



HAL
open science

Reconnaissance de comportements de navires dans une zone portuaire sensible par approches probabiliste et événementielle : application au Grand Port Maritime de Marseille

Salma Zouaoui-Elloumi

► **To cite this version:**

Salma Zouaoui-Elloumi. Reconnaissance de comportements de navires dans une zone portuaire sensible par approches probabiliste et événementielle : application au Grand Port Maritime de Marseille. Architecture, aménagement de l'espace. Ecole Nationale Supérieure des Mines de Paris, 2012. Français. NNT : 2012ENMP0022 . pastel-00737678

HAL Id: pastel-00737678

<https://pastel.hal.science/pastel-00737678v1>

Submitted on 2 Oct 2012

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.

École doctorale n°84 :
Sciences et technologies de l'information et de la communication

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

l' École Nationale Supérieure des Mines de Paris

Spécialité « Contrôle, Optimisation et Prospective »

présentée et soutenue publiquement par

Salma ZOUAOUI-ELLOUMI

le 23 Juillet 2012

**Reconnaissance de comportements de navires dans une zone
portuaire sensible par approches probabiliste et événementielle :
Application au Grand Port Maritime de Marseille**

Directeur de thèse : Mme Valérie ROY
Co-encadrement de la thèse : Mme Nadia MAÏZI

Jury

M. Mekki KSOURI , Professeur, ENIT Tunis, Tunisie	Rapporteur
M. Martin RICHARD , Chargé de Recherche, Université du Québec, Trois-Rivières, Canada	Rapporteur
Mr. Pierre BESSIERE , Professeur, Collège de France, Paris	Examinateur
Mme Annie RESSOUCHE , Chargé de Recherche, INRIA Sophia Antipolis	Examinateur
M. Patrice MISTRETTA , Architecte système et projets avancés, Thales Underwater System, Sophia Antipolis	Membre invité
M. Pierre DEROI , Chef du Département Sûreté Défense, Grand Port Maritime De Marseille	Membre invité

Table des matières

Introduction	1
1 La sécurité des zones portuaires face aux risques	6
1.1 Les zones portuaires et les services offerts	6
1.1.1 Les services de lignes régulières pour les marchandises	8
1.1.2 Les services du transport de marchandises en vrac	9
1.2 Les risques existants dans une zone maritime	11
1.3 Sécurisation des ports et de leurs équipements	16
1.4 La sécurité du Grand Port Maritime de Marseille (GPMM)	21
1.4.1 Le GPMM : un port d'importance vitale pour la France	21
1.4.2 La sécurité du GPMM	23
1.4.3 Le besoin d'un système de sécurité plus robuste	23
1.5 Le projet SECMAR	25
1.5.1 Instruments utilisés pour la récolte des données	26
1.5.2 Disposition des capteurs dans le GPMM	31
1.5.3 Le processus de fusion des données	32
1.5.4 Notre contribution	33
2 la technique de reconnaissance de comportements	37
2.1 Définition	37
2.2 Domaine d'application et méthodes utilisées	40
3 Approche probabiliste et approche événementielle	48
3.1 L'approche probabiliste par MMC	49
3.1.1 Généralités	49
3.1.1.1 Réseau Bayésien	52
3.1.1.2 Réseau Bayésien Dynamique	53
3.1.1.3 Chaîne de Markov	53
3.1.1.4 Filtre Bayésien	53

3.1.1.5	Modèle de Markov Caché	53
3.1.2	La théorie des MMC	54
3.1.2.1	Les trois paramètres d'un MMC	55
3.1.2.2	Types de MMC	58
3.1.2.3	Problématiques des MMC	60
3.1.2.4	Méthodes de résolution	61
3.2	L'approche événementielle par langage réactif synchrone	75
3.2.1	Généralités	75
3.2.1.1	Les systèmes réactifs synchrones	75
3.2.1.2	Les langages réactifs synchrones	75
3.2.1.3	Les autres types de systèmes informatiques	76
3.2.1.4	Implantation d'un système réactif synchrones	76
3.2.2	Un langage événementiel : ESTEREL	77
3.2.2.1	Concepts de base du langage	77
3.2.2.2	Les modules ESTEREL	79
3.2.2.3	La sémantique intuitive	81
3.2.2.4	La sémantique comportementale	82
3.2.2.5	Compilation du programme ESTEREL en automate	88
4	Application à la reconnaissance de comportements de navires	93
4.1	Reconnaissance de comportements de bateaux par MMC	95
4.1.1	Fonctionnement général du module BPAM	95
4.1.2	Construction des modèles de comportements	97
4.1.2.1	Initialisation des modèles de comportements	97
4.1.2.2	Réestimation des modèles de comportements de bateaux dans le port	103
4.1.3	Reconnaissance de comportements de bateaux	107
4.1.4	Exemples de comportements de navires qui se déplacent dans le port	108
4.2	Reconnaissance de comportement par approche événementielle	119
4.2.1	Fonctionnement général du module BRAM	119
4.2.2	Les scénarios de comportements	121
4.2.2.1	Trois types de scénarios	121
4.2.2.2	Scénarios et constats instantanés implémentés pour le GPMM	121
4.2.3	Les différents niveaux d'alarme détectés par la reconnaissance	123
4.2.4	Spécification de la phase de reconnaissance des scénarios ESTEREL	124

4.2.4.1	L'algorithme du module	124
4.2.5	Analyse de scénario ESTEREL	127
4.2.5.1	Un scénario agressif	128
4.2.5.2	Un scénario transgressif	129
4.2.5.3	Les scénarios nominaux	131
4.3	Discussion et complémentarité des deux modules	134
Conclusion		142
Annexe A		149
Annexe B		151

Liste des tableaux

1.1	L'évolution du transport maritime de pétrole brut	10
1.2	Bilan des actes de pirateries perpétrés dans le monde	13
3.1	Instructions manipulant ou non les signaux	80
4.1	L'évolution des valeurs du log-vraisemblance de la séquence d'observations $P(O_1 \dots O_t / \lambda_i)_{1 \leq i \leq 7, 1 \leq t \leq 14}$ par rapport à tous les modèles de comportements pour une séquence d'observation d'un comportement de bateau se dirigeant vers Darse 1.	110
4.2	Liste des constats instantanés	122
4.3	Liste des scénarios	123
4.4	Récapitulation des différents résultats obtenus par les BPAM et le BRAM.	140
4.5	Points forts et points faibles des deux sous-modules : BPAM et BRAM	141

Table des figures

1	Schéma récapitulant les différentes étapes du projet SECMAR	3
2	Les zones fonctionnelles constituant les chenaux d'accès au port qui sont délimitées par la législation	4
1.1	Echanges maritimes mondiaux depuis depuis 1970	8
1.2	Evolution du trafic de transports maritimes en GPL et en GNL.	10
1.3	Abordage d'un navire par des pirates armés	14
1.4	Exemples de scénarios de crise dans un espace maritime	15
1.5	Règles concernant le VTS	20
1.6	Trafic des principaux ports européens	22
1.7	Le radar terrestre et sa couverture circulaire	27
1.8	Les caméras optroniques et leurs couvertures	28
1.9	Schéma général du système AIS	30
1.10	Disposition des capteurs	31
1.11	principe de fusion dans le système SECMAR	32
1.12	Deux scénarios de déplacement de bateaux dans le GPMM.	34
2.1	Modélisation de l'interaction d'un robot avec son environnement par deux approches différentes [31].	42
2.2	Schéma fonctionnel de la méthode de traitement et d'analyse de données pour le développement du HIS [76].	45
3.1	Les approches probabilistes du plus général (en haut) au plus spécifique (en bas) [49].	51
3.2	Graphe d'un Réseau Bayésien	52
3.3	Représentation d'un MMC comme une spécialisation des Filtres Bayésiens	54
3.4	Evolution d'un Modèle de Markov Caché dans le temps	55
3.5	Un Modèle de Markov Caché est stationnaire : $\{a_{ij}\}_{1 \leq i, j \leq N}$ évoluent de la même manière dans le temps	57

3.6	Modèle ergodique	58
3.7	Modèle de Bakis	59
3.8	L'algorithme Baum-Welch	70
3.9	L'automate à états fini associé au module M	92
4.1	Behavior Analysis Modulus.	95
4.2	Module de reconnaissance de comportements par approche probabiliste : le BPAM	97
4.3	Graphe de transition d'états modélisant l'intention de passer d'une zone à une autre	98
4.4	Représentation de chaque zone fonctionnelle par une ellipse	101
4.5	Représentation des paramètres du modèle initial de MMC associé à un comportement de bateaux se dirigeant vers Lavera.	102
4.6	Les log-vraisemblances $\log \{P(O^i/\lambda_i)\}_{1 \leq i \leq 7}$ calculées respectivement pour les courbes (a), (b), (c), (d), (e), (f) et (g) en utilisant l'algorithme Baum-Welch.	105
4.7	Les ellipses associées aux différentes zones du port obtenue suite à leurs réestimation par l'algorithme Baum-Welch	106
4.8	Représentation des paramètres du modèle initial du comportement de bateaux qui se dirigent vers Lavera après apprentissage.	107
4.9	Les valeurs du log-vraisemblance $\{P(O_1 \dots O_t/\lambda_i)\}_{1 \leq i \leq 7}$ calculées pour chaque pas de distance parcourue de 1000m - Le bateau se dirige vers Darse 1.	109
4.10	Evolution de la distance $D(\lambda_i, \lambda_6)$.Après avoir parcouru 11km, D est proche de 0 pour les comportements de bateaux se dirigeant vers Darse 2 et Mouillage Nord Ouest (représentée respectivement en jaune et bleue pour $D(\lambda_5, \lambda_6)$ et $D(\lambda_5, \lambda_7)$). La valeur de D est petite pour le reste des modèles de comportements (pour $i \neq \{5, 6, 7\}$).	111
4.11	Le résultat de la reconnaissance du comportement de la figure (4.9) obtenu par la fenêtre de commande en MATLAB.	114
4.12	Figure (a) Ensemble de trajectoires de bateaux se dirigeant vers Lavera.(b) Les valeurs du log-vraisemblance $P(O_1 \dots O_t/\lambda_i)_{1 \leq i \leq 7}$ pour des observations enregistrées tous les 500 m.	115
4.13	Deux exemples de comportements de bateaux	117
4.14	Module de reconnaissance de comportements par approche réactive : le BRAM	120
4.15	simulateur XES	127
4.16	Scénario agressif	128

4.17	Scénario transgressif	129
4.18	Scénario nominal	131
4.19	Le BAM	134
4.20	Comportement habituel d'un navire se dirigeant vers Lavera.	137
4.21	Comportement anormal d'un bateau qui se dirige vers <i>Mouillage N</i> puis change brutalement de cap vers <i>Lavera</i>	139
4.22	Graphe causal associé à 4.1	145
4.23	Trajectoires de bateaux reconstruites à partir de données AIS pour l'année 2009 [77].	147
4.24	Les couloirs de navigations des navires de charges et des navires de passagers obtenus par le calcul de la densité du Kernel [77].	148
4.25	Les domaines portuaires et aéroportuaires	148

Liste des symboles

AIS Automatic Identification System

BAM Behavior Analysis Modulus

BPAM Behavior Probabilistic Analysis Modulus

BRAM Behavior Reactive Analysis Modulus

CMA Centre de Mathématiques Appliquées

CS Communication et System

GPMM Grand Port Maritime de Marseille

IA Intelligence Artificielle

IHM Interface Homme Machine

MMC Modèle de Markov Caché

PACA Provence-Alpes-Côte-d'Azur

SECMAR SEcurité des biens, des personnes et des installations dans une zone MA-
Ritime.

TRS Thales Radar System

TUS Thales Underwater System

Introduction

De nos jours, une zone portuaire joue un rôle prépondérant dans l'économie mondiale. En effet, la majorité des marchandises échangées entre les pays emprunte la mer. En outre, l'espace maritime a toujours été exploité pour le transport des passagers ou le tourisme.

Pour ces raisons tant économiques que sociales, il est indispensable de prêter à l'espace maritime une attention toute particulière. Ce dernier est en effet exposé à des risques considérables tels que les accidents ou les actes de terrorisme. Nous pouvons citer, à ce titre, le naufrage récent du navire de croisière « *Costa Concordia* » en janvier 2012 sur le littoral sud de la Toscane en Italie qui a fait une trentaine de morts ou encore l'attaque à la bombe contre le pétrolier français Limburg en octobre 2002 dans le port de Ash-Shihr dans le golfe d'Aden au Yemen.

En plus des graves pertes économiques et humaines, la menace écologique, qu'entraîne ce type d'accident, est considérable.

Par ailleurs, les événements du 11 septembre 2001 ont marqué un tournant important dans le renforcement des systèmes de sécurité. Pour cette raison et à partir de cette date, plusieurs normes et systèmes de sécurité ont été mis en place afin de mieux contrôler et surveiller les zones portuaires. Néanmoins, contrairement à l'espace aéroportuaire qui a connu une expansion de ses dispositifs de sécurité, à l'exemple de la mise en service dans plusieurs aéroports d'un scanner corporel afin de détecter des ports d'armes ou de bombes, les systèmes de sécurité dans un espace portuaire sont plus lents à évoluer à cause de la très grande taille de l'espace à surveiller ainsi que de sa complexité.

Dans ce contexte, le projet « Système de SECurité des biens, des personnes et des installations présents dans une zone MARitime sensible » (SECMAR) a été développé en 2007 par un ensemble d'industriels de la région Provence-Alpes-Côte-d'Azur (PACA) dans le cadre d'un pôle de compétitivité¹ afin d'apporter une réponse

1. Un pôle de compétitivité en France est selon la loi de finances pour 2005 : "le regroupement sur un même territoire d'entreprises, d'établissements d'enseignement supérieur et d'organismes de recherche publics ou privés qui ont vocation à travailler en synergie pour mettre en oeuvre des

concrète à la détection d'une menace terroriste dans une zone portuaire en prenant comme site d'expérimentation le Grand Port Maritime de Marseille (GPMM). Les principaux acteurs de ce projet sont Thales Underwater System (TUS), Bertin Technologies et Communication et System (C&S), Thales Radar System (TRS) ainsi que le Centre de Mathématiques Appliquées (CMA) de Mines-ParisTech au sein duquel s'est déroulée cette thèse.

