de procéder, pour corriger l'apprentissage, consiste à le poursuivre en trouvant un coefficient d'apprentissage qui permette d'intégrer les nouvelles informations sans toutefois altérer la mémoire du réseau. Dans (Heskes et Kappen 1991), les auteurs apportent une solution pour un apprentissage en ligne. Ils considèrent qu'à chaque instant *k*, l'adaptation des poids $\mathbf{w}_k$ par rapport au vecteur d'entrée $\mathbf{x}_k$ est de la forme :

$$\Delta\mathbf{w}_k = \mathbf{w}_{k+1} - \mathbf{w}_k = \eta\ \mathbf{f}(\mathbf{w}_k, \mathbf{x}_k)\,, \tag{6.18}$$

où η est un coefficient d'apprentissage et $\mathbf{f}(.,.)$ la règle d'apprentissage elle-même. La rétropropagation du gradient (BP), et la règle d'apprentissage de la carte auto-organisatrice de Kohonen fonctionnent sur cette description. Si le vecteur est tiré aléatoirement de la base d'apprentissage, alors l'apprentissage en ligne peut être considéré comme un processus de Markov du premier ordre (Wiegerinck et Heskes 1994). La règle (6.18) permet de directement déduire l'évolution de la probabilité $p(\mathbf{w},k)$ que le réseau soit dans l'état $\mathbf{w}$ à l'instant *k*. Il se trouve que dans la nature, les données sont le plus souvent corrélées. Autrement dit, la théorie qui suppose que les vecteurs d'entrée sont présentés aléatoirement pour l'apprentissage des poids d'un réseau n'est plus valable. Ces auteurs proposent une solution uniquement valable lorsque $\mathbf{x}$ suit la loi d'un processus de Markov d'ordre un, c'est-à-dire si $\mathbf{x}_k$ ne dépend que de son prédécesseur $\mathbf{x}_{k-1}$. Ceci revient à considérer de toutes petites variations de $p(\mathbf{w},k)$ à l'aide d'une probabilité de transition $p(\mathbf{x}|\mathbf{x}_{k-1})$. Leur méthode peut être étendue si $\mathbf{x}_k$ est Markovien à un ordre fini.

Lorsque les données sont corrélées dans l'espace temporel, on dit qu'elles ne sont pas Markoviennes et l'évolution de $p(\mathbf{w},k)$ est alors beaucoup moins triviale.

c L'adaptation pour suppléer l'apprentissage

Il faut pouvoir tenir compte de la dérive des propriétés statistiques d'un environnement non stationnaire. Dans cette optique, l'adaptation peut ainsi suppléer le processus d'apprentissage : si le réseau neuronal est chargé de reproduire le comportement du système, un processus d'adaptation peut se charger de compenser la dérive des propriétés statistiques observées de l'environnement. Ceci conduit à diverses combinaisons de processus d'apprentissage avec des algorithmes adaptatifs. Ces combinaisons peuvent être séquentielles ou être basées sur un schéma de prédiction/comparaison. Le processus d'adaptation peut également directement être inclus dans l'apprentissage.

L'approche logique la plus simple est une combinaison séquentielle. La compensation se fait en amont par le processus d'adaptation, le réseau apprend sur des données compensées (Figure 6-6). L'inconvénient de cette approche réside dans le fait que les performances du réseau sont directement tributaires de la qualité de la compensation.

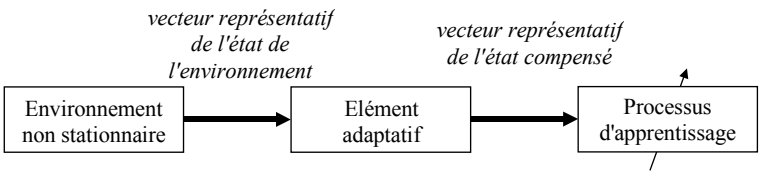


Figure 6-6 : Compensation séquentielle de l'apprentissage.

Une méthode souvent plus efficace que la compensation amont est basée sur le principe de la prédiction/correction. Cette méthode, une forme de compensation avale, illustrée par la Figure 6-7, procède en deux étapes : le signal de l'environnement $\mathbf{x}_{k-1}$ à l'instant précédent est prédit puis comparé avec la valeur courante $\mathbf{x}_k$. Le signal d'erreur $\mathbf{e}_k = \hat{\mathbf{x}}_k - \mathbf{x}_k$ obtenu, appelé innovation, est utilisé par la suite pour une étape de correction. Cette erreur contient l'information nouvelle du vecteur d'entrée à

l'itération courante. Une innovation nulle signifie qu'aucune nouveauté n'est apportée par le vecteur d'entrée (le réseau neuronal n'a pas besoin d'apprendre puisqu'il sait comment traiter l'entrée). A l'inverse, une innovation non nulle implique que quelque chose de neuf, de non prévu ou non attendu, s'est produit, et que le système d'apprentissage doit en tenir compte en l'intégrant dans sa mémoire.

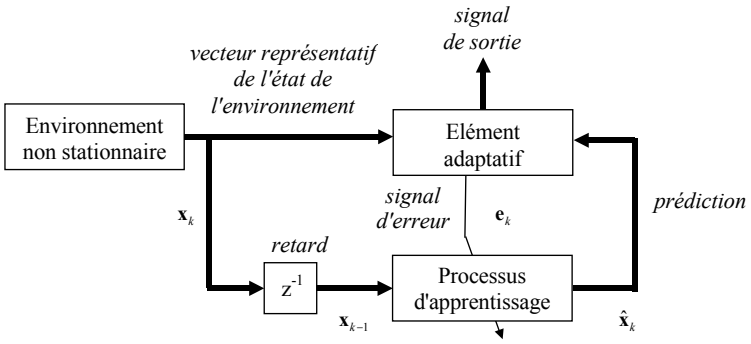


Figure 6-7 : Compensation de l'apprentissage basée sur le principe de la prédiction/comparaison.

L'innovation peut être exploitée de deux façons différentes (Haykin 1994, page 70) :

- Lorsqu'elle fournit un signal de correction au réseau afin que celui-ci puisse efficacement ajuster ses paramètres internes, on parle d'apprentissage par retour d'adaptation. La modification des poids du réseau suffit à délivrer un signal de sortie fonction des statistiques de l'environnement. Les filtres numériques adaptatifs, et le filtre de Kalman en particulier, sont connus pour être parmi les procédures les plus efficaces.
- Lorsqu'elle sert à la fois à régler l'apprentissage et à compenser le signal de sortie, elle va au fur et à mesure permettre un apprentissage plus fin.

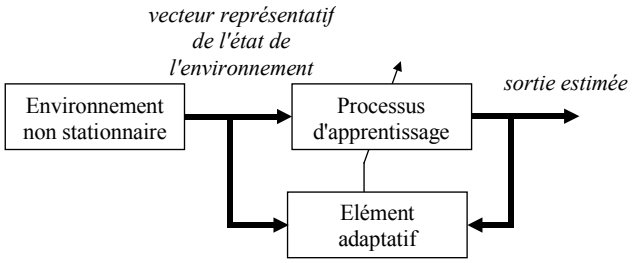


Figure 6-8 : Compensation de l'apprentissage par l'erreur.

Une dernière approche adaptative, beaucoup plus rare, utilise les informations de l'environnement comme support d'adaptation des poids du réseau. La procédure d'adaptation est alors directement intégrée dans le mécanisme d'apprentissage comme le montre la Figure 6-8. Cette approche est souvent difficile à mettre en œuvre. Elle est conçue spécifiquement pour chaque système, est difficilement généralisable, et dépend de l'application et de la loi d'apprentissage.

2.2.3 Bilan

Les réseaux neuromimétiques sont parfaitement à l'aise dans les environnements stationnaires. Leurs performances diminuent lorsque l'environnement dans lequel ils opèrent est non stationnaire. En effet, les connaissances acquises par un réseau ne sont valides que dans une portion de l'espace-temps

donnée. Lorsque les données à apprendre sont influencées ou varient au cours du temps, inconsciemment la procédure d'apprentissage élabore un modèle de connaissance dépendant du temps. Pour un apprentissage efficace, une compensation doit obligatoirement être envisagée.

Cette compensation, pour être efficiente, doit être fonction du degré de non stationnarité observé. Ainsi, l'utilisation de fenêtres d'apodisation constitue la solution la plus raisonnable dans le cas d'un système pseudo-stationnaire. Des solutions plus coûteuses consistent soit à rajouter au réseau des entrées supplémentaires, soit à utiliser un réseau récurrent et à introduire des lignes à retard (les "tapped time delay" très utilisées en traitement du signal).

Ajouter des entrées supplémentaires au réseau introduit un inconvénient majeur, celui du choix des grandeurs supplémentaires à fournir au réseau. Les grandeurs rajoutées doivent être suffisamment représentatives des variations des statistiques de l'environnement. En fait, dans cette approche, on cherche à ajouter des poids supplémentaires au réseau, pour pouvoir explicitement compenser les variations d'un modèle à apprendre. L'efficacité se paie au prix fort puisque la solution finale aboutit à des réseaux de taille plus que conséquente pour apprendre des modèles qui varient certes en fonction du temps, mais qui restent relativement simples en général.

Alors que de nouvelles voies sont prospectées, telles que l'insertion des statistiques d'ordres supérieurs (Sanchez et Bernal 2000) dans les modèles neuronaux, les approches les plus efficaces à ce jour reposent sur l'association d'un réseau de neurones avec un filtre adaptatif pour combiner apprentissage et adaptation.

Le principe de l'adaptation est fondamental dans l'implémentation VLSI ("Very Large Scale Integration") des réseaux de neurones. Les procédures adaptatives sont utilisées pour compenser les imperfections analogiques introduites par les systèmes physiques comme par exemple les dérives, les fluctuations et les incertitudes. L'application la plus populaire est l'élaboration d'une rétine électronique (Mead 1990) dans laquelle sont inclus de manière analogique, un processus d'apprentissage et une structure adaptative. Cette application a depuis été reprise et améliorée (Cauwenberghs 1996), mais elle a surtout entraîné une prise de conscience et relancé des développements plus théoriques.

2.3 Architecture

Les différents modèles neuromimétiques se distinguent par les contraintes de connectivité qu'ils supportent. L'architecture d'un réseau de neurones désigne donc la structure d'interconnexion des neurones qui constituent un réseau.

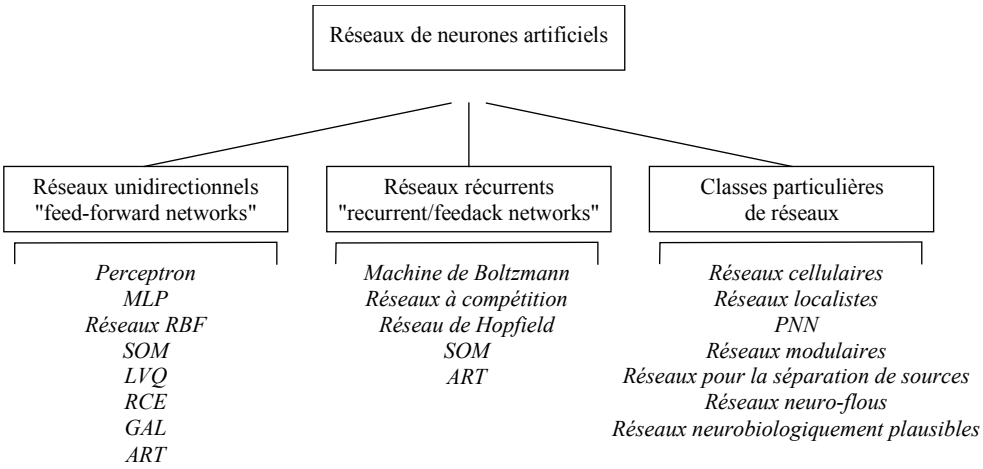


Figure 6-9 : Une taxonomie des principales architectures neuronales.

Dans la majorité des ouvrages de synthèses, tel l'incontournable écrit de Haykin (Haykin 1994), réédité et amélioré (Haykin 1999), ou (Jodouin 1994a), deux grandes classes de réseaux sont séparées : les réseaux récurrents (comme le réseau de Hopfield) et les réseaux unidirectionnels où les boucles ne sont pas autorisées (comme le Perceptron multicouche).

Nous n'allons pas développer toutes les architectures, ni même fournir une liste complète de toutes les architectures existantes. Cette modeste étude se veut juste d'introduire une classification des principales architectures afin de fournir un aperçu de la richesse et de l'ingéniosité de certaines structures neuronales, et ceci en vue d'introduire au Chapitre 3 la notion de modularité.

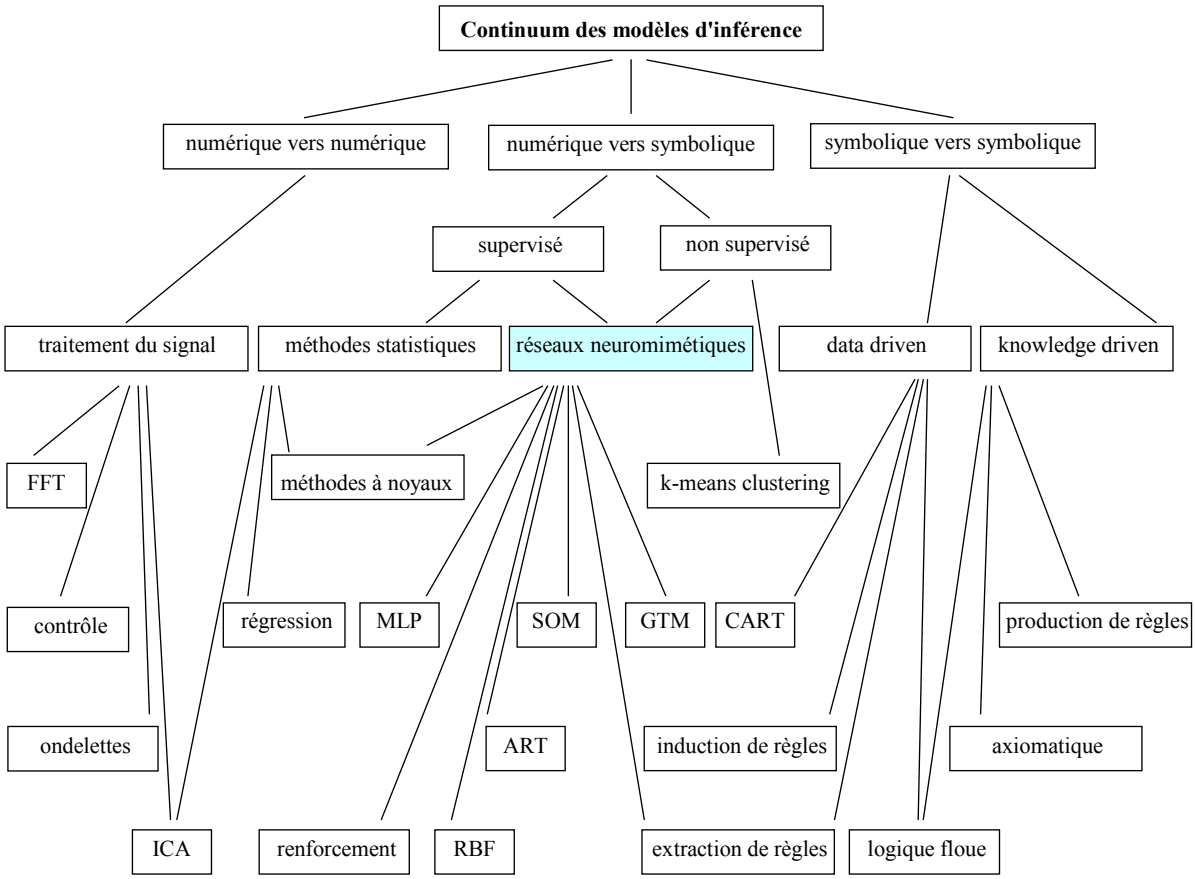


Figure 6-10 : Continuum des modèles d'inférence.

Certaines relations sont certainement ambiguës et cette description n'est pas exhaustive (il manque notamment les méthodes Bayésiennes qui découlent des trois principales branches).
Sur cette figure, pour les acronymes, il faut lire :
ART : "Adaptive Resonance Theory",
CART : "Classification And Regression Trees",
FFT : "Fast Fourier Transform",
GTM : "Generative Topographic Map,
ICA : "Independant Componant Analysis",
MLP : "Multi-Layer Perceptron",
RBF : "Radial Basis Function",
SOM : "Self-Organizing Map".

La taxonomie proposée par la Figure 6-9 n'est pas nouvelle. Elle est basée sur une classification classique (Amy 1996 ; Jain et al. 1996). L'idée ici n'est pas de présenter toutes les structures des architectures neuronales, mais plutôt de donner un aperçu rapide ainsi qu'une vue plus

générale. Cette classification est quelque peu réductrice (à vrai dire par leur nature même, toutes les classifications sont réductrices). Elle ne représente que les architectures à structures fixes, ne met en évidence ni les architectures à structure évolutive, ni les architectures à base de prototypes. De plus, toute architecture n'entrant ni dans le cadre des réseaux unidirectionnels, ni dans le cadre des réseaux récurrents, a été classée dans une troisième classe. Les réseaux topologiques ont été, quant à eux, considérés selon leurs variantes soit comme des réseaux unidirectionnels, soit comme des réseaux récurrents.

Aussi, et pour ne pas nous limiter à la seule perspective donnée par une taxonomie, nous proposons également, Figure 6-10, un continuum des modèles d'inférences qui a l'avantage de placer les réseaux de neurones dans un contexte plus général.

Ce continuum, du haut vers le bas, nous fait passer des concepts les plus abstraits aux modèles les plus applicatifs. A propos d'application, on trouvera de nombreux éléments sur la conception des réseaux neuronaux dans (Hagan et al. 1995) ou dans quelques autres ouvrages de synthèses, notamment sur les réseaux et traitement du signal (Hérault et Jutten 1994), sur les réseaux et les statistiques (Thiria et al. 1997), sur les réseaux et l'approche Bayésienne (Neal 1996), sur les réseaux et le "pattern recognition" (Bishop 1995), sur les réseaux et le contrôle (Miller et al. 1990b), et sur les réseaux et le contrôle robotique (Omidvar et van der Smagt 1997).

2.3.1 Une revue de quelques architectures

a Les réseaux unidirectionnels

Un réseau unidirectionnel (en anglais : "feedforward network") se présente sous la forme d'une ou de plusieurs couches successives. L'interconnexion des neurones est réalisée entre couches successives selon un sens unique de transfert des informations. De façon typique, les sorties des neurones d'une couche deviennent les entrées des neurones de la couche suivante. Le transfert de l'information se fait à partir d'une couche d'entrée qui souvent n'effectue aucun traitement, vers une couche de sortie, via les couches intermédiaires appelées couches cachées. L'interconnexion des neurones entre deux couches adjacentes est soit totale, soit partielle.

La Figure 6-11 fournit un exemple d'un réseau unidirectionnel : un réseau multicouche, avec une couche cachée, deux entrées et deux sorties.

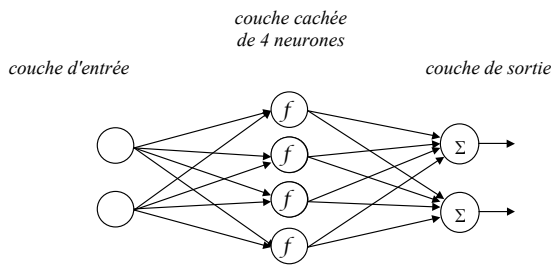


Figure 6-11 : Un réseau multicouche à 2 entrées, 2 sorties, et une couche cachée.
f représente la fonction d'activation du neurone.

b Les réseaux récurrents

Un réseau récurrent se présente sous la forme d'un réseau mono- ou multicouche et se différencie des réseaux unidirectionnels par la présence de boucles de réaction interne. Dans les réseaux récurrents, chaque neurone peut être connecté à des neurones voisins déjà activés, ou à lui-même. La Figure 6-12 montre un réseau récurrent typique constitué de deux entrées, de deux sorties et à interconnexion totale.

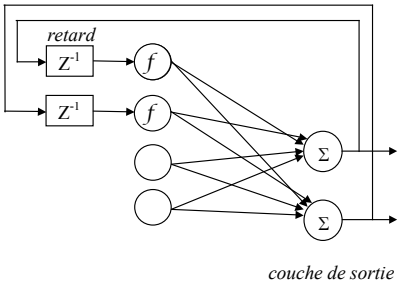


Figure 6-12 : Architecture d'un réseau récurrent mono-couche à interconnexion totale.

Grâce aux délais introduits par les boucles de réaction, un réseau récurrent présente un comportement dynamique qui lui permet d'intégrer explicitement la dimension temporelle. L'aboutissement est un processus de relaxation au cours duquel le réseau passe par une série d'états d'activation, l'état à l'instant k+1 étant calculé en fonction de l'état, ou des états précédents. Le réseau se comporte comme un système dynamique tendant vers des états attracteurs, des cycles attracteurs ou des comportements chaotiques.

c Les autres classes de réseaux

Parmi les réseaux qui peuvent être mentionnés, et qui sont mal mis en évidence par la taxonomie proposée par la Figure 6-9, citons les réseaux à base de prototypes.

Cette classe de réseaux a pour objectif de mémoriser un nombre limité d'exemples ou de prototypes représentatifs de chaque classe dans le cas de la classification, ou de chaque partie de la fonction dans le cas de l'approximation. Les réseaux à base de prototypes sont en général construits avec trois couches, dont la couche centrale (la couche cachée) possède des unités qui représentent chacune un prototype.

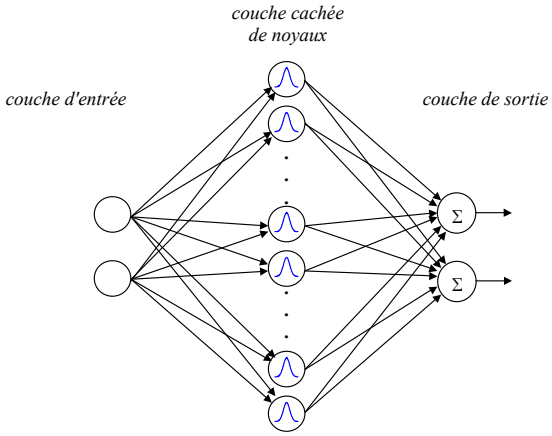


Figure 6-13 : Architecture du réseau RBF.

Le réseau typique est le réseau de neurones à fonctions radiales de base, le réseau RBF, tel que celui représenté sur la Figure 6-13. Le RBF est très utilisé en classification, en approximation fonctionnelle ou pour l'approximation de densités de probabilité dans les processus de décision Bayésienne. Dans cette dernière application, les neurones de la couche cachée sont des noyaux ou des fonctions Gaussiennes. Le principe général de fonctionnement de ce réseau est très bien présenté dans

(Haykin 1994). Parmi les autres réseaux appartenant à cette classe, citons succinctement les réseaux SOM, LVQ, ART, RCE, GAL, les réseaux probabilistes, ou encore les deux algorithmes développés dans (Fritzke 1994) : l'un est une carte auto-adaptative incrémentale et l'autre un réseau incrémental du type RBF.

Les réseaux topologiques peuvent également être considérés comme une classe de réseaux à part. Ils seront abordés plus avant dans ce document, Enfin, le concept de réseaux de neurones modulaires sera également le sujet d'une étude plus poussée.

2.3.2 Les réseaux à architecture évolutive

Les architectures évolutives sont conçues pour résoudre le problème difficile du choix de l'architecture du réseau. L'apprentissage de ce type de réseau fait l'objet d'une étude à part, en raison du changement de la structure du réseau au cours de la phase d'apprentissage.

Le principe qui préside au fonctionnement de ces modèles est d'ajuster à la fois la structure et les poids des neurones par rapport au corpus d'apprentissage qui leur est présenté. La phase d'apprentissage terminée, un tel réseau doit aboutir à une structure optimale par rapport à la complexité du phénomène appris.

On dit de ces réseaux qu'ils sont à architecture évolutive et la littérature anglophone utilise les termes de "growing/pruning". L'apprentissage évolutif utilise deux principes opposés, les méthodes d'accroissement et les méthodes de simplification (Alpaydin 1991; Hérault et Jutten 1994).

a Les méthodes d'accroissement

Un réseau à apprentissage incrémental fonctionne de la manière suivante : une taille minimale du réseau est fixée au départ, puis des neurones sont ajoutés lorsque l'erreur ne baisse plus au cours de la phase d'apprentissage. La difficulté de ces méthodes est liée au choix d'un bon critère d'ajout de neurone. Cette façon de construire un réseau est logiquement dénommée méthode par construction ou méthode constructive. Le terme approprié en anglais est "growing network".

b Les méthodes de simplification

A l'inverse des méthodes d'accroissement, les méthodes de simplification commencent avec un réseau sur-paramétré, puis des neurones sont supprimés au cours de l'apprentissage. Les critères de sélection des neurones (ou des connexions) sont plus faciles à élaborer dans l'hypothèse où l'élagage intervient lorsque le réseau a convergé, puisque cet état peut être utilisé comme référence. Ces méthodes sont appelées méthodes par suppression ou méthodes par élagage, et l'expression équivalente dans la littérature anglophone est "pruning network".

On trouvera dans (Alpaydin 1991) une synthèse succincte, qui sépare les méthodes fondées sur un calcul de sensibilité des méthodes basées sur un terme de pénalité. Cet auteur préférera presque toujours une méthode par élagage à une méthode par construction. Une synthèse plus détaillée est fournie dans (Reed 1993).

c Les méthodes évolutionnistes

Le problème de la structure qui sied le mieux à la complexité d'un problème peut être soumis aux théories évolutionnistes. Ces théories, qui n'ont pas encore atteint une pleine maturité, fournissent depuis quelques années une palette d'outils intéressants. En particulier, les algorithmes génétiques et les algorithmes évolutionnistes semblent être les plus adéquats pour paramétrer la structure d'un réseau de neurones.

Dans (Fogel et al. 1990), dans (Yao 1993), dans (Chellapilla et Fogel 1999) ou dans (Yao 1999), les auteurs tentent de faire une synthèse entre les théories évolutionnistes et les réseaux de neurones artificiels ; la dernière référence étant sans doute la plus complète.

La paramétrisation d'un réseau par un algorithme évolutionniste la plus au point à ce jour, concerne l'optimisation de cartes SOM (Villmann 2001). La limitation majeure de l'approche réside dans un effort d'apprentissage important. En effet, pour chaque génération (une étape pour l'algorithme évolutionniste), tous les individus d'une population représentant un réseau neuronal doivent être entraînés selon leur propre mécanisme.

On trouve également des imbrications de ces algorithmes avec les réseaux unidirectionnels et la règle BP (Zhao et Higuchi 1996), les réseaux RBF (Whitehead et Choate 1996) ou des réseaux récurrents (Angeline et al. 1994). Dans cette dernière référence, l'algorithme GNARL ("GeNeralized Acquisition of Recurrent Links") est présenté comme un algorithme évolutionniste qui construit le réseau et effectue son apprentissage structurel et paramétrique. Parmi les applications présentées, le « problème de la fourmi » semble être résolu avec élégance et efficacité (mais rappelons que ce type d'application est propice à ce type d'approche).

Pour la construction des réseaux neuronaux, la supériorité d'un algorithme évolutionniste (l'algorithme EP, "Evolutionnary Programming") sur les algorithmes génétiques est observée grâce à plusieurs applications telles que le problème de la parité, des problèmes de diagnostics en médecine, ou encore celui de la prédiction de la série temporelle de Mackey-Glass (Yao et Liu 1997).

Pour conclure, la combinaison des réseaux de neurones avec les algorithmes évolutionnistes offre de nouvelles possibilités pour accroître la puissance des approches par apprentissage. Les algorithmes évolutionnistes restent cependant des méthodes gourmandes en calculs et demandent généralement des bases d'apprentissage très grandes.

d Des exemples

Le réseau le plus propice à un changement d'architecture est le réseau RBF. Les exemples les plus connus des réseaux à architecture évolutive sont donc des variantes de ce réseau de base. Parmi ceux-ci, citons les réseaux RCE (de Reilly, Cooper et Erlbaum (Reilly et al. 1982)) et GAL ("Grow And Learn") de (Alpaydin 1991), tous deux détaillés dans (Hérault et Jutten 1994).