Dans la mesure où le système AIS (Automatic Identification System), qui est considéré comme le meilleur système de détection actuellement utilisé dans tous les ports mondiaux, ne permet pas de détecter tous les navires, il s'agissait pour les partenaires industriels du projet SECMAR de concevoir un nouveau système de détection robuste. Ce nouveau système était formé de capteurs : radar, Sonar et de caméras optroniques². Après que les capteurs aient été installés sur le site du quai pétrole du GPMM, notre objectif, dans le cadre de nos travaux de recherche, était de concevoir un module qui analyse et reconnaît en temps-réel les comportements de navires, à partir des données fusionnées issues de ce système de capteurs dites *pistes systèmes*. Le but de notre système est de faciliter la tâche de surveillance réalisée par le personnel du port. Compte tenu des enjeux concurrentiels du GPMM, la conception d'un tel module automatique de reconnaissance de comportements portuaire. Les étapes du projet SECMAR peuvent être récapitulées par le schéma de la figure (1) où nous pouvons voir que notre système d'analyse et de reconnaissance de comportements se place en aval du système capteur conçu par nos partenaires industriels.

projets de développement économique pour l'innovation."

2. L'optronique est une technique permettant de mettre en œuvre des équipements ou des systèmes utilisant à la fois l'optique et l'électronique. Elle associe généralement un capteur optique, un système de traitement d'images, un système d'affichage ou de mémorisation.

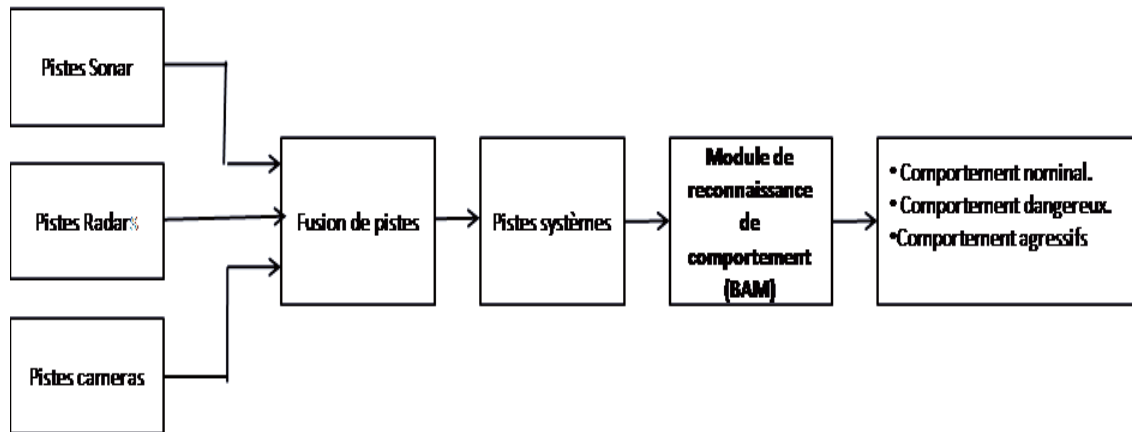


FIGURE 1 – Schéma récapitulant les différentes étapes du projet SECMAR

Nous avons choisi d'appeler ce système, que nous avons conçu dans le cadre de ce travail de thèse, le BAM³.

En plus des pistes systèmes, nous avons également exploité la carte du port partitionnée en zones délimitées par la législation (voir figure (2)) pour concevoir notre module.

Sa conception a été précédée par une phase d'observation de l'ensemble des mouvements de bateaux dans le port dans un intervalle d'une semaine. Nous avons ainsi constaté la présence de deux catégories de comportements. La première catégorie concerne les gros bateaux tels que les pétroliers et les chimiquiers qui se caractérisent par des mouvements réguliers et récurrents dans le port. En effet, vu leur importance, ces bateaux doivent impérativement respecter les règles de la navigation imposées par la législation et ceci en suivant exactement les chenaux d'accès au port qui sont représentés par l'ensemble des zones fonctionnelles de la figure (2).

La deuxième catégorie de comportements est celle des petites embarcations telles que les jets-ski et les voiliers qui se caractérisent par des comportements non-réguliers dans le port. Ces navires peuvent circuler librement dans le port et aucune réglementation ne leur a encore été imposée.

Compte tenu de notre problématique de reconnaissance de tout type de comportements de navires dans le port, nous avons conçu deux sous-modules que nous avons mis en parallèles (le BPAM⁴ et le BRAM⁵) et qui traitent séparément ces deux ca-

3. Behavior Analysis Modulus.

4. Behavior Probabilistic Analysis Modulus.

5. Behavior Reactive Analysis Modulus.

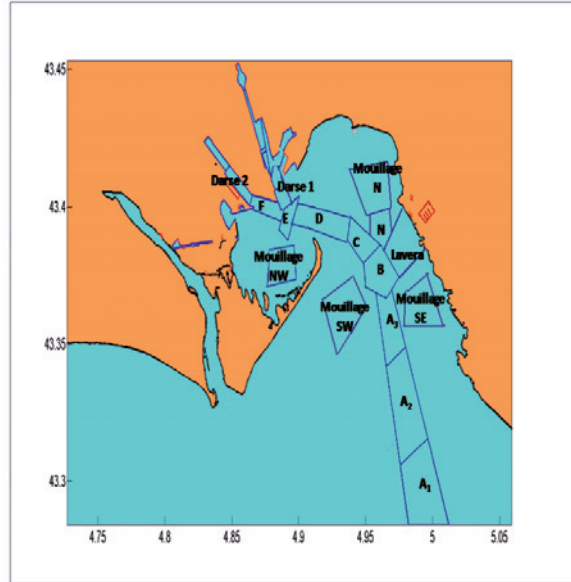


FIGURE 2 – Les zones fonctionnelles constituant les chenaux d'accès au port qui sont délimitées par la législation

tégories de comportements décrites. Nous avons construit ces deux sous-modules par deux méthodes différents :

- le BPAM a été conçu à partir de l'approche probabiliste Modèle de Markov Caché (MMC).
- le BRAM est basé sur l'approche réactive synchrone traduite par le langage ESTEREL.

L'ensemble de ces deux sous-modules constituent les composantes principales de notre BAM.

Ce manuscrit s'organise de la manière suivante :

Dans le chapitre 1, nous mettrons en évidence le dynamisme d'une zone portuaire à travers les différents services qu'elle peut offrir. Nous étudierons également les principaux risques auxquels un port est exposé et citerons les principaux dispositifs de sécurité qui existent actuellement dans tous les ports mondiaux et notamment dans le GPMM qui est notre site d'expérimentation. Après avoir démontré la faiblesse des méthodes actuelles de sécurité, nous présenterons le système des nouveaux capteurs SECMAR tel qu'il a été conçu par nos partenaires industriels du projet.

Etant donné que le module que nous avons conçu pour aider la surveillance dans

le GPMM est basé sur la technique de reconnaissance de comportements, nous introduirons dans le chapitre 2 un aperçu de cette technique, certains de ses domaines d'application ainsi que les différentes méthodes utilisées. Dans le cadre de notre problématique qui, à notre connaissance, n'a pas été traité auparavant, nous avons opté, comme nous l'avons indiqué précédemment pour deux méthodes qui ont été utilisées séparément dans la majorité des domaines d'application. Il s'agit d'une approche probabiliste par MMC et d'une approche réactive synchrone par ESTEREL.

Nous introduirons ces approches dans le chapitre 3. Dans une première section de ce chapitre, nous nous intéresserons au raisonnement probabiliste, aux différentes approches probabilistes liées au MMC puis nous exposerons la théorie des MMC. La section suivante sera consacrée à l'approche réactive synchrone, puis au langage ESTEREL, sa sémantique intuitive et sa sémantique opérationnelle.

La conception du BAM à partir de ces deux approches sera présentée dans le chapitre 4. Nous discuterons également, des résultats obtenus et de la complémentarité des deux sous-modules du BAM conçus par les deux approches.

Chapitre 1

La sécurité des zones portuaires face aux risques

Introduction

Nous traiterons dans ce chapitre des différentes raisons qui ont motivées nos travaux de recherche sur la problématique sécuritaire dans une zone portuaire. Nous étudierons le dynamisme d'un port à travers ses services, les risques qui le menacent ainsi que les dispositifs de sécurité actuels. Ceci sera présenté respectivement dans les sections 1.1, 1.2 et 1.3.

Par la suite, nous introduirons, dans la section 1.4, le GPMM et nous mettrons en évidence les manques des dispositifs de sécurité actuels puis nous présenterons dans la section 1.5 le nouveau système de capteurs SECMAR ainsi que notre module d'analyse de comportement qui prendra en entrée les données issues de ces capteurs.

1.1 Les zones portuaires et les services offerts

Une zone portuaire ou une Zone Industriale-Portuaire (ZIP) est un espace côtier qui associe à la fois des activités industrielles et portuaires. Il comprend d'une part une zone du port en eau profonde spécialement conçue pour accueillir d'énormes bateaux chargés en marchandises (conteneurs ou matières premières) et d'autre part une plate-forme multimodale qui permet de répartir ces marchandises, de façon rapide et de les acheminer par d'autres moyens de transport, vers des lieux différents : par exemple, des lieux de consommation pour le gaz, le charbon ou les denrées

alimentaires, des lieux de raffinage (cas du pétrole) ou encore des lieux de stockage (cas de certains conteneurs).

Une zone portuaire se caractérise tout d'abord par la taille et l'étendue de son port : établie sur des zones de plusieurs dizaines de kilomètres carrés, elle comporte de nombreux et profonds chenaux, prolongés par de profondes et larges darses (ou bassins) creusées et bétonnées pour pouvoir accueillir des navires de très gros tonnage sur plusieurs quais appelés terminaux.

Elle se caractérise également par le dynamisme de sa plate-forme multimodale. En effet, dès leur arrivée à quai, les portes-conteneurs déchargent leurs marchandises dont le transit est facilité par la proximité immédiate ou quasi-immédiate d'infrastructures aussi bien routières, autoroutières que ferroviaires.

Actuellement, la fonction industrielle des ports est en plein accroissement, ce qui leur a permis de jouer un rôle prépondérant dans l'économie mondiale et ceci grâce aux différents services qu'ils peuvent offrir :

1. les services de transport maritimes de passagers ;
2. les services de lignes régulières pour les marchandises ;
3. les services de transport de marchandises en vrac. Le vrac désigne des marchandises qui ne sont pas emballées à savoir les céréales, le minerai de fer, le charbon, etc.

Au 20^{ème} siècle, le service de transport maritime de passagers a perdu beaucoup d'importance au profit de l'aviation commerciale. Néanmoins, ce mode de transport reste la base du transport de marchandises à l'échelle internationale, ce qui a facilité le commerce international et a fortement contribué à son essor.

Selon un rapport publié en 2010 par le ministère de l'écologie, du développement durable, des transports et du logement sur l'analyse de l'offre et la demande du transport maritime, les échanges mondiaux de marchandises n'ont pas cessé de croître depuis 1970 (voir figure (1.1)).

Selon ce même rapport, 3 grandes tendances sont à retenir concernant l'évolution de ces échanges :

- Une croissance stable au cours des 10 dernières années avec un taux de croissance annuel moyen de près de 5%.
- Une baisse des volumes pour la période 2008-2009.
- Une évolution relativement similaire pour tous les types de marchandises (pétroles, céréales, charbon, pétroles, minerai de fer, conteneurs, gaz).

En somme, la figure (1.1) confirme la place occupée par les ports dans l'expansion du commerce international, surtout entre la fin du 20^{ème} et le début du 21^{ème} siècle.

Nous exposerons dans la section suivante les différents services offerts par une

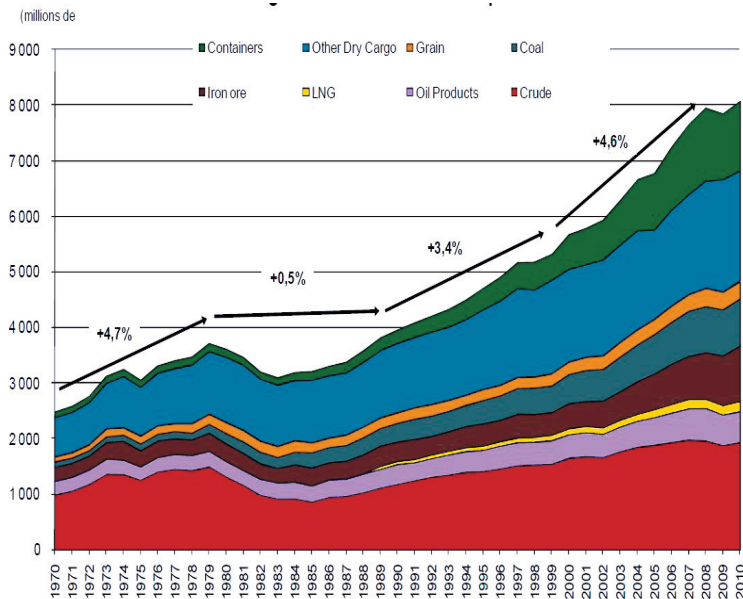


FIGURE 1.1 – Echanges maritimes mondiaux depuis 1970

zone portuaire qui participent à ces échanges maritimes. Il s'agit du service de lignes régulières et du service de transport de marchandises en vrac. Les différentes informations sur les services de lignes pour les marchandises ont été recueillies sur le site web de l'OCDE¹, de la CNUCED² et du ministère du développement de l'écologie, du développement durable, des transports et du logement³.

1.1.1 Les services de lignes régulières pour les marchandises

Ce sont des services réguliers, entre ports, programmés et connus à l'avance (trafics de lignes régulières). Ces services assurent le transport de marchandises autres qu'en vrac.

Les marchandises peuvent être transportées séparément (cas des véhicules automobiles) ou en chargements unitisés dans les conteneurs. Différents types de navires participent au trafic de lignes régulières : navires de marchandises générales, porte-conteneurs, navires réfrigérés, navires polyvalents et rouliers (« ro-ro », dans lesquels

1. Organisation de Coopération et de Développement Economique : <http://www.oecd.org>.

2. La CNUCED est la Conférence des Nations Unies sur le Commerce et le Développement : <http://www.unctad.org>.

3. www.developpement-durable.gouv.fr/

les camions et les remorques sont embarqués par leurs chauffeurs) et des navires spacieux comme les transporteurs de voitures. Jusqu'aux années 60, les navires de marchandises générales étaient prépondérants sur les lignes régulières mais l'apparition de cargaisons unitisées a été suivie par l'expansion massive du trafic conteneurisé. Aujourd'hui, il représente la majeure partie du transport maritime de marchandises dans les pays développés.

En effet, grâce à la naissance du porte-conteneur en 1956 aux Etats-Unis, le trafic mondial de marchandises conteneurisées s'est accru rapidement, passant selon les estimations⁴, de 80 à 185 millions d'EVP⁵ entre 1989 et 1999, soit une croissance annuelle d'environ 10 %. Il a continué à croître de façon importante pour s'établir en 2010 à 517 millions d'EVP, selon le CNUCED.

1.1.2 Les services du transport de marchandises en vrac

Le service de transport maritime en vrac assure en général le transport de cargaisons uniques en grandes quantités. Les marchandises transportées de la sorte se classent en quatre groupes principaux :

- les vracs secs : les matières premières industrielles, c'est-à-dire le charbon, le minerai de fer, le bauxite/alumine et le phosphate, ainsi que les denrées alimentaires telles que les céréales ;
- le pétrole : le pétrole brut, les produits raffinés ;
- les gazs liquéfiés : GNL et GPL ;
- les produits chimiques liquides.

Le commerce de vrac sec, qui représente environ un quart du commerce maritime total, a progressé de 1.4% en 2009. Néanmoins, ce chiffre cache des écarts considérables. Le bauxite et l'alumine, qui sont indispensables à la production d'aluminium et sont principalement utilisés dans les industries de transports et de la construction, ont reculé de 23.2%. Les roches phosphatées, utilisées pour la fabrication d'engrais dans l'agriculture, ont reculé de 38.7%. Les expéditions de minerai de fer se sont établies à 907 millions de tonnes environ en 2009, soit une hausse de 7.8% par rapport à l'année d'avant.

Le secteur pétrolier a également connu des évolutions positives au cours de la récente décennie : le volume des cargaisons acheminées dans le monde a été plus élevé qu'avant, et les distances moyennes d'acheminement ont augmenté. Le tableau

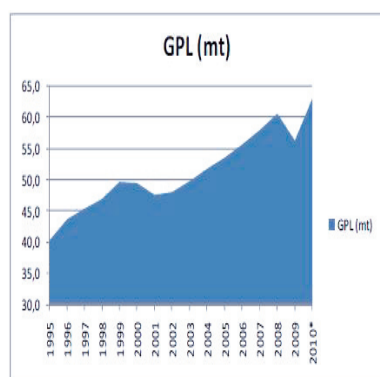
4. Sources : "Containerisation International", Yearbooks and UNCTAD Review of Maritime Transport, 2000.

5. Le TEU (twenty foot equivalent unit) ou l'EVP (Equivalent Vingt Pieds) en français est l'unité du conteneur.

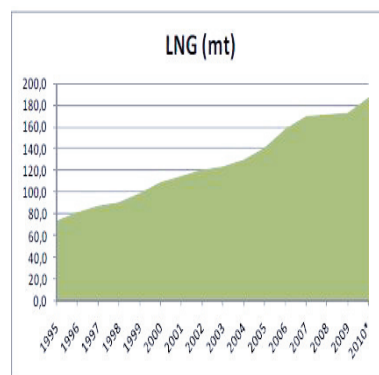
(1.1)⁶ illustre l'évolution du trafic maritime de pétroles brut où on peut voir que ce secteur a évolué de 7% entre 2005 et 2010 malgré une légère diminution enregistrée entre 2005-2006 et 2008-2009. Notons aussi que le trafic maritime de brut a connu une baisse de 0.7% au niveau global à cause de la crise économique mondiale qui s'est installée à partir de 2008.

	Trafic maritime de brut (Mt)	Variation (%)
2005	1720	-1.9
2006	1756	2.1
2007	1775	1.1
2008	1800	1.4
2009	1787	-0.7
2010	1850	3.5

TABLE 1.1 – L'évolution du transport maritime de pétrole brut



(a) Evolution du transport maritime de GPL



(b) Evolution du transport maritime de GNL

FIGURE 1.2 – Evolution du trafic de transports maritimes en GPL et en GNL.

Dans un autre secteur du transport maritime en vrac, la figure (1.2)⁷ nous montre que les échanges mondiaux de GNL et GPL ont également connu une forte croissance entre 1995 et 2010. On voit sur la figure (1.2a) que les volumes transportés en GPL

6. www.developpement-durable.gouv.fr.

7. source : <http://www.developpement-durable.gouv.fr>.

ont évolué de 55 % au cours de la même période malgré une inflexion de plus de 7% enregistrée entre 2009 et 2010. Cette forte demande de GPL est liée aux besoins en chauffage et de production industrielle et est donc fortement corrélée à l'activité économique, ce qui explique cette baisse en 2009. Le Moyen Orient domine les exportations en GPL devant l'Europe du nord et certaines autres pays comme l'Indonésie et l'Australie .

Quant aux services de transport en matière de GNL, ils ont fortement augmenté, enregistrant une croissance de 171% (voir figure (1.2b)) entre 1995 et 2010. Les principaux exportateurs de GNL sont situés dans des régions en développement, le Qatar étant le leader mondial dans cette production (17.5%). Parmi les autres exportateurs, on compte l'Algérie, l'Indonésie, la Malaisie et le Nigéria.

Compte-tenu de l'essor du transport maritime de marchandises et de la place qu'il occupe dans le commerce international, les ports sont renforcés dans leur rôle de lieux centraux et d'acteurs majeurs de la mondialisation. Par ailleurs, les différents services offerts par le port de chaque pays montrent le rôle stratégique qu'il joue pour l'ensemble de son économie. En effet, les activités maritimes sont vitales pour chaque pays et n'importe quel scénario de crise peut engendrer des conséquences économiques et humaines très graves. Nous nous sommes alors intéressés dans la section suivante aux différents risques qui menacent une zone portuaire, à savoir les activités criminelles et les actes terroristes afin de souligner le besoin d'assurer sa sûreté.

1.2 Les risques existants dans une zone maritime

Les événements du 11 septembre 2001 ont attiré l'attention sur la vulnérabilité de tous les moyens de transport à plusieurs types de menace. Concernant le domaine maritime, la marine nationale française, qui est chargée d'intervenir en cas de risques majeurs, a identifié des scénarios de crises possibles du domaine maritime. Ces scénarios sont inventoriés sur le site web du sénat⁸. Ils peuvent concerner la pollution maritime, la pêche illicite, les trafics illicites ou encore le terrorisme maritime.

Dans la mesure où nos travaux ont pour objectif d'apporter une aide pour la prévention d'une zone portuaire contre le terrorisme maritime, nous nous intéresserons particulièrement à ce dernier scénario de crises.

8. <http://www.senat.fr/rap/r04-418/r04-4182.html>.

Comment définit-on le terrorisme maritime ? Selon la Convention de Rome de 1988 (cf. article sur la répression d'actes illicites contre la sécurité de la navigation maritime), le terrorisme maritime peut être assimilé à une infraction pénale. Ainsi, est considérée en infraction pénale, toute personne qui :

- *Illicitement s'empare d'un navire ou exerce le contrôle par violence ou menace de violence.*
- *Accomplit un acte de violence à l'encontre d'une personne se trouvant à bord d'un navire, si cet acte est de nature à compromettre la sécurité de la navigation du navire.*
- *Détruit un navire ou cause à un navire ou à sa cargaison des dommages[...].*
- *Place ou fait placer sur un navire un dispositif ou une substance propre à causer au navire ou à sa cargaison des dommages.*

Les principales formes que pourraient revêtir cette menace dans le domaine maritime sont les suivantes :

1. détournement d'un navire et prise d'otages ;
2. destruction d'un navire par engin explosif ou incendie criminel ;
3. utilisation du navire lui-même comme une arme ;
4. blocage d'un port ou d'une voie de communication ;
5. infiltration de terroristes via des navires de commerce ;
6. manipulation criminelle de cargaison (conteneurs).

Nous effectuons une brève analyse des trois premiers scénarios de crise de cette liste :

Détournement d'un navire et prise d'otages

Le détournement de navires peut servir des causes et des intérêts bien différents :

- la piraterie qui sert des fins privées (par exemple, s'approprier les biens de passagers, demande de rançon contre la libération des otages, etc) ;
- le terrorisme maritime qui sert des fins idéologiques (par exemple, libération d'un leader terroriste, pression politique sur un Etat, etc).

Dans tous les cas, les modes d'action restent semblables : il s'agit pour les pirates de se rendre maître d'un navire en route ou au mouillage, aussi bien à proximité de la côte qu'en pleine mer.

Le tableau 1.2 reprend les principaux chiffres sur les actes de pirateries publiés par le Bureau Maritime International (BMI⁹) depuis 2003, ce qui permet de dresser

9. Le BMI est une division spécialisée de la Chambre de Commerce International (CCI). Il s'agit d'une organisation à but non lucratif établie en 1981 pour agir contre tous types de crimes et fraudes maritimes.

un constat nuancé de l'évolution quantitative de la piraterie. En effet, si le nombre global des attaques a baissé entre 2003 et 2007, une nette remontée est observée depuis cette date, en raison d'un fort accroissement des attaques en Somalie et dans le golfe d'Aden ainsi qu'au large du Nigéria.

	2003	2004	2005	2006	2007	2008
Total des attaques	445	329	276	239	263	293
<i>dont :</i>						
Indonésie	121	94	79	50	43	28
Détroit de Malacca	28	33	12	11	7	2
Nigéria	39	28	16	12	42	40
Golf d'Aden	18	8	10	10	13	92
Somalie	3	2	35	10	31	19

TABLE 1.2 – Bilan des actes de pirateries perpétrés dans le monde

Suite à l'analyse de plusieurs actes de pirateries répétés qui ont eu lieu en mer, il a été possible de dresser une liste approximative des éléments ayant favorisé l'attaque des pirates (voir figure (1.3)) :

- nature du navire : cas des navires de commerce, navires de tourisme en transit à vide, etc. Il s'agit de navires dont le nombre de membres d'équipage est réduit ;
- configuration de la cible : vitesse, hauteur du franc-bord, accès par l'arrière, capacités manoeuvrières ;
- environnement : rail commercial, zone de mouillage, détroits, hauts fonds ou dangers pour la navigation, nuit, météo, zone de pêche, proximité de la côte.



FIGURE 1.3 – Abordage d'un navire par des pirates armés

On a également pu dresser la liste des comportements suspects pouvant augurer d'un assaut par des pirates :

- embarcations de petites tailles à faibles écho Radar ;
- personnes armées à bord d'embarcations ;
- présence d'échelles dans les embarcations ;
- présence de flottille de pêche dans une zone à priori non propice à la pêche ;
- approche soudaine d'un groupe d'embarcations rapides au passage d'une zone côtière particulière comme un détroit ou un village.

Destruction d'un navire par un engin explosif ou un incendie criminel

Ce mode d'action terroriste est applicable aussi bien à un navire en route qu'à un navire en mouillage, voire à une installation portuaire ou côtière.

Il va s'agir dans ce cas pour les terroristes de toucher, avec un investissement relativement faible (une petite embarcation chargée d'explosifs) une cible ayant une forte valeur politico-stratégique (bâtiment de guerre, navire de passagers) ou commerciale (super tanker, terminal de chargement). Notons qu'un terroriste ayant l'intention de poser une bombe pourra avoir embarqué (officiellement ou clandestinement) avant l'appareillage.

On peut citer à ce titre, les deux événements marquants qui ont eu lieu dans les années 2000 :

- une attaque à la bombe contre le pétrolier Limburg Ash Shihr au Yémen en octobre 2002 (voir figure (1.4a)) ;
- un attentat suicide contre le navire américain USS Cole en octobre 2000 (voir figure (1.4b)).



(a) Attaque à la bombe contre le pétrolier Limburg



(b) Un attentat suicide contre le navire américain USS Cole en octobre 2000

FIGURE 1.4 – Exemples de scénarios de crise dans un espace maritime

Utilisation d'un navire lui-même comme arme

Dans ce cas, il s'agit de navires de forts tonnages lancés contre des objectifs à terre ou en mer. Notons que deux cas de figures sont alors envisageables (ce genre de scénarios n'a jamais été recensé, mais fait partie des scénarios plausibles) :

1. Collision volontaire avec un navire cible en mer (navire militaire, navire à passager, navire à transport d'hydrocarbures/gaz). Il s'agira d'une cible à fort impact médiatique, stratégique ou commercial. Les zones géographiques les plus propices à ce mode d'action sont :
 - (a) zones à forts trafics (accès aux ports, en particulier aux terminaux pétroliers) ;
 - (b) passages obligés (détroits) ;
 - (c) ports.
2. Collision volontaire avec un objectif à terre. Les zones géographiques sensibles à cette menace seront :
 - (a) zones de productions portuaires (cas du golfe de Guiné, golfe arabo-persique) ;
 - (b) zones portuaires majeures : par exemple le blocage du port de Nantes qui est le plus grand port céréalier français, handicaperait lourdement l'industrie agro-alimentaire française.

Compte tenu de l'importance des zones portuaires et des menaces auxquelles elles peuvent être exposées, les gouvernements de tous les pays ainsi que les organisations internationales se sont mobilisées pour mettre en place des systèmes de sécurité fiables et efficaces. Nous nous intéresserons dans la section suivante aux types de mesures et de systèmes de sécurité qui ont été développés en la matière.

1.3 Sécurisation des ports et de leurs équipements

Méthode américaine Après les événements du 11 septembre 2001, le gouvernement américain a décidé de protéger son espace maritime en mettant en place les mesures de sécurité suivantes :

1. C-TPAT : le *Customs-Trade Partnership Against Terrorism* qui est une certification, par laquelle tous les participants de la chaîne d'approvisionnement coopèrent avec la douane américaine afin d'élaborer des méthodes sécuritaires pour chacune des phases de leurs opérations. Le programme répond aux préoccupations de sécurité contre les menaces terroristes lors de l'importation de marchandises aux États-Unis.
2. CSI : le *Container Security Initiative* est un programme lancé en janvier 2002 par le *US Bureau of Customs and Border Protection* (CBP¹⁰). Il vise à s'assurer de la sécurité de la chaîne logistique et à contrôler les conteneurs et embarcations comportant des risques potentiels.
3. ISPS : le *International Ship and Port Security* est un code élaboré par l'Organisation Maritime Internationale (OMI¹¹) et proposé par les États-Unis. Il a été intégré en 2002 à la convention *Safety For Life and Security* (SOLAS¹²) et est entré en vigueur pour tous les navires depuis juillet 2004. Il s'agit d'un code associé à tous les navires et aux équipements du port. Il vise à éviter le transport et le débarquement de matériaux permettant des actes terroristes ainsi que des personnes malveillantes sur le territoire. Ce code est évalué individuellement par chaque pays selon les risques auxquels sont exposés leurs ports.