Le réseau RCE, également connu sous le nom de NLS ("Nestor Learning System"), est un réseau incrémental très proche du réseau RBF. Son architecture comprend une couche constituée d'autant de cellules RBF que de classes à séparer, suivie d'une couche de cellules de décision. Il est surtout dédié à la classification supervisée.

Le réseau GAL s'apparente également à la méthode des *k* plus proches voisins. L'architecture de ce réseau est basée sur un réseau unidirectionnel suivi d'une couche WTA ("Winner Takes All"), elle-même suivie d'une couche de cellules de décision. Pendant son apprentissage, le réseau GAL alterne des phases d'apprentissage incrémental et des phases d'élagage.

L'algorithme RCE modélise les classes, alors que l'algorithme GAL modélise les frontières entre les algorithmes. Ceci a plusieurs conséquences : l'algorithme RCE est incapable de délivrer une réponse lorsqu'un vecteur n'appartenant à aucune région d'influence est présenté en entrée ; l'algorithme GAL, contrairement à l'algorithme RCE, n'aboutit jamais à une confusion car il n'admet pas de recouvrement des zones d'influence. Il n'y a qu'un prototype valide, celui qui est le plus proche.

Ces algorithmes semblent à première vue très simples. Ils ont néanmoins, en contrepartie, des inconvénients : le nombre de neurones peut augmenter très rapidement, et surtout le réseau a la fâcheuse tendance à apprendre par cœur. Ce dernier point signifie que ses capacités de généralisation sont pauvres.

Il existe également des cartes auto-organisatrices évolutives, telles celles décrites dans (Fritzke 1994) et dans (Choi et Park 1994), ou des réseaux évolutifs multicouches (Mézard et Nadal 1989).

L'architecture minimale, qui s'enrichit progressivement, la plus célèbre est la corrélation en cascade (en anglais "cascade-correlation" (Fahlman et Lebiere 1990)). Cette méthode d'apprentissage commence avec une structure minimale et ajoute progressivement au cours de l'apprentissage des neurones cachés au réseau jusqu'à ce que l'effet de ces nouveaux neurones soit bénéfique. Deux étapes

constituent la procédure itérative d'apprentissage. La première consiste à entraîner le réseau avec un apprentissage conventionnel, une variante de l'algorithme BP, l'algorithme Quickprop (Fahlman et Lebiere 1990). Cet algorithme est plus rapide que l'algorithme BP et est explicité dans (Jodouin 1994a). Au cours de la seconde étape, des neurones *candidats* sont rajoutés et leurs connexions avec les autres neurones sont ajustées afin de réduire autant que possible l'erreur résiduelle du réseau. Parmi les candidats, seul le neurone qui a le mieux réussi est conservé et intégré au réseau. Ses poids d'entrées sont figés et la première étape est répétée. Une application récente est fournie par un problème de classification dans (Lehtokangas 2000).

Cet apprentissage est plus rapide que la conventionnelle règle BP et est surtout intéressant par l'automatisation de l'évolution de ses paramètres structurels. L'inconvénient connu de cette méthode d'apprentissage est sa propension à aboutir à des capacités de généralisation moindres par rapport à l'apprentissage BP.

Le principe a été étendu aux réseaux récurrents (Fahlman et Lebiere 1991). Dans (Kirschning et al. 1995), les auteurs proposent même un réseau "Recurrent Cascade Correlation" (RCC) parallélisé pour une reconnaissance efficace de phonèmes. Dans (Elman 1990), l'auteur s'intéresse plus à l'importance du temps dans ce type d'architecture, notamment à comment améliorer le principe de la corrélation en cascade grâce aux boucles récurrentes.

2.4 Terminologie

Dans cette étude, nous emploierons indifféremment les termes de processus, de système, de tâche, de problème, ou encore de phénomène physique pour désigner le comportement que doit reproduire un réseau de neurones. Ils sont des modèles de comportement, que le processus d'apprentissage doit viser à reproduire, qu'il soit supervisé ou non.

Nous emploierons les termes de réseaux de neurones, de réseaux neuronal, de réseaux neuromimétiques, de réseaux connexionnistes, de systèmes ou de processus d'apprentissage pour désigner ces modèles. Ils seront utilisés avec un sens parfaitement identique. Ces modèles connexionnistes ont un point commun, leur capacité d'apprendre et de mémoriser par l'exemple.

Nous avons introduit, et nous continuerons à distinguer un processus d'adaptation du processus d'apprentissage. L'apprentissage est un processus à mémoire, capable de se rappeler les exemples assimilés, alors que l'adaptation est un apprentissage à mémoire courte, capable de ne considérer que les exemples les plus récents.

3 Conclusion

Cette partie, en introduisant les réseaux de neurones artificiels, permet de fournir une perception globale du problème et des mécanismes d'apprentissage. La principale caractéristique des réseaux neuromimétiques est leurs facultés d'adaptation et de mémorisation. Sans connaissance préalable, et au fur et à mesure que leur sont fournis des exemples, les réseaux de neurones se forgent un modèle de connaissance. Toute cette capacité d'assimilation des exemples réside dans la possibilité d'adapter les vecteurs des poids synaptiques $\mathbf{w}$.

Les réseaux de neurones sont donc une façon d'élaborer un modèle de connaissance, à partir d'une base de données. La mise au point de ce modèle de connaissance se fait par une phase d'apprentissage. Cet apprentissage dépend de l'architecture du réseau et d'une règle d'adaptation des poids des neurones :

- le modèle de réseau doit être adapté à la complexité de la tâche à apprendre, i.e., doit disposer de paramètres latents suffisants lui assurant un degré de flexibilité qui permet d'être à la hauteur de la complexité de la tâche ;
- la règle d'adaptation des poids synaptiques découle d'un critère d'erreur, sa minimisation itérative doit conduire à une solution optimale des poids.

Le problème de l'apprentissage à partir d'un jeu de données constituées de mesures bruitées, est confronté au dilemme biais/variance. Lorsqu'un réseau de neurones n'est pas assez complexe, il lui manque la flexibilité nécessaire pour reproduire le comportement, et l'estimation en ressort biaisée. D'un autre côté, lorsque l'apprentissage s'effectue « trop bien », les capacités de généralisation se trouvent détériorées par un phénomène de sur-apprentissage. La complexité d'un réseau se doit d'être ajustée à la tâche à apprendre. Le choix d'une procédure d'apprentissage, et de l'architecture d'un réseau, sont de ce fait fortement dépendant de la tâche à apprendre.

Si l'apprentissage rend les réseaux de neurones fort attractifs pour bon nombre d'applications, ils comportent néanmoins certains inconvénients comme un manque de connaissances cohérentes et globales, des conflits d'objectifs (généraliser et distinguer), et quelques difficultés à tenir compte des environnements non stationnaires.

Les recherches récentes dans le domaines des réseaux de neurones vont dans le bon sens, avec la prise en compte d'un point de vue statistique – explicité dans le Chapitre 2 – et le développement d'architectures neuronales modulaires permettant d'intégrer un certain nombre de connaissances a priori. L'apprentissage modulaire est abordée et synthétisé dans le Chapitre 3.

Bibliographie

Akhmetov, D. F., Dote, Y., et Ovaska, S. J. (2001). "Fuzzy Neural Network with General Parameter Adaptation for Modeling of Nonlinear Time-Series." *IEEE Transactions on Neural Networks*, 12(1), 148-152.

Albus, J. S. (1975). "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (Cmac)." *Journal of Dynamic Systems, Measurement and Control*, 37(3), 220-227.

Allen, P., Timcenko, A., Yoshimi, B., et Michelman, P. (1993). "Automated Tracking and Grasping of a Moving Object with a Robotic Hand-Eye System." *IEEE Transactions on Robotics and Automation*, 9(2), 152-165.

Allen, P. K., Yoshimi, B., et Timcenko, A. (1991). "Real-Time Visual Servoing." *IEEE International Conference on Robotics and Automation*, Sacramento, C.A., 851-856.

Aloimonos, Y. (1993). "Active Perception." Lawrence Erlbaum Associates, 292.

Alpaydin, E. (1991). "Gal : Networks That Grow When They Learn and Shrink When They Forget." Technical report *TR 91-032*, International Computer Science Institute, Berkeley, CA. ftp://ftp.gmd.de/documents/neuroprose/tr-91-032.ps.gz

Alpaydin, E. (1993). "Multiple Networks for Function Learning." *IEEE International Conference on Neural Networks*, San Francisco, CA, 9-14.

Amaral, J. N., Tumer, K., et Ghosh, J. (1995). "Designing Genetic Algorithms for the State Assignment Problem." *IEEE Transactions on Systems, Man and Cybernetics*, 25(4), 687-694.

Amy, B. (1996). "Recherches Et Perspectives Dans Le Domaine Des Réseaux Connexionnistes." Rapport DRET *0262011484*, Laboratoire Leibniz-Imag, Grenoble, France. ftp://ftp.imag.fr/pub/labo-LEIBNIZ/RESEAUX-D-AUTOMATES/amy.nn.ps.tar.gz

Anderson, B. D. O., et Moore, J. B. (1979). *Optimal Filtering*, Prentice-Hall, Englewood Cliffs.

Angeline, P. J., Saunders, G. M., et Pollack, J. B. (1994). "An Evolutionary Algorithm That Constructs Recurrent Neural Networks." *IEEE Transactions on Neural Networks*, 5(1), 54-65.

Anouar, F., Badran, F., et Thiria, S. (1998). "Probabilistic Self Organizing Map and Radial Basis Function." *Neurocomputing*, 20, 83-96.

Ansley, C. F., et Kohn, R. (1985). "Estimation, Filtering and Smoothing in State Space Models with Incompletely Specified Initial Conditions." *Annals of Statistics*, 13, 1286-1316.

Astrom, K. J., et Wittenmark, B. (1995). *Adaptive Control*, Addison-Wesley, Reading, Massachusetts.

Auda, G., et Kamel, M. (1998). "Modular Neural Network Classifiers - a Comparative Study." *Journal of Intelligent & Robotic Systems*, 21(2), 117-129.

Aussem, A., et Murtagh, F. (1997). "Combining Neural Network Forecasts on Wavelet-Transformed Time Series." *Connection Science*, 9(1), 113-121.

Bar-Shalom, Y., et Birmiwal, K. (1983). "Consistency and Robustness of Pdaf for Target Tracking in Cluttered Environments." *Automatica*, 19(4), 431-437.

Basseville, M. (1988). "Detecting Changes in Signals and Systems - a Survey." *Automatica*, 24(3), 309-326.

Battiti, R., et Colla, A. M. (1994). "Democracy in Neural Nets: Voting Schemes for Classification." *Neural Networks*, 7(4), 691-707.

Bauer, H.-U., Herrmann, M., et Villmann, T. (1999). "Neural Maps and Topographic Vector Quantization." *Neural Networks*, 12(4-5), 659-676.

Bauer, H.-U., et Pawelzik, K. R. (1992). "Quantifying the Neighbourhood Preservation of Self-Organizing Feature Maps." *IEEE Transactions on neural networks*, 3(3), 570-579.

Bauer, H.-U., et Villmann, T. (1997). "Growing a Hypercubical Output Space in a Self-Organizing Feature Map." *IEEE Transactions on Neural Networks*, 8(2), 218-226.

Belanger, P. R. (1974). "Estimation of Noise Covariance Matrices for a Linear Time - Varying Stochastic Process." *Automatica*, 10(3), 267-275.

Bell, W., et Hillmer, S. (1991). "Initializing the Kalman Filter for Nonstationary Time Series Models." *Journal of Time Series Analysis*, 12(4), 283-300.

Benediktsson, J. A., et Swain, P. H. (1992). "Consensus Theoretic Classification Methods." *IEEE Transactions on Systems Man and Cybernetics*, 22(4), 688-704.

Bennani, Y. (1995). "A Modular and Hybrid Connectionist System for Speaker Identification." *Neural Computation*, 7(4), 791-798.

Bennani, Y., et Bossaert, F. (2001). "Modular Connectionist Modelling and Classification Approaches for Local Diagnosis in Telecommunication Traffic Management." *International Journal of Computational Intelligence and Applications*, 1(1), 53-70.

Bennani, Y., et Gallinari, P. (1992). "Une Architecture Connexionniste Modulaire Pour L'identification Automatique Du Locuteur." Rapport de recherche *739*, Laboratoire de Recherche en Informatique, Orsay (France).

Bensalah, F. (1996). "Estimation Du Mouvement Par Vision Active," Thèse de Doctorat, IRISA, Université de Rennes 1, Rennes.

Bensalah, F., et Chaumette, F. (1994). "Détection De Rupture De Modèle Appliquée À L'asservissement Visuel." Rapport de Recherche *RR-2425*, INRIA, Rennes. ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RR/RR-2425.pdf

Bishop, C. M. (1994). "Mixture Density Networks." *NCRG/94/004*, Neural Computing Research Group, Aston University, Birmingham. http://www.ncrg.aston.ac.uk

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford.

Bishop, C. M., Svensen, M., et Williams, C. K. I. (1998). "Gtm: The Generative Topographic Mapping." *Neural Computation*, 10(1), 215-234.

Boers, E. J. W., Kuiper, H., Happel, B. L. M., et Sprinkhuizen-Kuyper, I. G. (1993). "Designing Modular Artificial Neural Networks." *Computing Science in the Netherlands CSN'93*, 87-96.

Bonabeau, E., et Theraulaz, G. (1994). "Intelligence Collective." Hermès, Paris, 288.

Bottou, L. (1997). "La Mise En Oeuvre Des Idées De Vladimir N. Vapnik." Statistiques Et Méthodes Neuronales, S. Thiria, Y. Lechevallier, O. Gascuel, and S. Canu, eds., Dunod, Paris, 262-274.

Bouchard, M. (2001). "New Recursive-Least-Squares Algorithms for Nonlinear Active Control of Sound and Vibration Using Neural Networks." *IEEE Transactions on Neural Networks*, 12(1), 135-147.

Bouthemy, P. (1989). "A Maximum-Likelihood Framework for Determining Moving Edges." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5), 499-511.