10. CBP ou en français le service des douanes et de la protection des frontières des États-Unis est une agence du département de la sécurité intérieure des États-Unis.

11. L'OMI est une institution spécialisée des Nations Unies. Elle compte 170 États membres et 3 membres associés en 2011. Son siège est situé Londres.

12. SOLAS est un traité international visant à définir différentes règles relatives à la sécurité.

Méthode européenne Pour les mêmes enjeux sécuritaires, la communauté européenne a également adopté plusieurs mesures de sécurité notamment :

1. Les normes SAFE¹³ : il s'agit de normes mises en place par l'Organisation Mondiale des Douanes (OMD¹⁴) en juin 2005. Elles visent à offrir aux services douaniers et à leurs partenaires commerciaux un cadre structuré destiné à sécuriser la chaîne logistique internationale et à faciliter les mouvements légitimes de marchandises.
2. Les systèmes ICS pour *Import Control System* et ECS pour *Export Control System* : ce sont des réglementations mises en place par l'Union Européenne (UE) dans le cadre des normes SAFE qui visent à protéger les importations et les exportations en provenance de tous les pays hors UE. Cette mesure a pris effet officiellement le 1er juillet 2010 et est devenue obligatoire à partir du 1er janvier 2011.
3. La directive Seveso : cette directive européenne impose aux États membres de l'union européenne d'identifier les sites industriels présentant des risques d'accidents majeurs. Cette directive est nommée ainsi d'après la catastrophe de Seveso¹⁵ qui eut lieu en Italie (1976) et qui a incité les États européens à se doter d'une politique commune en matière de prévention des risques industriels majeurs.

Méthode française Dans ce même contexte, la France a également décidé de renforcer ses dispositifs de sécurité et de sûreté maritimes :

1. Le plan VIGIPIRATE est un dispositif de sécurité français destiné à prévenir les menaces ou à réagir face aux actions terroristes. Créé en 1978 sous la présidence de Valéry Giscard d'Estaing alors que l'Europe connaissait une vague d'attentats, ce plan a été actualisé à trois reprises, en juillet 1995, juin 2000 et mars 2003. Il a été déployé pour la première fois en 1991.

13. Le terme SAFE est celui d'un câble sous-marin de dernière génération reliant par fibre optique Melkbosstrand en Afrique du Sud à Penang en Malaisie. Son nom est l'acronyme de South Africa - Far East, soit celui de ses deux extrémités en anglais.

14. L'OMD est une organisation intergouvernementale, qui a son siège à Bruxelles, en Belgique. Elle est réputée pour élaborer des conventions internationales, des instruments et des outils. A titre d'exemple, cette organisation peut traiter la sécurité des chaînes logistiques, la lutte contre la fraude, la facilitation des échanges internationaux, etc.

15. La catastrophe de Seveso s'est produite le 10 juillet 1976. Un nuage contenant de la dioxine s'échappe d'un réacteur de l'usine chimique Icmesa, située dans la commune de Meda, et se répand sur la plaine lombarde en Italie. Quatre communes, dont Seveso, ont été touchées.

2. La Doctrine Nationale de Sûreté Maritime et Portuaire (DNS) a été établie à la demande du cabinet du premier ministre en 2004. Cette doctrine vise à donner une cohérence entre les dispositifs relevant du cadre communautaire et international et le dispositif français de lutte contre le terrorisme notamment avec le plan VIGIPIRATE et sa transposition en mer VIGIMER. La mission du DNS porte principalement sur la mise en oeuvre des dispositifs et des programmes nécessaires pour assurer la sécurité du transport et des ports maritimes français.
3. Le Peloton de Sûreté Maritime et Portuaire (PSMP) est une nouvelle unité de gendarmerie maritime qui vient compléter le dispositif inter-administrations de surveillance des approches maritimes. Le PSMP permet d'assurer la cohérence du dispositif terre-mer, puisque les gendarmes maritimes ont cette double compétence. Une première unité a été créée au Havre, à titre expérimental en 2006. L'expérience a été concluante, et il a été décidé d'étendre le dispositif à Marseille. Implanté sur la commune de Port de Bouc, à l'entrée de l'Étang de Berre et du Golfe de Fos, le Peloton a une situation idéale pour surveiller le bassin ouest marseillais. Le PSMP y assure la protection contre les actions terroristes, le transport illicite de marchandises, l'immigration irrégulière et les autres actes de malveillance de droit commun. Dans les années à venir, des PSMP verront le jour dans deux grands ports des différentes façades maritimes : à Dunkerque et à Nantes-Saint-Nazaire.

Systemes de détection En plus des mesures de sécurité, certains systemes de détection, de suivi et d'identification de position ont été développés, nous citerons certains d'entre eux :

1. L'*Automatic Identification System* (AIS) : A partir de 2007, l'OMI a imposé, au travers de la convention SOLAS, que tous les navires civils de plus de 300 tonneaux, destinés à la navigation internationale, soient équipés de l'AIS. Les navires militaires ainsi que les petits navires (voiliers, jets-ski) n'ont pas cette obligation. Ce systeme permet de transmettre en permanence, depuis l'émetteur embarqué vers une station réceptrice (à terre ou embarquée), des informations de position, route et vitesse (l'émetteur est couplé à un systeme de positionnement par satellite (GPS¹⁶), un ordinateur et un émetteur très haute fréquence (VHF¹⁷)) ainsi que des informations sur l'identité du navire. Par ailleurs, toutes les 6 minutes, l'émetteur AIS diffuse des données permettant l'identification des navires telles que le nom du navire, la destination et l'heure

16. Global Positioning System.

17. Very High Frequency.

estimée d'arrivée, le type de bâtiment ou de fret, etc. Notons que l'émetteur AIS, étant accessible depuis le bateau, il est possible d'en stopper l'émission, voire de modifier les données émises et enregistrées.

2. Le *Long Rang Identification and Tracking* (LRIT) : ce système, utilisant le système satellitaire, est le pendant du système AIS au-delà des zones couvertes par la VHF. La mise en place de ce système de localisation et d'identification à grande distance est réglementé par l'OMI, conformément au chapitre 5 de la convention SOLAS. Il est entré en vigueur en 2008. Ainsi, les Navires à Passagers (NAP) de plus de 300 tonnes sont équipés de ce système et doivent transmettre par satellite toutes les 6 heures un message LRIT indiquant leur identifiant, position, date et heure. Contrairement à l'AIS, ce message sera diffusé avec restriction, et ne s'adressera qu'à :
 - (a) l'Etat du pavillon, sans restriction pour les navires portant son pavillon dans le monde ;
 - (b) l'Etat du port de destination du navire ;
 - (c) les Etats côtiers par lesquels circulent le bateau dans une zone allant jusqu'à 1000 m de leurs côtes.
3. Le *Vessel Monitoring System* (VMS) : il s'agit d'un moyen de suivi de la position de navires de pêches. Ce système satellitaire permet à une unité à terre de recueillir la position des navires de pêches dans une zone donnée et ainsi de contrôler le respect des zones de pêches.
4. Le *Vessel Traffic System* (VTS) : c'est le système initialement adopté par SOLAS et l'OMI en novembre 1997 afin d'améliorer la sécurité et l'efficacité de la navigation dans un environnement marin. Il s'agit d'un système de surveillance du trafic de navires installé par les autorités du port. Ce système est similaire au système de contrôle du trafic aérien pour les avions. Un système VTS utilise un ensemble d'éléments : le Radar, la vidéo surveillance, l'émetteur VHF et le système AIS, afin d'assurer le suivi des mouvements de bateaux ainsi que la sécurité de la navigation dans une zone géographique limitée. A partir des données en provenance de plusieurs capteurs placés dans des zones différentes du port, le système VTS collecte les informations des capteurs pour fournir une image en temps réel du navire dans la zone de surveillance en question. L'évaluation de ces données dépend fortement de leur qualité et de la capacité de l'opérateur à les interpréter. Cet opérateur doit être suffisamment formé et qualifié pour pouvoir interpréter et manipuler correctement ces données.

Un guide d'information sur les systèmes VTS internationaux de plusieurs pays est disponible sur le web : www.worldvtsguide.org. Ce guide a été conçu pour les personnels de la navigation maritime, et pour toute autre personne intéressée, afin de leur fournir des informations sur les différents centres VTS dans le monde.

Le document de la figure (1.5) est un exemple des règles et des services du VTS du port néerlandais de Rotterdam, actuellement premier port européen et ancien premier port mondial jusqu'en 2005.

THE NETHERLANDS	ROTTERDAM	AREA PROCEDURES	EDITION No. 003	REVISION DATE 18-Nov-2009	PAGE No. RTM 1																										
<p align="center">GENERAL RULES AND REGULATIONS</p> <p>ENTRY ADMITTANCE: The Port Authority may refuse entry to ships with substantial safety deficiencies affecting in particular their navigation, radio, fire fighting, life saving and other safety equipment.</p> <p>VTS Area: Extends for 38 nm seawards of the port entrance to 2 nm east of the Van Brieneoord Bridge and 4 nm East of the Spijkenisse Bridge across the Oude Maas river, including all adjacent harbour basins. The VTS Area is divided into 12 sectors, each sector having its own dedicated VHF channel. See pages RTM 2, 5 and 6.</p> <p>VTS SERVICES: In order to control overall communications, the VTS operator will give information only when:</p> <ul style="list-style-type: none"> • Ships are in each other's vicinity, but have no visual contact • Ships intend to carry out a manoeuvre that deviates from the usual traffic pattern • Traffic guidance is considered necessary by Rotterdam VTS. More extensive traffic information can be expected during calamities, in the event of defects, and in difficult navigational or meteorological circumstances (e.g. visibility less than 1,000 metres on the river, or less than 2,000 metres in the approach area and Europort. • Navigational information, such as position, is requested: <ul style="list-style-type: none"> - In the approach area: a description of traffic in the vicinity and a warning of any risk of running aground - In each sector: longitudinal, transverse, and moving information in relation to common reference points, possibly with additional information on passing distances, sailing speeds and traffic. <p>For further information please refer to the Port Information Guide which may be found at: www.portofrotterdam.com/en/shipping/seashipping/</p>			<p align="center">VHF PROCEDURES</p> <p>Sectors: Rotterdam VTS consists of:</p> <table border="1"> <thead> <tr> <th>Sector</th> <th>VHF Channel</th> </tr> </thead> <tbody> <tr><td>Maas Approach</td><td>01</td></tr> <tr><td>Pilot Maas</td><td>02</td></tr> <tr><td>Maas Entrance</td><td>03</td></tr> <tr><td>Ship-Ship</td><td>10</td></tr> <tr><td>Rozenburg</td><td>65</td></tr> <tr><td>Europort</td><td>66</td></tr> <tr><td>MaasLuis</td><td>80</td></tr> <tr><td>Botiek</td><td>61</td></tr> <tr><td>Oude Maas</td><td>62</td></tr> <tr><td>Eemhaven</td><td>63</td></tr> <tr><td>Waalhaven</td><td>60</td></tr> <tr><td>Maasbruggen</td><td>81</td></tr> </tbody> </table> <p>(Shown on pages RTM 2, 6 and 7)</p> <p>OTHER CONTACTS</p> <p><i>Harbour Master's Reporting Centre:</i> VHF: CH 14 Tel: +31 10 252 10 00 Fax: +31 10 252 16 00 Email: CMH@portofrotterdam.com</p> <p><i>Vessel Traffic & Operations</i> Emergencies, ETD updates and listen out when alongside, etc.: VHF: CH 19 Tel: +31 10 252 24 00 Fax: +31 10 252 14 00 Email: hccv@portofrotterdam.com</p>			Sector	VHF Channel	Maas Approach	01	Pilot Maas	02	Maas Entrance	03	Ship-Ship	10	Rozenburg	65	Europort	66	MaasLuis	80	Botiek	61	Oude Maas	62	Eemhaven	63	Waalhaven	60	Maasbruggen	81
Sector	VHF Channel																														
Maas Approach	01																														
Pilot Maas	02																														
Maas Entrance	03																														
Ship-Ship	10																														
Rozenburg	65																														
Europort	66																														
MaasLuis	80																														
Botiek	61																														
Oude Maas	62																														
Eemhaven	63																														
Waalhaven	60																														
Maasbruggen	81																														

FIGURE 1.5 – Règles concernant le VTS

Après avoir fait un revue de l'ensemble des dispositifs de sécurité portuaire mis en place à l'échelle internationale ainsi que leurs intérêts, nous nous intéresserons dans la section suivante au GPMM comme étant le site d'expérimentation de nos travaux. Nous étudierons brièvement ce port ainsi que son dispositif de sécurité actuel afin de mettre en évidence leur faiblesse. Par la suite, nous justifierons le besoin d'un nouveau type de système plus robuste pour compléter les dispositifs déjà existants et nous introduirons, dans la section 1.5, la solution que nous proposons afin de remédier à ces faiblesses.

1.4 La sécurité du Grand Port Maritime de Marseille (GPMM)

1.4.1 Le GPMM : un port d'importance vitale pour la France

Le Grand Port Maritime de Marseille (GPMM), anciennement Port Autonome de Marseille (PAM)¹⁸ est composé de deux ports distants de plus de 50 km : à l'est, les bassins de Marseille, localisés dans la ville et s'étendant sur 400 hectares ; à l'ouest, le port de Fos sur mer, fer de lance d'une zone industrialo-portuaire, s'étendant sur près de 10 000 hectares. L'activité de ces deux ports n'est pas comparable : en 2008, Fos a traité un trafic onze fois plus important que Marseille, et près de trois fois plus important hors hydrocarbures. Il a réalisé, la même année, un chiffre d'affaires trois fois supérieur à celui de Marseille et légèrement supérieur hors hydrocarbures. En revanche, le port de Marseille est très actif dans le secteur « passagers » avec un trafic de plus de 2 millions de voyageurs en 2008.

Le GPMM ne manque pas d'avantages face à ses principaux concurrents européens :

1. il occupe une place stratégique puisqu'il se situe sur la route maritime reliant l'Asie à l'Europe par le canal de Suez ;
2. le coût du passage d'un navire n'y est pas particulièrement élevé, par comparaison avec ses principaux concurrents ;
3. le GPMM est l'un des ports les plus faciles d'accès en Europe. La part du coût du chenalage et du pilotage dans le coût global du passage portuaire d'un porte-conteneur s'élève à 8,1 % au GPMM, contre 33,5 % à Anvers, par exemple ;
4. il est desservi, à Fos et à Lavera, par un réseau d'oléoducs (une canalisation destinée au transport du pétrole) à vocation européenne, qui lui confère une place privilégiée dans le secteur des hydrocarbures ;
5. enfin, contrairement à certains ports européens qui se situent exactement entre la mer et la montagne à savoir ceux de Gênes ou de Trieste, son emplacement lui a permis de bénéficier de vastes réserves foncières.

Le GPMM accueille tous types de marchandises. Par exemple en 2009, il a accueilli :

- 57.07 Mt en hydrocarbures ;
- 8.46 Mt en vrac solide ;
- 3 Mt en vrac liquide ;

18. source sur le GPMM : www.ccomptes.fr

– 14.5 Mt en marchandises diverses et conteneurs.
 en plus d'un important trafic de passagers (2.07 millions de passagers en 2009 dont 639000 de passagers de croisière).

D'après le tableau (1.6) et pour la période de 2008 à 2010, le GPMM a occupé la 4^{ème} place en matière de trafic de conteneurs derrière les ports de Rotterdam, d'Anvers et de Hambourg.

	2008		2009		2010 - semestre 1	
	Trafic total (1)	Trafic conteneurs (2)	Total	Conteneurs	Total	Conteneurs
Marseille	96	0,851	83	1	44	0,510
Rotterdam	421	10,8	387	10	213	5,4
Anvers	189	8,7	158	7	87	4,2
Hambourg	140	9,7	110	7	59	3,7
Le Havre	81	2,5	74	2	36	1,2
Valence	59	3,6	58	4	30	2,0
Barcelone	51	2,6	42	2	20	0,9
Gênes	54	1,8	47	2	25	0,9
La Spezia	19	1,2	14	1	9	0,6

1) Trafic total en millions de tonnes

2) Trafic conteneurs en millions Evp – Equivalents vingt pieds

FIGURE 1.6 – Trafic des principaux ports européens

Ce port joue un rôle primordial dans l'approvisionnement de toute la France en marchandises diverses. Par conséquent, le GPMM est considéré comme un port d'importance vitale pour le pays. Pour cette raison, il est indispensable d'assurer convenablement sa sécurité. Nous décrirons ainsi, dans les sections suivantes, les mesures de sécurité qui ont été mises en place, puis nous décrirons les faiblesses de ces mesures avant de mettre en évidence le besoin d'un système innovant et robuste afin de lui apporter une meilleure protection.

1.4.2 La sécurité du GPMM

De la même manière que pour les différents ports mondiaux, le GPMM est régi par sa propre méthode de surveillance de sa zone industrialo-portuaire.

Certaines zones du GPMM présentant un niveau de risque très élevé notamment les terminaux pétroliers, les méthaniers et les usines de pétrochimie, ont été classées Seveso.

Par ailleurs, le GPMM s'est inscrit dans le programme *Container Security Initiative* (CSI) depuis janvier 2005 et le terminal conteneurs de Graveleau est considéré comme la 34^{ème} installation agréée CSI. Les installations portuaires du GPMM concernées par l'application du code ISPS sont au nombre de 24 dont 15 privées et 9 publiques. Suite à l'entrée en vigueur en 2004 du code ISPS proposé par les Etats-Unis et appliqué en Europe, un PSMP, comme nous l'avons vu précédemment, a été implanté, en octobre 2009, au Port de Bouc, à l'entrée de l'Étang de Berre et du golf de Fos. Cette unité de gendarmerie assure des missions de protection contre les actions terroristes, le transport illicite de marchandises, l'immigration irrégulière et les autres actes de malveillance de droit commun en étant présente sur les plans d'eau ainsi qu'aux abords et dans les zones portuaires.

1.4.3 Le besoin d'un système de sécurité plus robuste

Après avoir étudié dans la section 1.3 les différents dispositifs de sécurité qui existent à l'échelle internationale, on a constaté que ces systèmes reposent principalement et uniquement sur trois processus :

1. des normes et des réglementations de sécurité sur les navires, marchandises ou ports (ISPS, ICS,...);
2. des systèmes de détection d'objets (fixes ou mobiles) et des activités marines dans le port (VTS, VMS,...);
3. un personnel suffisamment formé qui utilise les normes de sécurité et les interprète efficacement pour pouvoir agir.

Actuellement et de la même manière que tous les autres ports mondiaux, la sécurité dans le GPMM est effectuée en imposant à l'ensemble des navires, à l'ensemble des équipements et des bâtiments du port, des normes et des réglementations à suivre afin d'assurer la sécurité de l'espace maritime ainsi que des systèmes de détection de type AIS.

L'AIS qui est considéré actuellement comme le système de détection de navires le plus précis et le plus fiable dans tous les ports, présente néanmoins plusieurs inconvénients. En effet, au delà de pouvoir stopper l'émission des données issues de

ce système voire de les modifier, certains petits navires tels que les voiliers et les jet-skis peuvent délibérément choisir de ne pas installer ce système à bord. Dans les deux cas, les bateaux ne sont pas détectés pendant leurs mouvements.

De plus, les systèmes de détection actuels ne prennent pas en compte certains aspects de la surveillance, à savoir :

- le suivi en temps-réel de l'évolution de tous les bateaux (notamment une surveillance nocturne) ;
- le suivi du mouvement de bateaux par vidéo surveillance ;
- le traitement de l'entourage d'un bateau pour détecter un petit bateau à proximité (ceci contribuera à éviter qu'un gros bateau ne soit abordé par des pirates) ;
- la détection du milieu sous-marin.

Par conséquent, les personnels du port, qui ont pour mission l'analyse et l'interprétation des résultats de suivi en provenance des systèmes de détection, n'ont pas suffisamment d'informations sur l'ensemble des scénarios de comportements de navires dans le port pour en assurer de façon optimale la surveillance.

En outre, vu la taille importante du port et le grand nombre de navires qui peuvent être présents en même temps dans la zone maritime, la surveillance de l'espace maritime par des êtres humains est une tâche fastidieuse, complexe et insuffisante pour déceler rapidement un événement ou un scénario menaçant sachant que les gendarmes maritimes, présents 24h/24 sur site, sont prêts à intervenir en cas de dangers imminents.

Pour toutes ces raisons, deux besoins se sont imposés :

- le premier concerne l'équipement du port d'un nouveau système de détection qui doit être plus précis et plus innovant en termes de détection (telle que la détection sous-marine, la détection nocturne ou encore la détection d'un bateau masqué par un autre, etc) que ceux qui existent actuellement à savoir, les systèmes AIS, VTS, etc ;
- le deuxième concerne le traitement automatique des données en provenance de ces systèmes précis afin de reconnaître les comportements de navires au cours de leurs déplacements dans le port (comportement nominal de bateau se dirigeant vers une zone de mouillage, bateau qui va trop vite dans une zone interdite, etc).

L'ensemble capteurs/module automatique de reconnaissance doit apporter une aide à l'équipe de surveillance du GPMM, aux gendarmes entre autres, dans le but de mieux surveiller le port et d'agir efficacement en cas de dangers.

De plus, étant donné les enjeux concurrentiels du GPMM, la mise en place d'un système robuste de surveillance du port, incluant de nouveaux capteurs et un système d'analyse de comportements de conception innovante lui permettra de se démarquer des autres ports qui ne dispose pas d'un tel système. Cela constituera sans doute un atout commercial indéniable pour le GPMM.

La place importante qu'occupe un port, notamment le GPMM, sur le plan économique, écologique et social, les scénarios de crises qu'il peut affronter, ainsi que les besoins précédemment décrits, ont motivé la naissance du projet SECMAR que nous allons décrire dans la section suivante.

1.5 Le projet SECMAR

Nous avons mis en évidence, dans la section précédente, les manques des méthodes de sécurité actuelles face aux risques qu'un port peut encourir, ce qui a motivé la naissance du projet SECMAR. L'objectif principal de ce projet est de concevoir, réaliser, déployer et expérimenter un système de surveillance portuaire robuste et plus performant que les systèmes existants en choisissant le GPMM comme site d'expérimentation.

Au cours d'une phase d'étude préliminaire, nous avons décidé avec l'ensemble des partenaires industriels du projet SECMAR que ce système robuste de surveillance devrait être basé sur des données de qualité, issues de nouveaux systèmes de détection. De ce constat ont découlé les deux phases principales de ce projet :

1. une première phase a consisté à instrumentaliser le port par un système de nouveaux capteurs capables d'enregistrer des données précises, relatives aux différents bateaux évoluant dans le port dans des conditions différentes. Cette phase a été assurée par nos partenaires industriels (TUS, Bertin Technologies, C&S, TRS).
2. Une deuxième phase a consisté à concevoir un module automatique de reconnaissance de comportements utilisant les données de haute qualité issues de ces capteurs afin de nous donner une information fiable en temps-réel sur le type de comportement que le bateau est en train d'accomplir. Ce module constitue, à travers ce travail de thèse, la contribution du Centre des Mathématiques Appliquées (CMA) dans le projet SECMAR.

Dans ce qui suit, nous allons introduire les systèmes de capteurs conçus par les partenaires du projet SECMAR. Notons que le processus détaillé de la conception de ces capteurs ne fera pas l'objet d'une étude approfondie dans ce manuscrit.

Rappelons les différents besoins de détection actuels présentés en début de section et qui ont motivé le développement de ces capteurs :

- détection globale de tous les bateaux, y compris ceux qui ne présentent pas de systèmes AIS tels que les petits bateaux ;
- détection jour et nuit des bateaux évoluant dans le port par un système de vidéo surveillance ;
- détection sous-marine des objets ;
- détection d'une cible masquée par un objet.

Pour répondre à ces besoins trois nouveaux types de capteurs ont été utilisés : un Radar, des caméras optroniques et un Sonar.

En général, chacun de ces capteurs traite les données sous la forme de deux processus consécutifs :

- un processus de détection.
- un processus de pistage.

Au cours du processus de détection, chaque capteur va enregistrer l'ensemble des données et d'informations perçues dans le port. Beaucoup de données enregistrées seront inutilisables. Pour cette raison, le développement d'algorithmes d'extraction des informations pertinentes s'est donc avéré indispensable. Les partenaires du projet se sont investis afin de rendre les données issues de ces capteurs d'un très haut niveau de précision et de qualité.

Le processus de pistage consiste alors à regrouper les données en rapport avec le même objet au cours du même pas temporel et ceci suivant un critère déterminé.

La section suivante est une description de chaque technologie et évoque pour chacune les deux processus de détection et de pistage.

1.5.1 Instruments utilisés pour la récolte des données

Le Radar : Le Radar¹⁹ a été fourni par Thales Radar System (TRS) qui a utilisé une technologie FURUNO (voir figure (1.7)[92]) sur laquelle plusieurs réglages ont été effectués afin de détecter des cibles de taille minimale de 1 m².

Tout système Radar est constitué d'un émetteur et d'un récepteur d'ondes radio quasi-planes. L'émetteur envoie une onde qui va se réfléchir partiellement dès la rencontre d'un objet métallique. Ce dernier renvoie une énergie qui sera détectée, à son tour, par un récepteur situé au même endroit que l'émetteur. Ceci permet de localiser la cible tout en enregistrant certaines informations la concernant. En effet, la position de la cible est estimée grâce au temps de retour du signal ainsi que

19. RAdio Detection And Ranging.

la fréquence de l'onde envoyée. La vitesse est mesurée à partir du changement de fréquence par effet Doppler²⁰.

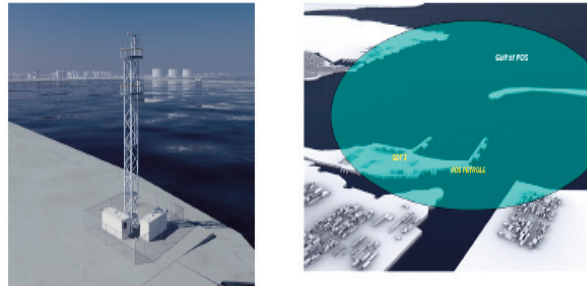


FIGURE 1.7 – Le radar terrestre et sa couverture circulaire

De plus, si on approxime l'onde incidente à une onde plane, ce qui reste cohérent avec la précision recherchée, on connaît la direction dans laquelle se trouve l'objet (l'azimut) et on trouve donc sa position ainsi qu'une estimation de sa taille.

Dans le cadre du projet SECMAR, le Radar tourne à 22 tours/minute (voir figure (1.7)) et récupère les informations au fur et à mesure de sa rotation. Il a une fréquence de 100 MHz et une antenne de 3,2 m.

Au cours de la première phase de détection et pour une grande sensibilité de l'appareil, le Radar permet de récupérer un maximum de données à savoir des données inutiles telles que les nuages, les vagues, la neige, etc.

De plus, si par exemple, un gros objet arrive devant une petite cible, le signal réfléchi par la petite cible sera complètement brouillé par celui de la grande.

Pour ces raisons, un ensemble d'algorithmes dits d'intégration a été développé sur ce système afin d'extraire les données inutiles (telles que vague, pluie, neige) et ne garder que les données qui nous intéressent (principalement les navires).

A la fin de ce processus de détection, on se retrouve avec un ensemble d'informations et ceci à chaque pas temporel. Ces informations concernent l'azimut, le doppler, l'énergie, la vitesse et la surface apparente de la cible. Le processus de pistage permet alors de regrouper les données en relation avec le même objet et au même instant selon le critère d'appartenance à une zone géographique. On parle alors de « pistes Radar ».

20. L'effet Doppler est le décalage de fréquence d'une onde acoustique ou électromagnétique entre la mesure à l'émission et la mesure à la réception lorsque la distance entre l'émetteur et le récepteur varie au cours du temps.

Les capteurs optroniques : Les capteurs optroniques (voir figure (1.8)[92]) conçus par Bertin Technologies et Communication et System (C&S) fournissent six caméras (configuration 3 têtes), qui fonctionnent jour et nuit. Trois caméras fonctionnent en infrarouge (pour la nuit) et trois en couleur. Elles sont dotées d'un système de relance qui leur permet de fonctionner 90 % du temps et de résister à un vent de 100 km/h. Elles offrent un champ de vision de 60°. Ces caméras servent à détecter les objets mobiles de jour comme de nuit ainsi qu'à les suivre si nécessaire. Elles disposent de calculateurs internes qui fournissent des informations concernant la localisation, la vitesse et le cap de la cible, à raison de 4 fois par seconde.