Bouton, C., et Pagès, G. (1993). "Self-Organization of the One-Dimensional Kohonen Algorithm with Non-Uniformly Distributed Stimuli." *Stochastic Processes and their Applications*, 47, 249-274.

Braitenberg, V. (1991). *Véhicules : Expériences En Psychologie Synthétique*, Presses Polytechniques et Universitaires Romandes, Lausanne, Suisse.

Breiman, L. (1996a). "Bagging Predictors." *Machine Learning*, 24(2), 132-140.

Breiman, L. (1996b). "Bias, Variance, and Arcing Classifiers." *TR 460*, Statistics Department, University of California. http://www.stat.berkeley.edu/tech-reports/460.ps.Z

Breiman, L., Friedman, J. H., Olshen, R. A., et Stone, C. J. (1984). *Classification and Regression Trees*, Wadsworth and Brooks/Cole, Monterey, CA.

Breton, S. (1996). "Sceno : Un Logiciel Scénographique Orienté Objet De Simulation De Scène Robotique - Guide Utilisateur." Rapport technique *EEA-TROP-TR-96-05*, Université de Haute-Alsace, Mulhouse.

Breton, S. (1999). "Une Approche Neuronale Du Contrôle Robotique Utilisant La Vision Binoculaire Par Reconstruction Tridimensionelle," Thèse de Doctorat, Université de Haute Alsace, Mulhouse.

Bridle, J. (1990). "Training Stochastic Model Recognition Algorithms as Networks Can Lead to Maximum Mutual Information Estimation of Parameters." *Advances in Neural Information Processing Systems*, San Mateo, CA, 211-217.

Brown, R. G., et Hwang, P. Y. C. (1996). *Introduction to Random Signals and Applied Kalman Filtering : With Matlab Exercises and Solutions*, John Wiley & Sons, New York.

Bryson, A. E. (1978). "Kalman Filter Divergence." *AIAA Journal of Guidance and Control*, 71-79.

Buessler, J.-L. (1999). "Architectures Neuro-Mimétiques Modulaires, Application À L'asservissement Visuel De Systèmes Robotiques," Thèse de Doctorat, Université de Haute-Alsace, Mulhouse.

Buessler, J. L., Kara, R., Wira, P., Kihl, H., et Urban, J. P. (1999). "Multiple Self-Organizing Maps to Facilitate the Learning of Visuo-Motor Correlations." *IEEE International Conference on Systems Man and Cybernetics*, Tokyo, Japan, 470-475.

Buessler, J.-L., Urban, J.-P., Kihl, H., et Gresser, J. (1996). "A Neural Module for the Visual Servoing of a Robotic Manipulator." *International Conference on Computational Engineering in Systems Applications (CESA'96)*, Lille, France, 246-251.

Bullock, D., et Grossberg, S. (1987). "A Neural Network Architecture for Automatic Trajectory Formation and Coordination of Multiple Effectors During Variable-Speed Arm Movements." *IEEE International Conference on Neural Networks*, San Diego, CA, 559-567.

Cacciatore, T. W., et Nowlan, S. J. (1994). "Mixtures of Controllers for Jump Linear and Non-Linear Plants." *Advances in Neural Information Processing Systems*, San Mateo, CA, 719-726.

Caelli, T., Guan, L., et Wen, W. (1999). "Modularity in Neural Computing." *Proceedings of the IEEE, Special Issue on Computational Intelligence*, 87(9), 1497-1518.

Campos, M. M., et Carpenter, G. A. (2001). "S-Tree: Self-Organizing Trees for Data Clustering and Online Vector Quantization." *Neural Networks*, 14(4-5), 505-525.

Castillo, E. (1998). "Functional Networks." *Neural Processing Letters*, 7(3), 151-159.

Castillo, E., Gutiérrez, J. M., Cobo, A., et Castillo, C. (2000). "A Minimax Method for Learning Functional Networks." *Neural Processing Letters*, 11(1), 39-49.

Catlin, D. E. (1993). "Fisher Initialization in the Presence of Ill-Conditioned Measurements." Approximate Kalman Filtering, G. Chen, ed., World Scientific Publishing, Singapore/New Jersey, 23-38.

Cauwenberghs, G. (1996). "Adaptation, Learning and Storage in Analog Vlsi." *9th Annual IEEE International ASIC Conference*, Rochester NY, 273-278.

Cesmeli, E., et Wang, D. L. (2001). "Texture Segmentation Using Gaussian-Markov Random Fields and Neural Oscillator Networks." *IEEE Transactions on Neural Networks*, 12(2), 394-404.

Cexus, J.-C. (2001). "Réseaux Neuronaux Linéaires Adaptatifs : Application À La Commande D'une Tête Robotique." Rapport Technique, Université de Haute Alsace, Mulhouse.

Chaer, W. S., et Bishop, R. H. (1995). "Adaptive Kalman Filtering with Genetic Algorithms." *Advances in the Astronautical Sciences*, 89, 141-155.

Chaer, W. S., Bishop, R. H., et Ghosh, J. (1997). "A Mixture-of-Experts Framework for Adaptive Kalman Filtering." *IEEE Transactions on Systems, Man, and Cybernetics, Part B : Cybernetics*, 27(3), 452-464.

Chaer, W. S., Bishop, R. H., et Ghosh, J. (1998). "Hierarchical Adaptive Kalman Filtering for Interplanetary Orbit Determination." *IEEE Transactions on Aerospace and Electronic Systems*, 34(3), 883-896.

Chappell, G. J., et Taylor, J. G. (1993). "The Temporal Kohonen Map." *Neural Networks*, 6(3), 441-445.

Chaumette, F. (1990). "La Relation Vision-Commande: Théorie Et Application À Des Tâches Robotiques," Thèse de Doctorat, Université de Rennes 1, IRISA, Rennes.

Chaumette, F. (1998). "De La Perception À L'action : L'asservissement Visuel, De L'action À La Perception : La Vision Active." Habilitation à diriger des recherches, Université de Rennes 1, Institut de Formation Supérieure en Informatique et en communication, Rennes. ftp://ftp.irisa.fr/techreports/2001/

Chellapilla, K., et Fogel, D. B. (1999). "Evolution, Neural Networks, Games, and Intelligence." *Proceedings of the IEEE, Special Issue on Computational Intelligence*, 87(9), 1471-1496.

Chen, B. S., et Peng, S. C. (1993). "Robust Stability Analysis of Kalman Filter under Parametric and Noise Uncertainties." Approximate Kalman Filtering, G. Chen, ed., World Scientific Publishing, Singapore/New Jersey, 179-192.

Chen, G. (1993). "Approximate Kalman Filtering." Series in Approximations & Decompositions, C. K. Chui, ed., World Scientific Publishing, Singapore/New Jersey, 240.

Chen, R., et Liu, J. S. (2000). "Mixture Kalman Filters." *Journal of the Statistical Society, Series B*, 62, 493-508.

Chinrungrueng, C., et Séquin, C. H. (1995). "Optimal Adaptive K-Means Algorithm with Dynamic Adjustment of Learning Rate." *IEEE Transactions on Neural Networks*, 6(1), 157-169.

Chiueh, T.-D., Tang, T.-T., et Chen, L.-G. (1994). "Vector Quantization Using Tree-Structured Self-Organizing Feature Maps." *IEEE Journal on Selected Areas in Communications*, 12(9), 1594-1599.

Cho, S.-B. (1997). "Combining Modular Neural Networks Developed by Evolutionary Algorithm." *IEEE International Conference on Evolutionary Computation*, 647-650.

Cho, S.-B., et Kim, J. H. (1995). "Multiple Network Fusion Using Fuzzy Logic." *IEEE Transactions on Neural Networks*, 6(2), 497-501.

Choi, D.-I., et Park, S.-H. (1994). "Self-Creating and Organizing Neural Networks." *IEEE Transactions on Neural Networks*, 5(4), 561-575.

Churchland, P. S., et Sejnowski, T. J. (1992). *The Computational Brain*, MIT Press.

Clarke, D. W., Mohtadi, C., et Tuffs, P. S. (1987). "Generalised Predictive Control. Part 1 : The Basic Algorithm, Part 2: Extensions and Interpretations." *Automatica*, 23(2), 137-160.

Clemen, R. T., et Winkler, R. L. (1985). "Limits for the Precision and Value of Information from Dependent Sources." *Operations Research*, 33, 427-442.

Coiffet, P. (1992). *La Robotique : Principe Et Applications*, Hermès, Paris.

Corke, P. I. (1996). "A Robotics Toolbox for Matlab." *IEEE Robotics and Automation Magazine*, 3(1), 24-32.
http://www.cat.csiro.au/cmst/staff/pic/robot/

Corne, C. (1999). "Parallélisation De Réseaux De Neurones Sur Architecture Distribuée," Thèse de Doctorat, Université de Haute-Alsace, Mulhouse.

Cottrell, M., et Fort, J. C. (1986). "A Stochastic Model of Retinotopy: A Self-Organizing Process." *Biological Cybernetics*, 53, 405-411.

Cottrell, M., et Fort, J. C. (1987). "Etude D'un Processus D'auto-Organisation." *Annales de l'Institut Poincarré*, 23(1), 1-20.

Cottrell, M., Fort, J. C., et Pagès, G. (1994). "Two or Three Things That We Know About the Kohonen Algorithm." *European Symposium on Artificial Neural Networks (ESANN'94)*, Brussels, Belgium, 235-244.

de Bodt, E., et Cottrell, M. (2000). "Bootstrapping Self-Organizing Maps to Assess the Statistical Significance of Local Proximity." *8th European Symposium on Artificial Neural Networks*, Bruges, Belgium, 245-254.

De Francesco, M. (1994). "Functional Networks: A New Computational Framework for the Specification Simulation and Algebraic Manipulatio of Modular Systems," PhD Thesis, Facultée des Sciences de l'Université de Genève, Genève.

De Nicolao, G., et Ferrari-Trecate, G. (2001). "Regularization Networks: Fast Weight Calculation Via Kalman Filtering." *IEEE Transactions on Neural Networks*, 12(2), 228-235.

Deguchi, K. (2000). "A Direct Interpretation of Dynamic Images with Camera and Object Motions for Vision Guided Robot Control." *International Journal of Computer Vision*, 37(1), 7-20.

Deguchi, K., et Noguchi, T. (1996). "Visual Servoing Using Eigenspace Method and Dynamic Calculation of Interaction Matrices." *International Conference on Pattern Recognition (ICPR'96)*, Vienna, Austria.

Deguchi, K., et Takahashi, I. (1999a). "Image-Based Control of Robot Ego-Motion and Target Object Motions in the Scene by Eigen Space Method." *IMEKO-XV World Congress*, Osaka, 65-72.

Deguchi, K., et Takahashi, I. (1999b). "Image-Based Simultaneous Control of Robot and Target Object Motions by Direct-Image-Interpretation Method." *International Conference on Intelligent Robotics and Systems (IROS'99)*, Kyongju, Korea, 375-380.

Deguchi, K., et Yonaiyama, K. (2000). "Real-Time 3d Shape Recognition of High Speed Moving Object by Fixation Point Tracking with Active Camera." *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2000)*, Takamatsu, Japan, 596-601.

Dempster, A. P., Laird, N. M., et Rubin, D. B. (1977). "Maximum Likelihood from Incomplete Data Via Em Algorithm." *Journal of the Royal Statistical Society*, B 39, 1-33.

Doucet, A., Freitas, J. F. G. d., et Gordon, N. J. (2001). "Sequential Monte Carlo Methods in Practice." Springer-Verlag, New York, 620.

Drolet, L., Michaud, F., et Côté, J. (2000). "Adaptable Sensor Fusion Using Multiple Kalman Filters." *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, Takamatsu, Japan, 1434-1439.

Drucker, H., Cortes, C., Jackel, L., Le Cun, Y., et Vapnik, V. (1994). "Boosting and Other Ensemble Methods." *Neural Computation*, 6(6), 1289-1301.

Duch, W., et Jankowski, N. (2001). "Transfert Functions: Hidden Possibilities for Better Neural Networks." *9th European Symposium on Artificial Neural Networks (ESANN'2001)*, Bruges (Belgium), 81-94.

Eaton, P. H., Prokhorov, D. V., et D. C. Wunsch, I. (2000). "Neurocontroller Alternatives for "Fuzzy" Ball-and-Beam Systems with Nonuniform Nonlinear Friction." *IEEE Transactions on Neural Networks*, 11(2), 423-435.

Elman, J. L. (1990). "Finding Structure in Time." *Cognitive Science*, 14, 179-211.

Erwin, E., Obermayer, K., et Schulten, K. (1992a). "Self-Organizing Maps: Ordering, Convergence Properties and Energy Functions." *Biological Cybernetics*, 67(1), 47-55.

Erwin, E., Obermayer, K., et Schulten, K. (1992b). "Self-Organizing Maps: Stationary States, Metastability and Convergence Rate." *Biological Cybernetics*, 67(1), 35-45.

Espiau, B., Chaumette, F., et Rives, P. (1992). "A New Approach to Visual Servoing in Robotics." *IEEE Transactions on Robotics and Automation*, 8(3), 313-326.

Euliano, N., et Principe, J. (1998). "Temporal Plasticity in Self-Organizing Networks." *International Joint Conference on Neural Networks (IJCNN '98)*, 1063-1067.

Euliano, N. R., et Principe, J. C. (1996). "Spatio-Temporal Self-Organizing Feature Maps." *International Conference on Neural Networks*, 1900-1905.

Fagin, S. L. (1964). "Recursive Linear Regression Theory, Optimal Filter Theory, and Error Analysis Optimal System." *IEEE International Convention Record*, 12, 216-240.

Fahlman, S. E., et Lebiere, C. (1990). "The Cascade-Correlation Learning Architecture." *Advances in Neural Information Processing Systems*, Los Altos, Ca., 524-532.

Fahlman, S. E., et Lebiere, C. (1991). "A Recurrent Cascade-Correlation Learning Architecture." *Advances in Neural Information Processing Systems*, 190-196.

Faugeras, O. (1993). *Three-Dimensional Computer Vision*, MIT Press.

Faugeras, O., Lustman, F., et Toscani, G. (1987). "Calcul Du Mouvement Et De La Structure À Partir De Points Et De Droites." Rapport de Recherche *661*, INRIA.