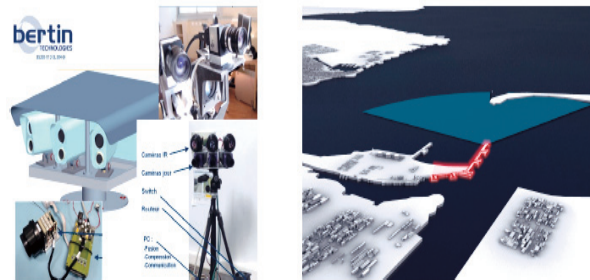


FIGURE 1.8 – Les caméras optroniques et leurs couvertures

Contrairement au Radar, les caméras sont fixes (donc leur couverture n'est pas circulaire) et ne donnent pas d'indications sur la distance à laquelle se trouve l'objet. Elles ont aussi été dotées d'un ensemble d'algorithmes de détection d'objets afin d'extraire de l'image uniquement l'information qui nous intéresse (par exemple enlever les vagues de l'image, etc).

De la même manière que pour le Radar, un processus de pistage des données a été effectué selon plusieurs critères choisis, tels que le regroupement des objets ayant la même intensité lumineuse.

Le Sonar : Le Sonar²¹ de TUS s'est avéré indispensable dans cette reconnaissance de comportement car c'est le seul capteur qui effectue une détection sous-marine. Contrairement au Radar, le Sonar émet une onde mais ne reçoit rien en retour. Il se contente d'écouter les ondes sonores provenant des bateaux. Le Sonar utilise les propriétés particulières de la propagation d'une onde dans l'eau pour positionner et détecter un objet sous l'eau.

L'onde émise par un Sonar va se propager de trois manières :

21. SOund Navigation And Ranging.

- sur la surface ;
- par réfraction à la surface de l'eau ;
- par réflexion sur le fond.

Grâce au principe de réfraction, il est possible de détecter un objet quand il est masqué par un autre.

Le processus de pistage du Sonar est basé sur la signature acoustique. En effet, les données qui sont dans la même direction et émettent un son à la même fréquence sont relatives au même objet.

Malheureusement la nature du fond (sable, rochers...) et l'agitation de la surface vont affaiblir le signal à détecter. De même, le bruit ambiant (trafic maritime, bruit des vagues, bruits biologiques de toutes sortes) gêne la bonne réception du signal. Cependant, les traitements d'antenne (permettant de n'écouter que dans certaines directions) et les algorithmes de traitement du signal (filtrage fréquentiel et spatial de tous les bruits parasites) vont aider le Sonar à s'en affranchir. Le Sonar détectera la cible dès que le niveau de l'écho ou du bruit spécifique de la cible sera suffisamment fort pour dépasser le seuil de déclenchement (prenant en compte les bruits de fond).

Le Sonar est placé à 100 mètres de la côte, à environ 18 mètres de profondeur. Il enregistre les mêmes types de données que le Radar à savoir l'azimut, la distance, l'énergie et la vitesse. Sa zone de couverture est plus large que celle des caméras optroniques.

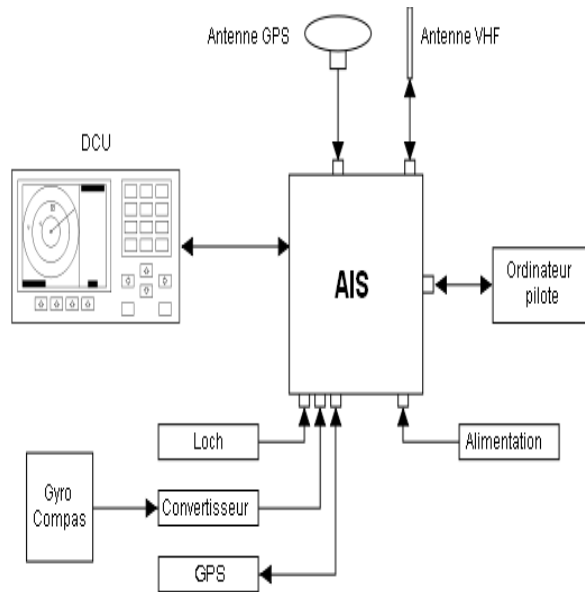


FIGURE 1.9 – Schéma général du système AIS

Le système AIS : Un système AIS (préalablement introduit dans la section 1.3) est un système international d'échanges automatisés de messages entre navires par radio VHF. Il permet aux navires et aux systèmes de surveillance du trafic de connaître l'identité, le statut, la position et la route des navires se situant dans la zone de navigation. Il est composé (voir figure (1.9)) d'un émetteur, de 2 récepteurs TDMA²² VHF, d'un calculateur, d'un système ASN²³, d'un système de positionnement par satellite, d'un DCU²⁴ et est interfacé avec les instruments du navire (compas gyroscopique ou satellitaire, indicateur de vitesse ou de changement de cap). Le signal est multiplexé pour éviter que les signaux des navires ne se brouillent mutuellement en étant émis au même moment.

Un émetteur AIS diffuse, toutes les 6 minutes, des données permettant l'identification des navires telles que le nom du navire, la destination, l'Heure Estimée d'Arrivée (HEA), ainsi que des informations qui concernent les positions en longitude et en latitude du bateau.

22. Time Division Multiple Access (TDMA) est une méthode d'accès au canal pour les réseaux à média (généralement sans fil) partagé.

23. ASN(Appel Sélectif Numérique) est un mode de communication utilisant une technique de transmission automatique des appels codés en format numérique.

24. Display Control Unit : écran de contrôle.

1.5.2 Disposition des capteurs dans le GPMM

Les données dont nous disposons proviennent des trois types d'instruments (Radar, Sonar et caméras optroniques) conçus par les partenaires industriels de SEC-MAR. Ces données sont stockées et analysées dans une infrastructure logistique centralisée dont les résultats sont affichés à l'IHM (Interface Homme-Machine) de la Vigie de Port-de-Bouc où les décisions d'intervenir sur un objet suspect sont prises.

La disposition des équipements dans le port est représentée à la figure (1.10) [92] :

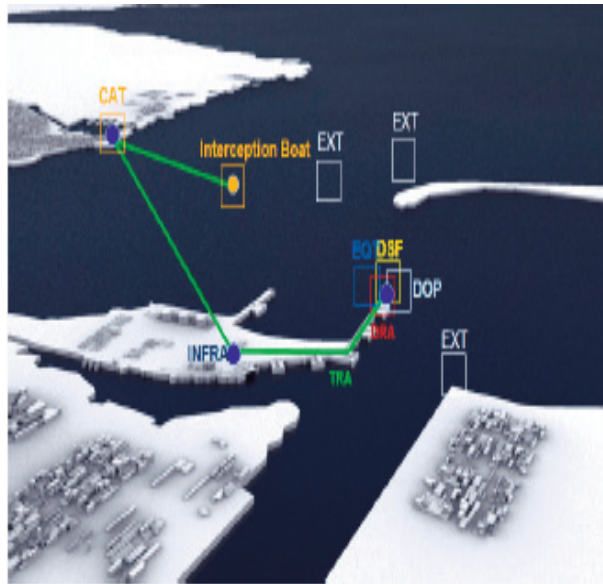


FIGURE 1.10 – Disposition des capteurs

Où :

- CAT est le Centre de Traitement des Données (Centre À Terre) ;
- INFRA désigne les infrastructures logistiques ;
- DRA est le Radar ;
- DSF est le Donar ;
- DOP est le système de Détection OPtronique.

Un processus de fusion des données issues de tous les capteurs du port, à savoir les données AIS a été effectué par l'équipe de TUS afin de simplifier considérablement ces données et d'obtenir un seul point par bateau sur la carte en fonction du temps et quel que soit le nombre d'instruments de détection l'ayant repéré.

1.5.3 Le processus de fusion des données

Pour récapituler, chaque capteur (Sonar, Radar, caméras optroniques et AIS) va extraire des mesures physiques des événements de détection. Ces événements sont généralement obtenus par filtrage du bruit (intégration) et par détection de pics de mesure. Ces événements, évalués périodiquement, vont être reliés entre eux afin de former des pistes systèmes. Le lien en événement se fait sur un ensemble de grandeurs mesurées (spectre, cinématique, énergie, doppler) propre à chaque technologie.

Les pistes capteurs ainsi obtenues ne sont pas forcément synchrones et sont à des niveaux de précisions très différents pour le même objet et à chaque pas temporel. D'où le rôle du processus de fusion qui a été effectué par TUS. Notons que des algorithmes ont été mis en place afin de fusionner les pistes systèmes du Radar, des caméras optroniques, du Sonar ainsi que du système AIS. Ce processus de fusion illustré par la figure (1.11) ne sera pas développé dans ce manuscrit [92].

Les pistes systèmes résultantes, qui sont de très haut niveau de précision et de qualité, sont des suites temporelles d'informations en position, vitesse instantanée, temps et cap pour l'ensemble des navires du port.

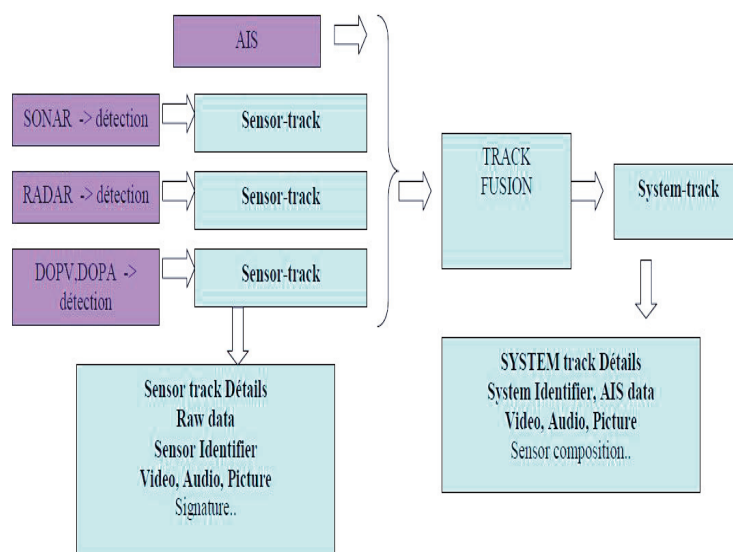


FIGURE 1.11 – principe de fusion dans le système SECMAR

Ainsi, à partir d'une ensemble d'algorithmes de fusion choisi par l'équipe de TUS, il a été possible d'obtenir un ensemble de pistes systèmes (figure (1.11))[92], c'est à

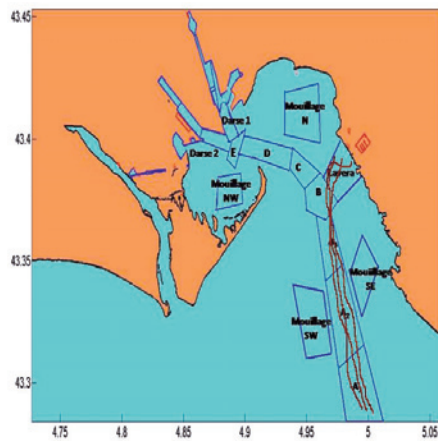
dire une suite temporelle d'informations (position, cap, vitesse, instant, année, etc) pour chaque bateau. Nous avons, par la suite, élaboré un module qui exploite ces pistes systèmes dans le but d'apporter une aide aux personnels du port afin de mieux surveiller les comportements de navires au cours de leurs déplacements dans le port. Nous décrirons alors dans la section suivante comment nous avons choisi de traiter notre problématique ainsi que les solutions que nous proposons pour la construction de ce module.

1.5.4 Notre contribution

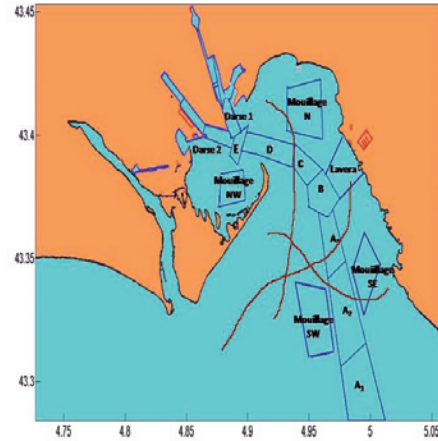
Pour traiter notre problématique, nous avons tout d'abord commencé par observer à l'interface de visualisation les trajectoires des navires dans le GPMM obtenues à partir des pistes systèmes afin d'analyser leurs évolutions dans le port. Comme nous l'avons déjà indiqué en introduction, nous avons identifié deux types de trajectoires de bateaux :

1. Les trajectoires des bateaux commerciaux et à cargaison dangereuse empruntent des chenaux d'accès et des zones préalablement définies par l'arrêté préfectoral (voir figure (1.12a)). Cet arrêté²⁵ définit impose des chenaux d'accès, des zones de mouillages ainsi que d'autre zones spécifiques que chaque catégorie de bateaux doit prendre en circulant dans le port. A titre d'exemple, les pétroliers doivent suivre les zones du chenal d'accès allant jusqu'au quai pétrole.
2. Les petites embarcations telles que les jet-ski, les voiliers, les bateaux de pêcheur et les bateaux de plaisance circulent librement dans le port et leurs trajectoires sont très aléatoires (voir figure(1.12b)). En effet, les comportements de ces petits navires ont toujours été considérés comme peu dangereux. Pour cette raison, toutes les règles maritimes qui ont été mises en place concernent principalement les gros bateaux et n'ont jusqu'à présent pas été généralisées aux petites embarcations, ce qui rend leur surveillance très difficile.

25. Arrêté interpréfectoral de l'année 2010. Portant réglementation de la navigation dans la zone maritime et fluviale de régulation du Grand Port Maritime de Marseille.



(a) Exemples de trajectoires de gros bateaux ayant un comportement régulier dans le GPMM.



(b) Exemples de trajectoires de petits bateaux qui se déplacent librement dans le GPMM.

FIGURE 1.12 – Deux scénarios de déplacement de bateaux dans le GPMM.

Les gros bateaux qui doivent suivre des règles de navigation établies par la législation se caractérisent par des comportements réguliers et récurrents. Par conséquent, les pistes systèmes que nous avons eues à notre disposition représentent une base de données d'informations précises (positions, vitesses, caps, temps) et régulières enregistrées par les nouveaux capteurs au cours des déplacements de tous ces navires dans le port.