Feddema, J. T., et Lee, C. S. G. (1990). "Adaptive Image Feature Prediction and Control for Visual Tracking with a Hand-Eye Coordinated Camera." *IEEE Transactions on Systems, Man and Cybernetics*, 20(5), 1172-1183.

Fiordaliso, A. (1999). *Systèmes Flous Et Prévision Des Séries Temporelles*, Hermès Science, Paris.

Fiori, S., et Bucciarelli, P. (2001). "Probability Density Estimation Using Adaptive Activation Function Neurons." *Neural Processing Letters*, 13(1), 31-42.

Fisher, R. A. (1912). "On an Absolute Criterion for Fitting Frequency Curves." *Messenger of Math.*, 41, 155.

Fitzgerald, R. J. (1967). "Error Divergence in Optimal Filtering Problems." *2nd IFAC Symp. Automatic Control in Space*, Vienna, Austria.

Fitzgerald, R. J. (1971). "Divergence of the Kalman Filter." *IEEE Transactions on Automatic Control*, AC-1(6), 736-747.

Fleuret, F., et Brunet, E. (2000). "Dea: An Architecture for Goal Planning and Classification." *Neural Computation*, 12(9), 1987-2008.

Fogel, D., Fogel, L., et Porto, V. (1990). "Evolving Neural Networks." *Biological Cybernetics*, 63, 487-493.

Fort, J. C., et Pagès, G. (1995). "On the A.S. Convergence of the Kohonen Algorithm with a General Neighborhood Function." *The Annals of Applied Probabilities*, 5(4), 1177-1216.

Franklin, G. F., J.D. Powell, et Workman, M. L. (1998). *Digital Control of Dynamic Systems*, Addison-Wesley, Menlo Park, California.

Freitas, J. F. G. d. (1999). "Bayesian Methods for Neural Networks," PhD Thesis, Trinity College, University of Cambridge, Cambridge.

Freitas, J. F. G. d., Niranjan, M., et Gee, A. H. (1997). "Hierarchical Bayesian-Kalman Models for Regularisation and Ard in Sequential Learning." *Technical Report CUED/F-INFENG/TR 307*, Cambridge University Department of Engineering.

Freitas, J. F. G. d., Niranjan, M., et Gee, A. H. (1998a). "The Em Algorithm and Neural Networks for Nonlinear State Space Estimation." *Technical report CUED/F-INFENG/TR 313*, Cambridge University Department of Engineering.

Freitas, J. F. G. d., Niranjan, M., et Gee, A. H. (1998b). "Nonlinear State Space Learning with Em and Neural Networks." *IEEE Workshop on Neural Networks for Signal Processing (NNSP'98)*, Cambridge, England, 254-263.

Freitas, J. F. G. d., Niranjan, M., Gee, A. H., et Doucet, A. (2000). "Sequential Monte Carlo Methods to Train Neural Network Models." *Neural Computation*, 12(4), 955-993.

Friedman, J. H. (1991). "Multivariate Adaptive Regression Splines." *Annals of Statistics*, 19(1), 1-67.

Fritzke, B. (1994). "Growing Cell Structures - a Self-Organizing Network for Unsupervised and Supervised Learning." *Neural Networks*, 7(9), 1441-1460.

Fukuda, T., Michelini, R., Potkonjak, V., Tzafestas, S., Valavanis, K., et Vukobratovic, M. (2001). "How Far Away Is "Artificial Man"?" *IEEE Robotics and Automation Magazine*, 7(1), 66-73.

Gallinari, P. (1995). "Modular Neural Net Systems, Training Of." The Handbook of Brain Theory and Neural Networks, M. A. Arbib, ed., MIT Press, Cambridge, Mass., 438-441.

Gan, Q., et Harris, C. J. (2001). "A Hybrid Learning Scheme Combining Em and Masmod Algorithms for Fuzzy Local Linearization Modeling." *IEEE Transactions on Neural Networks*, 12(1), 43-53.

Gangloff, J. (1999). "Asservissements Visuels Rapides D'un Robot Manipulateur À Six Degrés De Liberté - Suivi De Cible Et De Profilé," Thèse de Doctorat, Université Louis Pasteur, Ecole Nationale supérieure de Physique de Strasbourg, Strasbourg.

Gangloff, J., Mathelin, M. d., et Abba, G. (1999). "High Performance 6 Dof Visual Servoing Using Generalized Predictive Control." *5th European Control Conference (ECC'99)*, Karlsruhe, Germany.

Gascuel, O. (1997). "La Dimension De Vapnik-Chervonenkis, Application Aux Réseaux De Neurones." Statistiques Et Méthodes Neuronales, S. Thiria, Y. Lechevallier, O. Gascuel, and S. Canu, eds., Dunod, Paris, 244-261.

Ge, S. S., Lee, T. H., et Harris, C. J. (1998). "Adaptive Neural Network Control of Robotic Manipulators." Robotics and Automated Systems, T. M. Husband, ed., World Scientific, Singapore, 396.

Gelb, A. (1974). *Applied Optimal Estimation*, The MIT Press, Cambridge.

Geman, S., Bienenstock, E., et Doursat, R. (1992). "Neural Networks and the Bias/Variance Dilemma." *Neural Computation*, 4(1), 1-58.

Ghahramani, Z., et Hinton, G. E. (1996). "The Em Algorithm for Mixtures of Factor Analysers." Technical Report *CGR-TR-96-1*, Department of Computer Science, University of Toronto, Toronto, Ontario. http://www.gatsby.ucl.ac.uk/hinton/absps/tr-96-1.ps.gz

Ghahramani, Z., et Wolpert, D. M. (1997). "Modular Decomposition in Visuomotor Learning." *Nature*, 386(6623), 392-395.

Gómez, V., et Maravall, A. (1993). "Initializing the Kalman Filter with Incompletely Specified Initial Conditions." Approximate Kalman Filtering, G. Chen, ed., World Scientific Publishing, Singapore/New Jersey, 39-62.

Gómez, V., et Maravall, A. (1994). "Estimation, Prediction and Interpolation for Nonstationary Series with the Kalman Filter." *Journal of the American Statistical Association*, 89, 611-624.

Gomi, H., et Kawato, M. (1993). "Recognition of Manipulated Objetcs by Motor Learning with Modular Architecture Networks." *Neural Networks*, 6(4), 485-497.

Gordon, N., Salmond, D., et Ewing, C. (1995). "Bayesian State Estimation for Tracking and Guidance Using the Bootstrap Filter." *Journal of Guidance Control and Dynamics*, 18(6), 1434-1443.

Gourdeau, R. (1997). "Object-Oriented Programming for Robotic Manipulator Simulation." École Polytechnique de Montréal, Montréal. http://www.cours.polymtl.ca/roboop/docs/robot.html

Guan, L., Anderson, J. A., et Sutton, J. P. (1997). "A Network of Networks Processing Model for Image Regularization." *IEEE Transactions on Neural Networks*, 8(1), 169-174.

Güler, M. (2001). "A Model with an Intrinsic Property of Learning Higher Order Correlations." *Neural Networks*, 14(4-5), 495-504.

Gullapalli, V., Grupen, R., et Barto, A. G. (1992). "Learning Reactive Admittance Control." *International Conference on Robotics and Automation*, Los Alamitos, CA, 1475-1480.

Haese, K. (1998). "Self-Organizing Feature Maps with Self-Adjusting Learning Parameters." *IEEE Transactions on Neural Networks*, 9(6), 1270-1278.

Haese, K. (1999). "Kalman Filter Implementation of Self-Organizing Feature Maps." *Neural Computation*, 11(5), 1211-1233.

Haese, K., et Goodhill, G. J. (2001). "Auto-Som: Recursive Parameter Estimation for Guidance of Self-Organizing Feature Maps." *Neural Computation*, 13(3), 595-619.

Hagan, M. T., Demuth, H. B., et Beale, M. (1995). *Neural Network Design*, PWS Publishing Company.

Hanlon, P. D., et Maybeck, P. S. (1998). "Interrelationship of Single-Filter and Multiple-Model Adaptive Algorithms." *IEEE Transactions on Aerospace and Electronic Systems*, 34(3), 934-946.

Harris, C. (1992). "Tracking with Rigid Models." Active Vision, A. Blake and A. Yuille, eds., MIT Press, Cambridge, MA.

Hashem, S. (1997). "Optimal Linear Combinations of Neural Networks." *Neural Networks*, 10(4), 599-614.

Hashemipour, H. R., Roy, S., et Laub, A. J. (1988). "Decentralized Structures for Parallel Kalman Filtering." *IEEE Transactions on Automatic Control*, 33(1), 88-93.

Hashimoto, H., Takashi, K., Kudou, M., et Harashima, F. (1992). "Self-Organizing Visual Servo System Based on Neural Networks." *IEEE Control Systems Magazine*, 12(2), 31-36.

Hashimoto, K. (1993). "Visual Servoing." Robotics and Automated Systems, T. M. Husband, ed., World Scientific, Singapore, 372.

Haykin, S. (1994). *Neural Networks : A Comprehensive Foundation*, Macmillan College Publishing Company, Inc., New York.

Haykin, S. (1999). *Neural Networks : A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, N.J.

Haykin, S. (2001). "Kalman Filtering and Neural Networks." John Wiley & Sons, 320.

Haykin, S. S. (1996). *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, New Jersey.

Hecht-Nielsen, R. (1987). "Counterpropagation Networks." *Applied Optics*, 26(23), 4979-4984.

Hecht-Nielsen, R. (1990). *Neurocomputing*, Addison-Wesley, Reading, MA.

Hérault, J., et Jutten, C. (1994). *Réseaux Neuronaux Et Traitement Du Signal*, Hermès, Paris.

Hermann, G. (2000). "Asservissement Visuel D'une Tête Robotique Avec Un Filtre De Kalman." Rapport Technique *EEA-TROP-TR-00-01*, Université de Haute Alsace, Mulhouse.

Herrmann, M. (1995). "Self-Organizing Feature Maps with Self-Organizing Neighborhood Widths." *IEEE International Conference on Neural Networks (ICNN'95)*, Piscataway, NJ, 2998-3003.

Herrmann, M., Bauer, H.-U., et Villmann, T. (1997). "Topology Preservation in Neural Maps." *European Symposium on Artificial Neural Networks (ESANN'97)*, 205-210.

Heskes, T. (1998). "Bias/Variance Decompositions for Likelihood-Based Estimators." *Neural Computation*, 10(6), 1425-1433.

Heskes, T., et Kappen, B. (1991). "Neural Networks Learning in Changing Environment." *IEEE International Conference*, 823-828.

Hillmer, S. C., et Bell, W. R. (1987). "Initializing the Kalman Filter for Nonstationary Time Series Models." Research Report *RR87/33*, U.S. Census Bureau, Statistical Research Division, Washington.
http://www.census.gov/srd/papers/pdf/rr87-33.pdf

Hofmann, T., et Puzicha, J. (1998). "Statistical Models for Co-Occurrence Data." *A.I. Memo No. 1625, C.B.C.L Memo No. 159*, Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Horio, K., et Yamakawa, T. (2001). "Feedback Self-Organizing Map and Its Application to Spatio-Temporal Pattern Classification." *International Journal of Computational Intelligence and Applications*, 1(1), 1-18.

Hrycenko, N. (2000). "Etude D'une Structure Intelligente De Commande Adaptative : Le Mélange D'experts." Rapport Technique *EEA-TROP-TR-00-02*, Université de Haute Alsace, Mulhouse.

Hutchinson, S., Hager, G., et Corke, P. (1996). "A Tutorial on Visual Servo Control." *IEEE Transactions on Robotics and Automation*, 12(5), 651-670.

IEEE. (1996). "Ieee Transactions on Robotics and Automation." *Special Section on Vision-Based Control of Robot Manipulators*, 12(5).

IEEE. (1999). "Ieee Transactions on Neural Networks." *Special Issue on VC Learning Theory and its Applications*, 10(5).

Igelnik, B., Tabib-Azar, M., et LeClair, S. R. (2001). "A Net with Complex Weights." *IEEE Transactions on Neural Networks*, 12(2), 236-249.

Iiguni, Y., Sakai, H., et Tokumaru, H. (1992). "A Real-Time Learning Algorithm for a Multilayered Neural Network Based on the Extended Kalman Filter." *IEEE Transactions on Signal processing*, 40(4), 959-966.

Ishikawa, M. (1996). "Structural Learning with Forgetting." *Neural Networks*, 9(3), 509-521.

Jacobs, R. A. (1995). "Methode for Combining Experts' Probability Assessments." *Neural Computation*, 7(5), 867-888.

Jacobs, R. A. (1997). "Bias/Variance Analyses of Mixtures-of-Experts Architectures." *Neural Computation*, 9(2), 369-383.

Jacobs, R. A., et Jordan, M. I. (1991a). "A Competitive Modular Connectionist Architecture." *Neural Information Processing Systems*, San Mateo, CA, 767-773.

Jacobs, R. A., et Jordan, M. I. (1991b). "A Modular Connectionist Architecture for Learning Piecewise Control Strategies." *American Control Conference*, Boston, Massachusetts, 1597-1602.

Jacobs, R. A., et Jordan, M. I. (1993). "Learning Piecewise Control Strategies in a Modular Neural Network Architecture." *IEEE Transactions on Systems, Man, and Cybernetics*, 23(2), 337-345.

Jacobs, R. A., Jordan, M. I., et Barto, A. G. (1991a). "Task Decomposition through Competition in a Modular Connectionist Architecture - the What and Where Vision Tasks." *Cognitive Science*, 15(2), 219-250.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., et Hinton, G. E. (1991b). "Adaptive Mixtures of Local Experts." *Neural Computation*, 3(1), 79-87.

Jacobs, R. A., Peng, F. C., et Tanner, M. A. (1997). "A Bayesian Approach to Model Selection in Hierarchical Mixtures-of-Experts Architectures." *Neural Networks*, 10(2), 231-241.

Jacquet, W. (2001). "Paradigmes Connexinnistes Pour L'apprentissage De Fonctions Robotiques : Application Aux Problèmes D'asservissement Visuel De Bras Manipulateur," Thèse de Doctorat, Université de Haute-Alsace, Mulhouse.