Nous avons alors choisi d'utiliser ces données stockées associées aux gros bateaux ainsi que des informations sur les zones fonctionnelles afin de construire et d'apprendre, de façon probabiliste, tous les modèles de comportements de bateaux dans ce port (par exemple, le modèle probabiliste m_i construit est associé au comportement de bateaux se dirigeant vers la zone i , etc). L'ensemble des modèles construits a été utilisé par la suite pour reconnaître en temps-réel, par des calculs probabilistes, le comportement de tout nouveau bateau entrant dans le port et ceci en distinguant entre un comportement connu et un comportement non reconnu donc signalé comme suspect.

Néanmoins, dans le cas des petites embarcations ayant des comportements non-réguliers, la phase d'apprentissage n'était pas possible. Il était alors indispensable de trouver une solution pour traiter le cas où les navires se déplacent librement dans le port (voir figure (1.12b)). Nous avons ainsi opté pour une interprétation en temps-réel des informations qui étaient à notre disposition (pistes systèmes, zones géographique

et réglementations) afin d'en extraire des *événements*. A titre d'exemple, l'augmentation de la vitesse du navire au cours de son mouvement dans le port est traduite par l'événement « *le bateau accélère* », l'entrée d'un bateau dans une zone n est traduit par l'événement « *le bateau est dans la zone n* », ou encore le type du navire (un navire commercial par exemple) obtenu par un module extérieur développé par nos partenaires est traduit par l'événement « *le bateau est commercial* ». Toute information physique, telles que les pistes systèmes et les zones géographiques, est ainsi transformée en une information logique sous la forme d'un événement. Chaque événement, que nous avons appelé *constat*, est calculé à chaque pas temporel (c'est-à-dire pour chaque piste système enregistrée).

Cette suite d'événements reçus au cours du temps a permis de décrire le comportement en cours. Par exemple, si au cours du temps on dispose de la suite des événements suivants : « *un bateau commercial* », « *le bateau entre dans une zone interdite* », « *le bateau accélère* » et « *le bateau change de cap brutalement* », le comportement en cours, considéré comme dangereux, peut-être construit. Il s'agit de manipuler cet ensemble d'événements booléens par un ensemble d'instructions simples afin de spécifier ce comportement dangereux. Dans le cadre de cette approche temps-réel, nous avons choisi d'appeler un comportement de bateau *scénario*. Nous construisons, de cette manière, plusieurs types de scénarios qui envoient instantanément des alarmes qui signalent la nature du comportement du bateau.

Pour modéliser et reconnaître, dans un même module (c'est-à-dire le BAM), ces deux catégories de comportements différents de navires, nous avons conçu deux sous-modules basés sur deux approches différentes. Il s'agit du :

- BPAM, basée sur l'approche probabiliste Modèles de Markov Cachés (MMC), est dédiée à l'analyse et la reconnaissance des comportements réglementés des gros bateaux.
- BRAM, basée sur une approche réactive synchrone traduite par le langage ESTEREL, est dédiée à la reconnaissance des comportements non-réglementés des petites embarcations.

Nous expliquerons, dans le chapitre 4, le choix précis des MMC et de l'approche réactive synchrone et nous détaillerons l'utilisation de ces approches pour la construction du BRAM et du BPAM.

Conclusion

Dans ce chapitre, nous avons mis en évidence l'importance d'une zone portuaire sur les plans économique, écologique et social ainsi que les risques auxquels elle fait face. Nous avons également introduit les différents dispositifs de sécurité qui

existent actuellement, à savoir les normes et les systèmes de détection. Puis, nous avons souligné certaines faiblesses des dispositifs actuels en prenant le GPMM, qui est notre site d'expérimentation, comme exemple. Dans le but d'apporter une aide à la sécurisation de ce port, le projet SECMAR a été mis en place.

Dans le cadre de ce projet et afin d'améliorer les systèmes de détection actuels, un ensemble de nouveaux capteurs regroupant un Radar, un Sonar et des caméras optroniques a été développé par les partenaires industriels de ce projet. Il a permis de mettre à notre disposition un ensemble de données de très bonne qualité sur les bateaux.

Dans le cadre de nos travaux de recherche, nous avons exploité ces données afin d'élaborer un module de reconnaissance de comportements de navires dans le port. Puisque notre objectif était de trouver un moyen pour modéliser et reconnaître « les comportements de bateaux » dans le port, nous allons nous intéresser dans le chapitre suivant à la notion de « reconnaissance de comportement » en regardant les différents domaines qui ont eu recours à cette technique pour répondre à des besoins particuliers. Pour étudier cette technique, nous allons essayer de répondre aux trois questions suivantes :

- Comment peut-on définir la technique de *reconnaissance de comportements* ?
- Sur quelles *types de données* cette technique est-elle basée ?
- Quelles sont alors ses différents *domaines d'application* ainsi que *les différentes méthodes utilisées* ?

Chapitre 2

la technique de reconnaissance de comportements

Depuis plusieurs décennies, les recherches dans le domaine de la modélisation et la reconnaissance de comportements n'ont cessé de prospérer afin de répondre à des besoins particuliers. La reconnaissance de comportements constitue une branche de l'Intelligence Artificielle (IA), permettant à travers des techniques solides, de rendre des véhicules ou des robots autonomes ou semi-autonomes [49, 72, 73, 85, 86, 96] afin de développer des systèmes de sécurité ou d'aide à la décision. La conception de ces systèmes est souvent basée sur le modèle du comportement humain.

La reconnaissance de comportements a aussi été utilisée en vision par ordinateur pour analyser et interpréter des images ou des séquences vidéos dans le cadre de la vidéosurveillance. Nous présenterons dans ce chapitre un aperçu de la technique de reconnaissance de comportements en présentant certains de ses domaines d'application ainsi que les différentes méthodes utilisées.

2.1 Définition

De manière générale, on peut définir la technique de *reconnaissance de comportements* comme l'utilisation de différents types de modèles préalablement construits grâce à des informations pertinentes, des langages, des algorithmes et un ensemble de règles afin de classifier et reconnaître en temps-réels des comportements à priori inconnus. Cette technique est donc basée sur deux phases :

1. Une phase de construction du modèle.
2. Une phase de reconnaissance.

Dans certaines applications, la construction du modèle passe par son *apprentissage*. L'*apprentissage* peut-être défini comme étant l'utilisation d'un ensemble d'informations issues de plusieurs systèmes de détection (à savoir des systèmes de capteurs) qui enregistrent des données différentes (position d'un objet, d'une image, etc) afin de construire, grâce à des algorithmes, le comportement d'un phénomène en question. En pratique, cela pourrait se traduire par la construction de plusieurs modèles de comportements effectués par des personnes différentes à partir d'images d'une caméra (par exemple, on modélise à partir d'images issues d'une caméra, le comportement d'une personne qui ouvre une porte ou qui lève la main ou encore qui se met à courir, etc) ou encore la modélisation du comportement d'un conducteur d'une voiture à partir de données capteurs sur la position et la vitesse de sa voiture (par exemple, on modélise le comportement d'un conducteur qui veut tourner à gauche ou qui change de voie sur la route, etc).

La phase de *reconnaissance* utilise les modèles préalablement construits (au cours d'une phase d'apprentissage par exemple) et des nouvelles données temps-réel issues des mêmes catégories de capteurs afin de distinguer et « reconnaître » le comportement en cours. Dans l'exemple de l'activité de conduite, après avoir construit un ensemble de modèles associés à plusieurs comportements possibles d'un conducteur, il est possible de reconnaître et de déduire en temps-réel, le type de comportements qu'il est en train d'adopter (par exemple, le conducteur est en train de tourner à gauche, etc).

La problématique de reconnaissance de comportement a été traitée depuis plusieurs décennies afin de répondre à plusieurs types de besoins spécifiques (assistance aux personnes, développement de systèmes de sécurité, etc). Ainsi dans les années 80, les travaux de Kautz [70] se sont concentrés sur la définition de la reconnaissance de l'intention « plan recognition ». En partant d'un ensemble d'actions accomplies par un ou plusieurs agents, il était question de trouver un raisonnement logique pour inférer le scénario ou l'intention voulue par ses actions. Puis en 1997, Oliver, a utilisé la théorie des Coupled Hidden Markov Models (CHMM) dans l'article [36] pour reconnaître de simples activités humaines (par exemple, lever la main, sauter, marcher, etc). Bui a introduit en 2003, la théorie de Abstract Hidden Markov Models (AHMM) [38] afin de modéliser des comportements humains plus complexes (par exemple, la modélisation du comportement « se diriger vers un campus » est composé des sous-comportements suivants « aller dans une station de bus », « prendre le bus au campus », « marcher jusqu'à une office », etc).

Cette problématique de reconnaissance de comportement s'intègre dans le cadre du développement de systèmes robustes en Intelligence Artificielle (IA), ce qui consiste à rendre un système informatique capable de se comporter pertinemment comme un

être humain. Cette science a été définie par l'un de ses créateurs Marvin Lee Minsky comme « *la construction de programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique* ». L'usage des ordinateurs permet d'atteindre le côté « *artificiel* » et le côté « *intelligence* » est acquis en imitant le comportement d'un être humain. Cette imitation peut se faire dans le raisonnement comme dans la pratique des mathématiques, dans la perception telle que l'aspect visuel (interprétation des images et des scènes), auditif (compréhension du langage parlé) ou par d'autres capteurs.

Par ailleurs, il existe plusieurs autres domaines d'applications basées sur la technique de reconnaissance à savoir les systèmes de sécurité pour la vidéosurveillance et les systèmes médicaux d'assistance aux personnes atteintes de troubles cognitifs notamment la démence ou de la maladie d'Alzheimer.

Dans tous les cas, trois facteurs ont contribué au développement de ces systèmes [83] :

1. Le développement de plusieurs technologies de perception (détection) à savoir le *Global Positionning System* (GPS), la Radio Identification (RFID ¹ en anglais), les caméras numériques, les capteurs ultrasons, les capteurs infra-rouges, les capteurs de lumière et les capteurs de mouvements, ce qui a permis aux systèmes informatiques de mieux percevoir le monde réel en utilisant les données issues de ces capteurs pour modéliser un certain type de comportement par exemple.
2. Les avancées dans le raisonnement probabiliste (par exemple, Réseau Bayésien (RB), Modèle de Markov Caché (MMC), etc). En effet, le comportement humain inclut plusieurs facteurs incertains et par conséquent il est difficile de le modéliser de façon déterministe.
3. L'utilisation pour la modélisation des comportements d'un nombre de connaissances de base disponible sur internet.

Nous pouvons conclure que la technique de reconnaissance de comportements est la base de tout système intelligent. Avant de reconnaître n'importe quel type de comportement, une phase préalable de construction du modèle de comportements est indispensable. Cette phase de modélisation peut-être effectuée par un apprentissage automatique du modèle grâce aux données issues des systèmes de détection (à savoir

1. La RFID est une méthode de mémorisation et de récupération de données à distance en utilisant des marqueurs appelés "radio-étiquettes".

les capteurs). Le développement de plusieurs types de données ainsi que le développement de plusieurs approches probabilistes (par exemple RB, MMC, etc) ont permis de mieux analyser et reconnaître les différents types de comportements (comportement d'un conducteur, comportement d'un passager dans un aéroport, etc) pour répondre à plusieurs besoins (sécurité, assistance aux personnes, etc) dans différents domaines d'applications (Intelligence Artificielle, vision par ordinateur, etc).

2.2 Domaine d'application et méthodes utilisées

Nous présenterons dans cette section certains des domaines d'application que nous avons rencontrés dans la littérature basés sur la modélisation et la reconnaissance de comportement et nous citerons, pour la plupart, de ces applications les méthodes utilisées.

La robotique En robotique, la technique de reconnaissance de comportements a été adoptée afin de développer des robots autonomes. Selon, la thèse de Julien Diard [49] sur la conception d'un robot autonome par une carte bayésienne, la robotique autonome a été définie comme étant « *l'ambition de réaliser des systèmes sensori-moteurs complexes capable de travailler dans des environnements naturels* » de la même manière qu'un être humain. On appellera ces robots « des robots humanoïdes ».

Des industriels et des laboratoires se sont alors lancés sur des projets à grande échelle. A titre d'exemple, Honda a conçu Asimo, un robot capable de courir à 3km/h, de communiquer par la parole avec un interlocuteur et doté d'une interaction très élaborée avec l'environnement humain. Fujitsu a également créé un robot capable de faire l'équilibre sur la tête, alors que Kawada a créé le seul robot humanoïde capable de chuter au sol et de se relever ensuite, de façon complètement autonome.

D'après les travaux de Pierre Bessière [31], la construction d'un robot autonome passe par une modélisation de l'interaction d'un robot avec son environnement qui peut être fermé (indoor) ou ouvert (outdoor). Cette modélisation peut se faire en utilisant deux approches :

- Les travaux se basant sur les approches objectivistes.
- Les travaux se basant sur les approches subjectivistes.

L'approche objectiviste illustrée par la figure (2.1a) consiste à construire le modèle de l'interaction du robot avec son environnement selon la propre conception du programmeur. Le programmeur impose au robot un certain programme basé sur des termes géométriques (la forme des objets et la carte de l'environnement), des formules analytiques (obtenues grâce aux lois physiques) et des symboles (obtenus en

nommant les objets et leurs caractéristiques). D'après les travaux des auteurs dans [15], cette approche permet d'utiliser les méthodes de logiques formelles, certains langages ainsi que la théorie des automates afin de résoudre les problèmes de planification et de contrôle associés à des robots ou à un ensemble de robots autonomes. Dans le cadre d'une approche objectiviste, le modèle est construit par un programme impératif (voir figure (2.1a)). Nous pouvons citer à ce titre, les travaux de Somya [122] qui utilise deux robots programmés en langage impératif (le langage ESTEREL) pour modéliser le comportement des fourmis.

Néanmoins, le choix de cette approche pour rendre un robot autonome présente certaines limites. En effet, les données expérimentales enregistrées par le robot autonome grâce aux capteurs et aux effets sur les actionneurs sont non-compatibles avec le modèle du programmeur qui n'a pas pris en compte certaines variables cachées. Le modèle du programmeur construit par approche objectiviste est dit incomplet et le problème qui se pose dans ce cas est le suivant : Comment le robot peut utiliser un modèle incomplet pour prédire, inférer, décider et agir efficacement ?