Jägersand, M. (1996). "Visual Servoing Using Trust Region Methods and Estimation of the Full Coupled Visual-Motor Jacobian." *International Conference on Applications of Robotics and Control*, 105-108.

Jain, A. K., Mao, J., et Mohiuddin, K. M. (1996). "Artificial Neural Networks: A Tutorial." *IEEE Computer Magazine*, 29(3), 56-63.

Jankowski, N., et Duch, W. (2001). "Optimal Transfert Function Neural Networks." *9th European Symposium on Artificial Neural Networks (ESANN'2001)*, Bruges (Belgium), 101-106.

Jazwinsky, A. H. (1968). "Limited Memory Optimal Filtering." *IEEE Transactions on Automatic Control*, AC-13(5), 558-563.

Jazwinsky, A. H. (1969). "Adaptive Filtering." *Automatica*, 5(4), 475-485.

Jin, Y., et Sendhoff, B. (1999). "Knowledge Incorporation into Neural Networks from Fuzzy Rules." *Neural Processing Letters*, 10(3), 231-242.

Jodouin, J. F. (1994a). *Les Réseaux Neuromimétiques*, Hermès.

Jodouin, J.-F. (1994b). *Les Réseaux De Neurones : Principes Et Définitions*, Hermès, Paris.

Jong, d. P. (1991). "The Diffuse Kalman Filter." *Annals of Statistics*, 19, 1073-1083.

Jordan, M. I. (1990). "Motor Learning and the Degrees of Freedom Problem." Attention and Performance Xiii: Motor Representation and Control, M. Jeannerod, ed., Erlbaum, Hillsdale, New York, 796-836.

Jordan, M. I., et Jacobs, R. A. (1994). "Hierarchical Mixtures of Experts and the Em Algorithm." *Neural Computation*, 6(2), 181-214.

Jordan, M. I., et Jacobs, R. A. (1995a). "Learning in Modular and Hierarchical Systems." The Handbook of Brain Theory and Neural Networks, M. Arbib, ed., MIT Press, Cambridge, MA.

Jordan, M. I., et Jacobs, R. A. (1995b). "Modular and Hierarchical Learning Systems." The Handbook of Brain Theory and Neural Networks, M. Arbib, ed., MIT Press.

Julier, S. J., et Uhlmann, J. K. (1997). "A New Extension to the Kalman Filter to Nonlinear Systems." *AeroSense: The 11th Internatioanl Symposium on Aerospace/Defense Sensing, Simulation and Controls*, Orlando, Florida.

Julier, S. J., Uhlmann, J. K., et Durrant-Whyte, H. F. (1995). "A New Approach for Filtering Nonlinear Systems." *1995 American Control Conference*, Seattle, Washington, 1628-1632.

Jutten, C., et Comon, P. (1993). "Neural Bayesian Classifier." *International Workshop on Artificial Neural Networks (IWANN'93)*, Barcelona (Spain).

Kalman, R. E. (1960). "A New Approach to Linear Filtering and Prediction Problems." *Transactions ASME, Journal of Basic Engineering*, 82, 35-45.

Kangas, J. (1990). "Time-Delayed Self-Organizing Maps." *International Joint Conference on Neural Networks*, San Diego, 331-336.

Kangas, J. (1991). "Phoneme Recognition Using Time-Dependent Versions of Self-Organizing Maps." *International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, 101-104.

Kara, R., Wira, P., et Kihl, H. (2000a). "Hierarchical Neural Controller and Adaptive Kalman Filter for Visual Robot Tracking Tasks." *International ICSC Congress on Intelligent Systems and Applications (ISA'2000)*, University of Wollongong (near Sydney), Australia.

Kara, R., Wira, P., et Kihl, H. (2000b). "Robot Vision Tracking with a Hierarchical Cmac Controller." *Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies (KES'2000)*, Brighton, UK, 271-274.

Kaski, S., Kangas, J., et Kohonen, T. (1998). "Bibliography of Self-Organizing Map (Som) Papers: 1981-1997." *Neural Computing Surveys*, 1(3-4), 1-176.

Kawato, M. (1989). "Adaptation and Learning in Control of Voluntary Movement by the Central Nervous System." *Advanced Robotics*, 3(3), 229-249.

Kawato, M., Isobe, M., Maeda, Y., et Suzuki, R. (1988). "Coordinates Transformation and Learning Control for Visually-Guided Voluntary Movement with Iteration: A Newton-Like Method in a Function Space." *Biological Cybernetics*, 59, 161-177.

Kawato, M., Maeda, Y., Uno, Y., et Suzuki, R. (1990). "Trajectory Formation of Arm Movement by Cascade Neural Network Model Based on Minimum Torque-Change Criterion." *Biological Cybernetics*, 62, 275-288.

Kawato, M., Uno, Y., Isobe, M., et Suzuki, R. (1987). "A Hierarchical Model for Voluntary Movement and Its Application to Robotics." *IEEE International Conference on Neural Networks*, San Diego, CA, 573-582.

Kelly, A. (1994). "A 3d Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles." *CMU-RI-TR-94-19*, The Robotic Insitute, Carnegie Mellon University, Pittsburgh.
http://www.ri.cmu.edu/people/kelly_alonzo.html

Kirlin, R. L., et Moghaddamjoo, A. (1986a). "Robust Adaptive Kalman Filtering for Systems with Unknown Step Inputs and Non-Gaussian Measurement Errors." *IEEE Transactions On Acoustics, Speech, and Signal Processing*, 34(2), 252-263.

Kirlin, R. L., et Moghaddamjoo, A. (1986b). "A Robust Running Window Detector and Estimator for Step Signals in Contaminated Gaussian Noise." *IEEE Transactions On Acoustics, Speech, and Signal Processing*, 34(4), 816-823.

Kirschning, I., Tomabechi, H., et Aoe, J. I. (1995). "A Parallel Recurrent Cascade-Correlation Neural Network with Natural Connectionist Glue." *IEEE International Conference on Neural Networks*, Perth, Australia, 953-956.

Kohavi, R., et Wolpert, D. H. (1996). "Bias Plus Variance Decomposition for Zero-One Loss Functions." *13th International Conference on Machine Learning (ICML'96)*, Bari, Italy, 275-283.

Kohn, R., et Ansley, C. F. (1986). "Estimation, Prediction, and Interpolation for Arima Models with Missing Data." *Journal of the American Statistical Association*, 8l, 751-761.

Kohonen, T. (1982a). "Analysis of a Simple Self-Organizing Process." *Biological Cybernetics*, 44(2), 135-140.

Kohonen, T. (1982b). "Self-Organized Formation of Topologically Correct Feature Maps." *Biological Cybernetics*, 43(1), 59-69.

Kohonen, T. (1984). *Self-Organization and Associative Memory*, Springer-Verlag, Berlin.

Kohonen, T. (1995). *Self-Organizing Maps*, Springer-Verlag, Berlin.

Kohonen, T., Kaski, S., et Lappalainen, H. (1997). "Self-Organized Formation of Various Invariant-Feature Filters in the Adaptive-Subspace Som." *Neural Computation*, 9(6), 1321-1344.

Kolmogoroff, A. N. (1941). "Interpolation and Extrapolation Von Stationären Zufälligen Folgen." *Bulletin of the Academic Sciences of U.S.S.R, Series in Mathematics*, 5, 3-14.

Kosaka, A., et Nakazawa, G. (1995). "Vision-Based Motion Tracking of Rigid Objects Using Prediction of Uncertainties." *International Conference on Robotics and Automation*, Nagoya, Japan, 2637-2644.

Koskela, T., Varsta, M., Heikkonen, J., et Helsinki, K. K. (1998). "Recurrent Som with Local Linear Models in Time Series Prediction." *6th European Symposium on Artificial Neural Networks (ESANN'98)*, Brussels, Belgium, 167-172.

Kosko, B. (1988). "Bidirectional Associative Memories." *IEEE Transactions on Systems, Man, and Cybernetics*, 18, 49-60.

Kröse, B. J. A., Van Der Smagt, P., et Groen, F. C. A. (1993). "A One-Eyed Self Learning Robot Manipulator." Neural Networks in Robotics Publishers, G. Bekey and K. Golberg, eds., Kluwer Academic Publishers, Dordrecht, 19-28.

Kubota, T., et Hashimoto, H. (1996). "Visual Control of Robotic Manipulator Based on Neural Networks." Neural Networks for Robotic Control, Theory and Applications, A. M. S. Zalgala and A. S. Morris, eds., Ellis Horwood.

Kuperstein, M. (1988). "Neural Model for Adaptive Hand-Eye Coordination for Single Postures." *Sciences*, 239, 1308-1311.

Lallemand, J.-P., et Zeghloul, S. (1994). *Robotique : Aspects Fondamentaux, Modélisation Mécanique, Cao Robotique, Commande Avec Exercices Corrigés*, Masson, Paris.

Lampinen, J., et Vehtari, A. (2001). "Bayesian Approach for Neural Networks - Review and Case Studies." *Neural Networks*, 14(3), 257-274.

Lange, F., Langwald, J., et Hirzinger, G. (1999). "Predictive Feedforward Control of High Speed Tracking Tasks." *European Control Conference (ECC'99)*, Karlsruhe, Germany.

Lange, F., Wunsch, P., et Hirzinger, G. (1998). "Predictive Vision Based Control of High Speed Industrial Robot Paths." *IEEE International Conference on Robotics and Automation (ICRA'98)*, Leuven, Belgium, 2646-2651.

Lebart, L. (1997). "Validation Et Rééchantillonnage." Statistiques Et Méthodes Neuronales, S. Thiria, Y. Lechevallier, O. Gascuel, and S. Canu, eds., Dunod, Paris, 275-291.

LeCun, Y., Denker, J. S., et Solla, S. A. (1990). "Optimal Brain Damage." *Advances in Neural Information Processing Systems*, San Mateo, CA, 598-605.

Lee, K. Y., et Jung, S. W. (2000). "Time-Domain Approach Using Multiple Kalman Filters and Em Algorithm to Speech Enhancement with Nonstationary Noise." *IEEE Transactions on Speech & Audio Processing*, 8(3), 282-291.

Legnani, G. (1998). "Spacelib in C - User's Manual." University of Brescia, Mechanical Engineering Department, Brescia.
http://www.ing.unibs.it/~legnani/

Lehtokangas, M. (2000). "Cascade-Correlation Learning for Classification." *IEEE Transactions on Neural Networks*, 11(3), 795-798.

Lengellé, R., et Schaltenbrand, N. (1997). "Détermination De L'architecture Des Réseaux Multicouches, Application À Un Problème De Discrimination : L'analyse Automatique De Sommeil." Statistiques Et Méthodes Neuronales, S. Thiria, Y. Lechevallier, O. Gascuel, and S. Canu, eds., Dunod, Paris, 53-74.

Leung, C.-S., Wong, K.-W., Sum, P.-F., et Chan, L.-W. (2001). "A Pruning Method for the Recursive Least Squared Algorithm." *Neural Networks*, 14(2), 147-174.

Lin, S. H., Kung, S. Y., et Lin, L. J. (1997). "Face Recognition/Detection by Probabilistic Decision-Based Neural Network." *IEEE Transactions on Neural Networks*, 8(1), 114-132.

Lippmann, R. P. (1987). "An Introduction to Computing with Neural Nets." *IEEE Acoustical Speech and Signal Processing Magazine*, 4-22.

Lowe, D. G. (1992). "Robust Model-Based Motion Tracking through the Integration of Search and Estimation." *International Journal of Computer Vision*, 8(2), 113-122.

Luttrell, S. P. (1989). "Hierarchical Vector Quantizations." *Proceedings of the IEE, Part I*, 136, 405-413.

Luttrell, S. P. (1994). "A Bayesian Analysis of Self-Organising Maps." *Neural Computation*, 6(5), 767-794.

Mackay, D. J. (1992a). "A Practical Bayesian Framework for Backpropagation Networks." *Neural Computation*, 4(3), 448-472.

Mackay, D. J. C. (1992b). "Bayesian Interpolation." *Neural Computation*, 4(3), 415-447.

Magill, D. T. (1965). "Optimal Adaptive Estimation of Sampled Stochastic Processes." *IEEE Transactions on Automatic Control*, AC-10(4), 434-439.

Malis, E., Chaumette, F., et Boudet, S. (1998). "Positioning a Coarse-Calibrated Camera with Respect to an Unknown Object by 2d 1/2 Visual Servoing." *IEEE International Conference on Robotics and Automation (ICRA'98)*, Leuven, Belgium, 1352-1359.

Malis, E., Chaumette, F., et Boudet, S. (1999). "2-1/2-D Visual Servoing." *IEEE Transactions on Robotics and Automation*, 15(2), 238-250.

Mani, G. (1991). "Lowering Variance of Decisions by Using Artificial Neural Network Portfolios." *Neural Computation*, 3(4), 484-486.

Marchand, E., Bouthemy, P., et Chaumette, F. (2001). "A 2d-3d Model-Based Approach to Real-Time Visual Tracking." *Image and Vision Computing*, 19(13), 941-955.

Marchand, E., Bouthemy, P., Chaumette, F., et Moreau, V. (1999). "Robust Real-Time Visual Tracking Using a 2d-3d Model-Based Approach." *IEEE International Conference on Computer Vision (ICCV'99)*, Corfou, Grêce, 262-268.

Marr, D. (1982). *Vision*, Freeman, San Francisco.

Martinet, P. (1999). "Asservissement Visuel." Habilitation à diriger des recherches, Université Blaise Pascal, LASMEA, Clermont-Ferrand.

Martinet, P., Daucher, N., Gallice, J., et Dhome, M. (1997). "Robot Control Using Monocular Pose Estimation." *Workshop on New Trends in Image-based Robot Servoing, (IROS'97)*, Grenoble, France, 1-12.

Martinetz, T. M., Berkovich, S. G., et Schulten, K. J. (1993). "Neural-Gas Network for Vector Quantization and Its Application to Time-Series Prediction." *IEEE Transactions on Neural Networks*, 4(4), 558-569.

Martinetz, T. M., et Schulten, K. J. (1991). "A "Neural-Gas" Network Learns Topologies." *International Conference on Artificial Neural Networks*, Espoo, Finland, 397-402.

Martinetz, T. M., et Schulten, K. J. (1994). "Topology Representing Networks." *Neural Networks*, 7(3), 507-522.

Massone, L. L. E. (1995). "Sensorimotor Learning." Handbook of Brain Theory and Neural Networks, M. A. Arbib, ed., The MIT Publisher, 860-864.

Mathis, H., von Hoff, T. P., et Joho, M. (2001). "Blind Separation of Signals with Mixed Kurtosis Signs Using Threshold Activation Functions." *IEEE Transactions on Neural Networks*, 12(3), 618-624.

Mazor, E., Averbuch, A., Bar-Shalom, Y., et Dayan, J. (1998). "Imm Methods in Target Tracking: A Survey." *IEEE Transactions on Aerospace and Electronic Systems*, 34(1), 103-123.

Mead, C. (1990). "Neuromorphic Electronic Systems." *Proceedings of the IEEE*, 78(10), 1629-1636.

Mehra, R. K. (1970). "On the Identification of Variances and Adaptive Kalman Filtering." *IEEE Transactions on Automatic Control*, AC-15(2), 175-184.

Mehra, R. K. (1972). "Approaches to Adaptive Filtering." *IEEE Transanction on Automatic Control*, AC-17(5), 693-698.

Mel, B. W. (1990). "Vision Based Robot Motion Planning." Neural Networks for Control, W. T. Miller, III, R. S. Sutton, and P. J. Werbos, eds., The MIT Press, Cambridge, MA, 229-254.

Merwe, R. v. d., Doucet, A., Freitas, N. d., et Wan, E. (2001). "The Unscented Particle Filter." *Advances in Neural Information Processing Systems (NIPS13)*.

Merwe, R. v. d., et Wan, E. A. (2001). "Efficient Derivative-Free Kalman Filters for Online Learning." *European Symposium on Artificial Neural Networks (ESANN'2001)*, Bruges, Belgium, 205-210.

Mézard, M., et Nadal, J.-P. (1989). "Learning in Feedforward Layered Networks: The Tiling Algorithm." *Journal of Physics*, A, 22(12), 2191-2203.

Miller, W. T., III. (1989). "Real-Time Application of Neural Networks for Sensor-Based Control of Robots with Vision." *IEEE Transactions on Systems, Man, and Cybernetics*, 19(4), 825-831.

Miller, W. T., III, Hewes, R. P., Glanz, F. H., et Kraft, L. G. (1990a). "Real-Time Dynamic Control of an Industrial Manipulator Using a Neural Network Based Learning Controller." *IEEE Transactions on Robotics and Automation*, 6, 1-9.

Miller, W. T., III, Sutton, R. S., et Werbos, P. J. (1990b). "Neural Networks for Control." The MIT Press, Cambridge, MA, 524.

Miyamoto, H., Kawato, M., Setoyama, T., et Suzuki, R. (1988). "Feedback-Error-Learning Neural Network for Trajectory Control of a Robotic Manipulator." *Neural Networks*, 1, 251-265.

Moerland, P. (1997). "Mixtures of Experts Estimate a Posteriori Probabilities." *International Conference onArtificial Neural Networks (ICANN'97)*, Lausanne, 499-504.

Moghaddamjoo, A. (1989). "Approaches to Adaptive Kalman Filtering." *Control Theory and Advanced Technology*, 5(1), 1-18.

Moghaddamjoo, A., et Kirlin, R. L. (1989). "Robust Adaptive Kalman Filtering with Unknown Inputs." *IEEE Transactions On Acoustics, Speech, and Signal Processing*, 37(8), 1166-1175.

Moghaddamjoo, A., et Kirlin, R. L. (1993). "Robust Adaptive Kalman Filtering." Approximate Kalman Filtering, G. Chen, ed., World Scientific Publishing Company., Singapore/New Jersey, 65-85.

Murakami, H., et Kumar, V. (1982). "Efficient Calculation of Primary Images from a Set of Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(5), 511-515.

Murase, H., et Nayar, S. (1994). "Learning, Positioning and Tracking Visual Appearance." *ICRA'94*.

Murray, J. D. (1993). *Mathematical Biology*, Springer-Verlag, Berlin.

Myers, K. A., et Tapley, B. D. (1976). "Adaptive Sequential Estimation with Unknown Noise Statistics." *IEEE Transactions on Automatic Control*, AC-21, 520-523.

Nagahama, K., Hashimoto, K., Noritsugu, T., et Takaiwa, M. (2000). "Visual Servoing Based on Object Motion Estimation." *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, Takamatsu, Japan, 245-250.

Narendra, K. S. (1990). "Adaptive Control Using Neural Networks." Neural Networks for Control, W. T. Miller, III, R. S. Sutton, and P. J. Werbos, eds., The MIT Press, Cambridge, MA, 115-142.

Nayar, S., Nene, S., et Murase, H. (1996). "Subspace Methods for Robot Vision." *IEEE Transactions on Robotic and Automation*, 12(5), 750-758.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*, Springer, Heidelberg.

Nechyba, M. C., et Xu, Y. (1997). "Cascade Neural Networks with Node-Decoupled Extended Kalman Filtering." *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 214-219.

Nethery, J., et Spong, M. W. (1994). "Robotica: A Mathematica Package for Robot Analysis." *IEEE Robotics and Automation Magazine*, 1(1), 13-21. ftp://ftp.csl.uiuc.edu/pub/robotica/

Neubauer, A. (1995). "A Study of Extended Learning Strategies for Bidirectional Associative Memories." *International Conference on Neural Networks (ICNN'95)*, 1772-1776.

Niemeyer, G., et Slotine, J. J. E. (1991). "Performance in Adaptive Manipulator Control." *International Journal of Robotics Research*, 10(2).

Omidvar, O., et van der Smagt, P. (1997). "Neural Systems for Robotics." Academic Press, Boston, Massachusetts, 346.

Ormoneit, D. (1999). "A Regularization Approach to Continuous Learning with an Application to Financial Derivatives Pricing." *Neural Networks*, 12(10), 1399-1404.

Oudjane, N. (2000). "Stabilité Et Approximations Particulaires En Filtrage Non Linéaire, Application Au Pistage," Thèse de Doctorat, Université de Rennes I, Rennes.

Oudjane, N., et Musso, C. (1999). "Mutliple Model Particle Filter." *17ème Colloque GRETSI sur le Traitement du Signal et des Images*, Vannes, France, 681-683.

Papanikolopoulos, N., Khosla, P. K., et Kanade, T. (1993). "Visual Tracking of a Moving Target by a Camera Mounted on a Robot: A Combination of Control and Vision." *IEEE Transactions on Robotics and Automation*, 9(1), 14-35.

Park, J., Kwon, C.-H., et Park, D.-C. (2001). "An Optimization-Based Design Procedure for Asymmetric Bidirectional Associative Memories." *IEEE Transactions on Neural Networks*, 12(1), 169-170.

Paul, R. P. (1981). *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press.

Perello, S. (1999). "Simulation D'une Scène Robotique Pour L'asservissement Visuel." Rapport technique *EEA-TROP-TR-99-01*, Université de Haute-Alsace, Mulhouse.

Perrone, M. P. (1993). "Improving Regression Estimation: Averaging Methods for Variance Reduction with Extensions to General Convex Measure Optimization," PhD Thesis, Brown University.

Perrone, M. P., et Cooper, L. N. (1993). "When Networks Disagree: Ensemble Methods for Hybrid Neural Networks." Artificial Neural Networks for Speech and Vision, R. J. Mamone, ed., Chapman & Hall, London, 126-142.

Picard, J. (1989). "Efficiency of the Extended Kalman Filter for Non Linear Systems with Small Noise." Rapport de Recherche *RR-1068*, INRIA, Sophia Antipolis.

Piepmeier, J. A., Lipkin, H., et McMurray, G. V. (1998a). "A Predictive Estimation Approach to Model Independent Visual Servoing." *Robotics'98*, Albuquerque, New Mexico.

Piepmeier, J. A., McMurray, G. V., et Lipkin, H. (1998b). "Tracking a Moving Target with Model Independent Visual Servoing: A Predictive Estimation Approach." *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 2652-2657.

Price, C. F. (1968). "An Analysis of the Divergence Problem in the Kalman Filter." *IEEE Transactions on Automatic Control*, AC-13(6), 699-702.

Puskorius, G. V., et Feldkamp, L. A. (1991). "Decoupled Extended Kalman Filter Training of Feedforward Layered Neural Networks." *Int. Joint Conf. Neural Networks*, 771-777.

Puskorius, G. V., et Feldkamp, L. A. (1994). "Neurocontrol of Nonlinear Dynamical Systems with Kalman Filter Trained Recurrent Networks." *IEEE Transactions on Neural Networks*, 5, 279-297.

Quinlan, J. R. (1993). "C4.5: Programs for Machine Learning." San Mateo, CA.

Ramamurti, V., et Ghosh, J. (1999). "Structurally Adaptive Modular Networks for Nonstationary Environments." *IEEE Transactions on Neural Networks*, 10(1), 152-160.

Reed, R. (1993). "Pruning Algorithms - a Survey." *IEEE Transactions on Neural Networks*, 4(5), 740-747.

Reilly, D. L., Cooper, L. N., et Elbaum, C. (1982). "A Neural Model for Category Learning." *Biological Cybernetics*, 45, 35-41.

Reyneri, L. M. (1999). "Unification of Neural and Wavelet Networks and Fuzzy Systems." *IEEE Transactions on Neural Networks*, 10(4), 801-814.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge.

Ritter, H. J., Martinetz, T. M., et Schulten, K. J. (1992). *Neural Computation and Self-Organizing Maps*, Addison-Wesley, Reading, MA.

Ritter, H. J., et Schulten, K. J. (1986). "Topology-Conserving Mappings for Learning Motor Tasks." *International Conference on Neural Networks for Computing*, Snowbird, Utah, 376-380.

Ritter, H. J., et Schulten, K. J. (1988a). "Convergence Properties of Kohonen's Topology Preserving Maps: Fluctuations, Stability, and Dimension Selection." *Biological Cybernetics*, 60, 59-71.

Ritter, H. J., et Schulten, K. J. (1988b). "Extending Kohonen's Self-Organizing Mapping Algorithm to Learn Ballistic Movements." Neural Computers, R. Eckmiller and C. von der Malsburg, eds., Springer-Verlag, Berlin, 393-406.

Rogova, G. (1994). "Combining the Results of Several Neural Network Classifiers." *Neural Networks*, 7(5), 777-781.

Ronco, E. (1997). "Incremental Polynomial Controller Networks: Two Self-Organising Non-Linear Controllers," PhD Thesis, Faculty of Mechanical Engineering, University of Glasgow, Glasgow.

Ronco, E., et Gawthrop, P. (1995). "Modular Neural Networks : A State of the Art." Rapport Technique *CSC-95026*, Center of System and Control, University of Glasgow. http://www.mech.gla.ac.uk/control/report.html

Rose, K., Gurewitz, E., et Fox, G. C. (1992). "Vector Quantization by Deterministic Annealing." *IEEE Transactions on Information Theory*, 38(4), 1249-1258.

Rousseaux, P., Mallieu, D., Van Cutsem, T., et Ribbens-Pavella, M. (1988). "Dynamic State Prediction and Hierarchical Filtering for Power System State Estimation." *Automatica*, 24(5), 595-618.

Rouzier, S. (1998). "Réseaux Neuronaux Et Modularité," Thèse de Doctorat, INPG, Grenoble.

Roweis, S., et Ghahramani, Z. (1999). "A Unifying Review of Linear Gaussian Models." *Neural Computation*, 11(2), 305-345.

Ruck, D. W., Rogers, S. K., Kabrisky, M., Maybeck, P. S., et Oxley, M. E. (1992). "Comparative Analysis of Backpropagation and the Extended Kalman Filter for Training Multilayer Perceptrons." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(6), 686-690.

Rumelhart, D. E., et McClelland, J. L. (1986). "Parallel Distributed Processing." MIT Press.

Salut, G. (1999). "La Technique Particulaire Et Ses Développements." *Journées Doctorales d'Automatique (JDA'99)*, Nancy (France).

Sanchez, E. N., et Bernal, M. A. (2000). "Adaptive Recurrent Control for Nonlinear System Tracking." *IEEE Transaction on Systems, Man, and Cybernetics, Part B: Cybernetics*, 30(6), 886-889.

Sanderson, A. C., et Weiss, L. E. (1980). "Image Based Visual Servo Control Using Relational Graph Error Signals." *International Conference on Cybernetics Graph Error Signal*, 1074-1077.

Savkin, A. V., et Petersen, I. R. (1996). "Model Validation for Robust Control of Uncertain Systems with an Integral Quadratic Constraint." *Automatica*, 32(4), 603-606.

Savkin, A. V., et Petersen, I. R. (1998). "Robust State Estimation and Model Validation for Discrete-Time Uncertain Systems with a Deterministic Description of Noise and Uncertainty." *Automatica*, 34(2), 271-274.

Sayed, A. H., et Kailath, T. (1994). "A State-Space Approach to Adaptive Rls Filtering." *IEEE Signal Processing Magazine*, 11(3), 18-60.

Sejnowski, T.J., K., C., et Churchland, P. S. (1988). "Computational Neuroscience." *Science*, 241, 1299-1306.

Shah, S., Palmieri, F., et Datum, M. (1992). "Optimal Filtering Algorithms for Fast Learning in Feedforward Neural Networks." *Neural Networks*, 5(5), 779-787.

Sharkey, N. E., et Sharkey, A. J. C. (1992). "Adaptive Generalization and the Transfer of Knowledge." *Second Irish Neural Networks Conference*, Belfast.

Shin, Y., et Ghosh, J. (1995). "Ridge Polynomial Networks." *IEEE Transactions on Neural Networks*, 6(3), 610-622.

Shumway, R. H., et Stoffer, D. S. (1982). "An Approach to Time Series Smoothing and Forecasting Using the Em Algorithm." *Journal of Time Series Analysis*, 3(4), 253-264.

Slotine, J.-J. E., et Lohmiller, W. (2001). "Modularity, Evolution, and the Binding Problem: A View from Stability Theory." *Neural Networks*, 14(2), 137-145.

Smotroff, I. G., Friedman, D. H., et Connolly, D. (1991). "Self Organizing Modular Neural Networks." *International Joint Conference on Neural Networks*, Seattle, WA, USA, 187-192.

Sorenson, H.-W., et Sacks, J. E. (1971). "Recursive Fading Memory Filtering." *Information Sciences*, 3, 101-119.

Soucek, B. (1989). *Neural and Concurrent Real-Time Systems the Sixth Generation*, Wiley Interscience, New York.

Specht, D. F. (1990). "Probabilistic Neural Networks." *Neural Networks*, 3(1), 109-118.

Stan, O., et Kamen, E. (2000). "A Local Linearized Least Squares Algorithm for Training Feedforward Neural Networks." *IEEE Transactions on Neural Networks*, 11(2), 487-495.

Stengel, R. F. (1986). *Stochastic Optimal Control, Theory and Application*, John Willey and sons, New York.

Strandlie, A., et Zerubia, J. (1999). "A Deterministic Annealing Pmht Algorithm with an Application to Particle Tracking." Rapport de Recherche *RR-3711*, INRIA, Sophia Antipolis. ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RR/RR-3711.pdf

Sugiyama, M., et Ogawa, H. (2000). "A New Information Criterion for the Selection of Subspace Models." *8th European Symposium on Artificial Neural Networks*, Bruges, Belgium, 69-74.

Sum, J., Chan, L.-w., Leung, C.-s., et Young, G. H. (1998). "Extended Kalman Filter-Based Pruning Method for Recurrent Neural Networks." *Neural Computation*, 10(6), 1481-1505.

Sum, J., Leung, C.-S., Young, G. H., et Kan, W.-K. (1999). "On the Kalman Filtering Method in Neural-Network Training and Pruning." *IEEE Transactions on Neural Networks*, 10(1), 161-166.

Sutton, R. S. (1988). "Learning to Predict by the Methods of Temporal Difference." *Machine Learning*, 3, 9-44.

Sutton, R. S. (1992a). "Adapting Bias by Gradient Descent: An Incremental Version of Delta-Bardelta." *Tenth National Conference on Artificial Intelligence*, 171-176.

Sutton, R. S. (1992b). "Gain Adaptation Beats Least Squares?" *Seventh Yale Workshop on Adaptive and Learning Systems*, New Haven, CT., 161-166.

Sutton, R. S., et Barto, A. G. (1998). *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA.

Thiria, S., Lechevallier, Y., Gascuel, O., et Canu, S. (1997). "Statistiques Et Méthodes Neuronales." Dunod, Paris, 311.

Tibshirani, R. (1996). "Bias, Variance and Prediction Error for Classification Rules." Technical report, University of Toronto, Toronto. ftp://utstat.toronto.edu/pub/tibs/biasvar.ps

Tresp, V., Briegel, T., et Moody, J. (1999). "Neural-Network Models for the Blood Glucose Metabolism of a Diabetic." *IEEE Transactions on Neural Networks*, 10(5), 1204-1213.

Tresp, V., et Taniguchi, M. (1995). "Combining Estimators Using Non-Constant Weighting Functions." *Advances in Neural Information Processing Systems*, 419-426.

Tumer, K., et Ghosh, J. (1996). "Error Correlation and Error Reduction in Ensemble Classifiers." *Connection Science, Special issue on combining artificial neural networks: ensemble approaches*, 8(3/4), 385-404.

Turney, R. D., Reza, A. M., et Delva, J. G. R. (1999). "Fpga Implementation of Adaptive Temporal Kalman Filter for Real Time Video Filtering." *1999 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP99)*, Phoenix, Arizona.

Ukrainec, A. M., et Haykin, S. (1996). "A Modular Neural Network for Enhancement of Cross-Polar Radar Targets." *Neural Networks*, 9(1), 143-168.

Urban, J.-P., Buessler, J.-L., et Gresser, J. (1998). "Neural Networks for Visual Servoing in Robotics." Soft Computing for Intelligent Robotic Systems, L. C. Jain and T. Fukuda, eds., Springer-Verlag, Heidelberg, 81-111.

Urban, J. P., Buessler, J. L., et Wira, P. (1997). "Neuromodule-Based Visual Servoing of a Robot Arm with a 2 D.O.F. Camera." *IEEE International Conference on Systems Man and Cybernetics*, Orlando, 3507-3512.

Utsugi, A. (1996). "Topology Selection for Self-Organizing Maps." *Network: Computation in Neural Systems*, 7(7), 727-740.

Vaidyanathan, P. P. (1990). "Multirate Digital Filters, Filter Banks, Polyphase Networks and Applications:A Tutorial." *Proceedings of the IEEE*, 78(1), 56-91.

Vaidyanathan, P. P. (1993). *Multirate Systems and Filter Banks*, Prentice Hall, Englewood Cliffs, NJ.

van der Smagt, P. (1994). "Simderella: A Robot Simulator for Neuro-Controller Design." *Neurocomputing*, 6(2), 281-285.
http://www.robotic.dlr.de/Smagt/software/simderella/

Vapnik, V. N. (1995). *The Nature of Statistical Learning*, Springer, New York.

Varsta, M., Millán, J. d. R., et Heikkonen, J. (1997). "A Recurrent Self-Organizing Map for Temporal Sequence Processing." *7th International Conference on Artificial Neural Networks*, 421-426.

Venayagamoorthy, G. K., Harley, R. G., et Wunsch, D. C., II. (2000). "Comparison of a Heuristic Dynamic Programming and a Dual Heuristic Programming Based Adaptive Critics Neurocontroller for a Turbogenerator." *International Joint Conference on Neural Networks (IJCNN'00)*, Como, Italy.

Villmann, T. (2001). "Evolutionary Algorithms and Neural Networks in Hybrid Systems." *9th European Symposium on Artificial Neural Networks (ESANN'2001)*, Bruges (Belgium), 137-152.

Villmann, T., Der, R., et Martinetz, T. (1997). "Topology Preservation in Self-Organizing Feature Maps: Exact Definition and Measurement." *IEEE Transactions on Neural Networks*, 8(2), 256-266.

Wada, Y., Kaneko, Y., Nakano, E., Osu, R., et Kawato, M. (2001). "Quantitative Examinations for Multi Joint Arm Trajectory Planningusing a Robust Calculation Algorithm of the Minimum Commanded Torque Change Trajectory." *Neural Networks*, 14(4-5), 381-393.

Wada, Y., et Kawato, M. (1993). "A Neural Network Model for Arm Trajectory Formation Using Forward and Inverse Dynamics Models." *Neural Networks*, 6(7), 919-932.

Walcott, B. L., et Zak, S. H. (1987). " State Observation of Nonlinear Uncertain Dynamical Systems." *IEEE Transactions on Automatic Control*, 32, 166-170.

Walter, J., et Ritter, H. (1995). "Local Psoms and Chebyshev Psoms Improving the Psom." *International Conference on Artificial Neural Networks (ICANN'95)*, 95-102.

Wan, E. A., et Merwe, R. v. d. (2001). "The Unscented Kalman Filter." Kalman Filtering and Neural Networks, S. Haykin, ed., Wiley Publishing.

Wan, E. A., et Nelson, A. T. (1996). "Dual Kalman Filtering Methods for Nonlinear Prediction, Estimation, and Smoothing." *Advances in Neural Information Processing Systems*.

Wang, H., Liu, G. P., Harris, C. J., et Brown, M. (1995). *Advanced Adaptive Control*, Pergamon Press, Oxford, OX, U.K.

Wang, Z.-Q., Manry, M. T., et Schiano, J. L. (2000). "Lms Learning Algorithms: Misconceptions and New Results on Convergence." *IEEE Transactions on Neural Networks*, 11(1), 47-56.

Waterhouse, S. R. (1997). "Classification and Regression Using Mixtures of Experts," PhD Thesis, Cambridge University, Cambridge.

Waterhouse, S. R., et Robinson, A. J. (1996). "Constructive Algorithms for Hierarchical Mixtures of Experts." *Neural Information Processing Systems*, 584-590.

Watkins, C. J. C. H., et Dayan, P. (1992). "Q-Learning." *Machine Learning*, 8(3-4), 279-292.

Weiss, L. E. (1984). "Dynamic Visual Servo Control of Robots: An Adaptive Image-Based Approach," PhD Thesis, Carnegie-Mellon University.

Weiss, L. E., Sanderson, A. C., et Neuman, C. P. (1987). "Dynamic Sensor-Based Control of Robots with Visual Feedback." *IEEE Journal of Robotics and Automation*, 3(5), 404-417.

Wen, W. X., Jennings, A., et Liu, H. (1992). "Learning a Neural Tree." *International Conference on Neural Networks*, Beijing, China, 283-288.

Werbos, P. J. (1992). "Neurocontrol and Supervised Learning: An Overview and Evaluation." Handbook of Intelligent Control: Fuzzy, Neural, and Adaptive Approaches, D. A. White, D. A. Sofge, and R. van Nostrand, eds., 65-89.

Whitehead, B. A., et Choate, T. D. (1996). "Cooperative Competitive Genetic Evolution of Radial Basis Function Centers and Widths for Time Series Prediction." *IEEE Transactions on Neural Networks*, 7(8), 869-880.

Widrow, B., et Walach, E. (1995). *Adaptive Inverse Control*, Prentice Hall Press, Upper Saddle River.

Wiegerinck, W., et Heskes, T. (1994). "On-Line Learning with Time-Correlated Patterns." *Europhysics Letters*, 28(6), 451-455.

Wiener, N. (1948). *Interpolation, Extrapolation and Smoothing of Stationnary Time Series*, Wiley, New York.

Wiener, N. (1961). *Cybernetics: Or Control and Communication in the Animal and in the Machine*, MIT Press, Cambridge, Massachusetts.

Williams, R. J. (1992). "Training Recurrent Networks Using the Extended Kalman Filter." *International Joint Conference on Neural Networks*, Baltimore, MD, 241-246.

Wilson, W., Hulls, C. C. W., et Bell, G. S. (1996). "Relative End-Effector Control Using Cartesian Position Based Visual Servoing." *IEEE Transactions on Robotics and Automation*, 12(5), 684-696.

Wilson, W. J. (1993). "Visual Servo Control of Robots Using Kalman Filter Estimates of Robot Pose Relative to Work-Pieces." Visual Servoing, K. Hashimoto, ed., World Scientific, Singapore, 71-104.

Wira, P. (1997). "Contrôle Neuronal D'un Bras Robotique Avec Une Caméra À Deux Degrés De Liberté." Rapport Technique *EEA-TROP-TR-97-02*, Université de Haute Alsace, Mulhouse.

Wira, P., et Urban, J.-P. (2000). "A New Adaptive Kalman Filter Applied to Visual Servoing Tasks." *Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies (KES'2000)*, Brighton, UK, 267-270.

Wira, P., et Urban, J.-P. (2001a). "Predicting Unknown Motion for Model Independent Visual Servoing." *International Journal of Computational Intelligence and Applications*, 1(3), 287-302.

Wira, P., et Urban, J.-P. (2001b). "Prédiction De Trajectoire Par Filtrage De Kalman Adaptatif À État Discret Multiple." *Journées Doctorales d'Automatique (JDA 2001)*, Toulouse, France, 93-98.

Wira, P., Urban, J.-P., et Gresser, J. (2001). "A Divide-and-Conquer Learning Architecture for Predicting Unknown Motion." *9th European Symposium on Artificial Neural Networks (ESANN'2001)*, Bruges, Belgium, 391-396.

Wojcik, P. J. (1993). "On-Line Estimation of Signal and Noise Parameters and the Adaptive Kalman Filtering." Approximate Kalman Filtering, G. Chen, ed., World Scientific Publishing, Singapore/New Jersey, 87-111.

Wolpert, D. H. (1992). "Stacked Generalization." *Neural Networks*, 5(2), 241-260.

Wolpert, D. H. (1997). "On Bias Plus Variance." *Neural Computation*, 9(6), 1211-1243.

Wolpert, D. M., et Kawato, M. (1998). "Multiple Paired Forward and Inverse Models for Motor Control." *Neural Networks, Special Issue "Neural Control and Robotics : Biology and Technology"*, 11(7-8), 1317-1329.

Xia, Q., Rao, M., Ying, Y., et Shen, X. (1994). "Adaptive Fading Kalman Filter with an Application." *Automatica*, 30(8), 1333-1338.

Xie, X., Sudhakar, R., et Zhuang, H. (1998). "A Cascaded Scheme for Eye Tracking and Head Movement Compensation." *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28(4), 487-490.

Xu, L., et Jordan, M. (1996). "On Convergence Properties of the Em Algorithm for Gaussian Mixtures." *Neural Computation*, 8(1), 129-151.

Xu, L., Krzyzak, A., et Suen, C. Y. (1992). "Methods of Combining Multiple Classifiers and Their Applications to Hand-Written Character Recognition." *IEEE Transactions on Systems, Man and Cybernetics*, 22(3), 418-435.

Yao, X. (1993). "Evolutionary Artificial Neural Networks." *International Journal of Neural Systems*, 4(3), 203-222.

Yao, X. (1999). "Evolving Artificial Neural Networks." *Proceedings of the IEEE, Special Issue on Computational Intelligence*, 87(9), 1423-1447.

Yao, X., et Liu, Y. (1997). "A New Evolutionary System for Evolving Artificial Neural Networks." *IEEE Transactions on Neural Networks*, 8(3), 694-713.

Yaz, E., et Azemi, A. (1993). "Variable Structure Observer with a Boundary-Layer for Correlated Noise/Disturbance Models and Disturbance Minimization." *International Journal of Control*, 57(5), 1191-1206.

Yin, H., et Allinson, N. M. (2001). "Self-Organizing Mixture Networks for Probability Density Estimation." *IEEE Transactions on Neural Networks*, 12(2), 405-411.

Yonghong, S., Saratchandran, P., et Sundararajan, N. (2000). "A Direct Link Minimal Ressource Allocation Network for Adaptive Noise Cancellation." *Neural Processing Letters*, 12(3), 255-265.

Yuille, A. L., et Geiger, D. (1995). "Winner-Take-All Mechanisms." Handbook of Brain Theory and Neural Networks, M. A. Arbib, ed., The MIT Press, 1056-1060.

Zhao, Q., et Higuchi, T. (1996). "Evolutionary Learning of Nearest-Neighbor Mlp." *IEEE Transactions on Neural Networks*, 7(3), 762-767.