Selon certains chercheurs, l'approche subjectiviste est considérée comme une alternative à l'approche objectiviste pour réduire cette incomplétude due à un manque d'information sur le modèle. Cette nouvelle approche consistera à modéliser l'interaction d'un robot avec son environnement en utilisant une représentation probabiliste de l'environnement obtenue grâce à une phase d'apprentissage au cours de laquelle le programmeur va introduire les connaissances initiales et le robot va collecter des données expérimentales dans le but d'affiner cette représentation (voir figure (2.1b)).

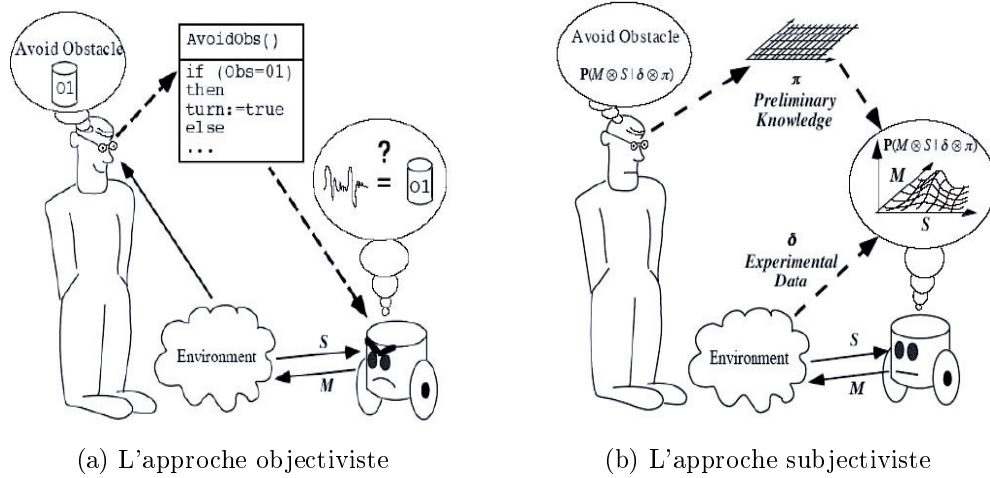


FIGURE 2.1 – Modélisation de l’interaction d’un robot avec son environnement par deux approches différentes [31].

Cette phase d’apprentissage, au cours de laquelle le robot et le programmeur coopèrent ensemble, permet alors de trouver la représentation probabiliste la plus adéquate en réduisant au maximum l’incertitude sur le modèle de l’interaction du robot avec son environnement (notons que l’incertitude est une conséquence de l’incomplétude dans le cadre de cette approche subjectiviste).

Dans le cadre de cette approche subjectiviste, les travaux des auteurs dans [138], dans le domaine spatial, ont permis à un robot de téléopération (opération d’une machine à distance similaire au principe de la télécommande) d’acquérir des compétences lui permettant de remplacer efficacement une Unité Remplaçable sur Orbit (URO²). Ces compétences consistent à « avancer », « tenir fermement et dévisser l’URO » puis « reculer ». Pour concevoir un tel robot, l’approche probabiliste MMC a été adoptée permettant ainsi la modélisation de la compétence humaine puis son transfert au robot. En considérant la performance humaine comme naturellement stochastique, elle a ainsi été modélisée par un processus stochastique représenté par l’ensemble des actions humaines telles que le mouvement et la force (représentant les observations du modèle) sous-jacentes à l’ensemble des intentions humaines (représentent les états du modèle). Les paramètres des MMC associés aux compétences sont, par la suite, mis à jour au cours d’une phase d’apprentissage afin d’améliorer la modélisation de ces compétences. Par la suite, le robot utilisera ces modèles construits

2. l’URO est un module indépendant et interchangeable pouvant être remplacé durant les sorties spatiales ou à distance.

pour sélectionner la bonne commande qu'il est censé effectuer. De cette manière, le robot est considéré comme ayant acquis une certaine « compétence humaine ».

De la même façon, dans [67], les auteurs ont développé un ensemble de robots autonomes jouant au football. Chaque robot est censé inférer les intentions des autres robots à partir de leurs actions. Le comportement de chaque robot a été modélisé par un MMC. Par exemple, pour « un robot qui essaie d'aller derrière le ballon », le comportement est fait en 4 étapes : « commencer à exécuter le comportement en question », « tourner autour du ballon », « se mettre en face du ballon » et « se rapprocher du ballon ». Ces étapes constituent les états (les intentions) des modèles de MMC associés aux comportements et les observations qui ont été prises en compte dans la modélisation sont celles des positions des autres robots et des ballons.

Les activités de conduite A partir des années 80, les travaux relatifs aux systèmes d'assistance à la conduite basés sur la reconnaissance de comportement ont pris leur essor, notamment dans le cadre du programme européen PROMETHEUS, lancé en 1986, qui visait l'amélioration de la sécurité routière européenne. Pendant huit ans, ce programme a ainsi rassemblé des constructeurs automobiles et des laboratoires de recherche autour du concept de « véhicule intelligent », composé de systèmes visant à assister le conducteur sur différentes dimensions de la conduite présentant un enjeu du point de vue de la sécurité, telles que la gestion des vitesses, des distances, de la trajectoire du véhicule, d'un itinéraire, d'une flotte de véhicules, de la communication entre véhicules et entre véhicules et infrastructure, etc.

Les chercheurs du domaine ont alors considéré que cette assistance n'était possible qu'en essayant de modéliser le comportement du conducteur. Cette modélisation est basée sur la notion d'« intention » et la notion d'« action » du conducteur. Nous prendrons dans la suite, des exemples des travaux de recherche dans le cadre de cette modélisation d'intention/action.

Dans la perspective d'intégrer dans un véhicule un système intelligent d'assistance au conducteur, Kuge a utilisé dans l'article [73] les Modèles de Markov Cachés (MMC) pour apprendre et reconnaître son comportement. L'idée générale était donc de construire par apprentissage le modèle d'intention/action du comportement du conducteur. Puisque le comportement de l'être humain est basé sur des intentions qui sont inconnues, on essaiera à partir de ses actions d'inférer ses intentions. En partant de cette idée, le modèle du comportement humain a été rapproché d'un modèle de MMC où les variables du modèle sont construites à partir d'états cachés (intentions) et d'observations (actions). Dans cet article, les auteurs ont choisi de modéliser trois types possibles de comportements : changement de voie en urgence, changement ordinaire de voie et conservation de voie. Les modèles de MMC associés

à ces trois types de comportements ont été construits grâce à des données récoltées par un simulateur de conduite. Ces données (les angles de direction, les vitesses angulaires et les forces données pour tourner) ont été obtenues à partir des actions du conducteur. Les résultats de modélisation et de reconnaissance du comportement de changement de voie étaient très prometteurs d'après cet article.

Dans le même contexte, Liu [85] considère le comportement du conducteur comme étant un Modèle de Markov formée à partir d'une séquence d'états modélisant chacun l'état mental interne du conducteur. Chaque état est considéré comme un modèle dynamique non observable qui doit-être estimé par un filtre de Kalman [6] à partir d'une séquence d'observations correspondant aux actions du conducteur.

Par exemple, le comportement du conducteur « changer de voie » est considéré comme la succession des états suivants « centrage sur la voie », « adaptation de sa propre vitesse avec une voiture parallèle » puis « centrage sur la nouvelle ligne ».

Les auteurs ont alors défini un Modèle Dynamique de Markov Cachés (MDMC) afin de modéliser le comportement du conducteur. L'action que le conducteur a l'intention de reproduire doit alors être inférée en comparant des échantillons d'observations du comportement du conducteur (vitesse, accélération) avec les MDMC construits.

Dans [119] et [90], Sekizawa et Xiaoning ont plutôt utilisé les MMC pour modéliser le comportement de chaque conducteur puisque chaque conducteur a sa propre façon de conduire. Cette idée s'est avérée très efficace pour développer un système intelligent de prévention contre les vols de véhicule. Ainsi, le système est capable de reconnaître son propre conducteur et de déclencher une alarme s'il s'agit d'une tentative de vol.

Assistance aux personnes âgées Selon des chiffres publiés par l'Institut National de Statistique et des Etudes Economiques (INSEE), la France métropolitaine compterait en 2050 plus de un habitant sur trois âgé de plus que 60 ans, contre un sur cinq en 2005.

Cette augmentation du nombre de personnes âgées entraîne une hausse du nombre de places dans les hôpitaux et dans les maisons de retraites. Et par conséquent, le nombre de place dans les hôpitaux et dans les maisons de retraite devrait exploser à son tour.

Le maintien des personnes âgées à domicile le plus longtemps possible dans un premier temps (avant de les confier à des maisons de retraite) pourrait alors être une solution face à l'explosion du nombre de places requises dans ces établissements.

Néanmoins, le vieillissement n'est pas sans risque d'une mauvaise alimentation, d'une mauvaise hygiène ou encore d'une perte d'autonomie de ces personnes qui sont

maintenues à leur domicile. Pour cette raison, plusieurs chercheurs travaillent sur le développement d'un habitat sécurisé pour ces personnes âgées.

Xuan Hoa Binh LE a développé, au cours de ses travaux de thèse [76], un Habitat Intelligent pour la Santé (HIS) équipé d'un système de capteurs infrarouges qui enregistrent des données permettant un suivi à distance de l'occupant dans la pièce. Le but était de parvenir à une reconnaissance automatique des activités réalisées quotidiennement par la personne suivie, afin de détecter une éventuelle perte d'autonomie.

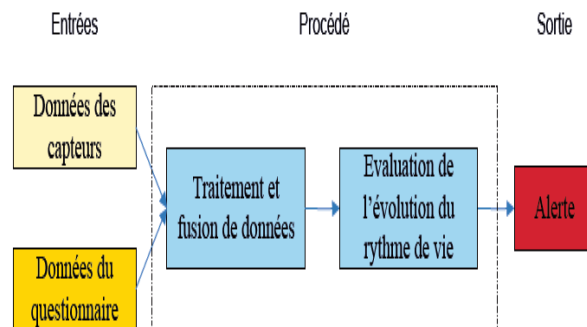


FIGURE 2.2 – Schéma fonctionnel de la méthode de traitement et d'analyse de données pour le développement du HIS [76].

Un questionnaire a initialement été effectué auprès des occupants des habitats afin d'avoir des informations sur leurs horaires et leurs endroits habituels dans la journée au sein du domicile (aller aux toilettes, dormir, sortir, etc). Ceci a aussi été utile afin d'établir des règles de reconnaissance. Par ailleurs, les capteurs infrarouges ont permis de recueillir un ensemble de données sur ces mêmes informations. Les données capteurs sont comparées aux données initiales établies au cours du questionnaire puis le système envoie un message ou une alerte sur l'état d'autonomie de l'occupant. Le mode de fonctionnement du HIS est illustré par le schéma de la figure (2.2).

La vision par ordinateur La vision par ordinateur est une branche de l'Intelligence Artificielle qui cherche à apprendre à une machine à interpréter ce qu'elle voit. La problématique de la reconnaissance de comportement a été utilisée en vision par ordinateur à travers la reconnaissance et l'analyse de l'action humaine à partir d'images de séquences vidéos. Ceci s'explique par l'apparition de différents types d'application à savoir la vidéosurveillance, les interactions homme-machine ou les jeux vidéo, ou encore l'analyse des performances des athlètes, etc.

D'après l'article [107], une action peut-être classée en trois catégories : une action primitive, une action définie comme étant un ensemble d'actions primitives et une activité qui contient un nombre d'actions conséquent. Par exemple, « lever la jambe gauche » est une action primitive, « courir » est considéré comme une action et « faire un saut » est une activité qui est formée des actions « commencer », « courir » et « sauter ».

Un système de reconnaissance d'action effectue une classification d'actions à partir d'images vidéos. Les algorithmes de classification des actions sont généralement appris à partir de données d'entraînement issues des caméras vidéos.

Dans l'article [107], Poppe propose un aperçu des différentes études sur la reconnaissance et l'analyse du comportement humain basées sur la vision. Dans le même contexte, Poppe s'est concentré dans l'article [106] sur les travaux de récupération des poses et des mouvements humains à partir d'une séquence d'images. On peut également citer les travaux de Enzweiler dans [54] qui se sont basés sur la localisation d'un piéton ou d'un être humain dans une image.

Conclusion

Nous avons exposé dans ce chapitre plusieurs domaines d'application qui ont eu recours à la technique de reconnaissance de comportement à savoir la robotique, l'assistance aux conducteurs, l'assistance aux personnes âgées et la vision par ordinateur. Par ailleurs, nous avons cité pour la plupart de ces applications les différentes méthodes utilisées. Nous avons alors remarqué que le raisonnement probabiliste a souvent été adopté par plusieurs chercheurs dans différents domaines d'application pour modéliser le comportement humain.

Pour traiter notre problématique qui, à notre connaissance, n'a pas été traité auparavant et qui consiste à reconnaître les comportements de navires dans le GPMM, nous nous sommes principalement inspirés des domaines d'application suivants :

- La robotique qui oppose les approches probabilistes (RB, MMC, etc) aux approches impératives (symboliques, à base de règles, etc) pour la modélisation des comportements de robots autonomes.
- Les activités de conduite qui, à plusieurs reprises, ont utilisé [73, 103, 85, 67, 138, 56, 135] les MMC pour modéliser les comportements des conducteurs afin de construire des systèmes de sécurité.

Nous avons alors choisi de combiner une approche probabiliste et une approche impérative pour construire notre module d'analyse et de reconnaissance de comportements (BAM) de navires dans le port. Nous rappelons alors qu'une approche probabiliste

par MMC a été choisie pour modéliser et reconnaître les comportements des gros bateaux qui ont des comportements réguliers dans le port et qu'une approche impérative événementielle par le langage réactif synchrone ESTEREL a été utilisée pour modéliser et reconnaître les comportements des petites embarcations qui circulent librement dans le port.

Chapitre 3

Approche probabiliste et approche événementielle

L'approche probabiliste par MMC et l'approche événementielle réactive synchrone par le langage ESTEREL constituent respectivement les outils que nous avons adoptés pour construire les deux sous-modules, le BPAM et le BRAM, du système global d'analyse de comportements (BAM). Pour cette raison, nous organiserons ce chapitre en deux grandes parties : la section 3.1 dans laquelle nous étudierons les MMC et la section 3.2 qui sera consacrée à l'approche réactive synchrone.

Nous étudierons, dans la première partie de ce chapitre, la théorie des MMC (section 3.1). Dans la section 3.1.1, nous introduirons tout d'abord certaines approches probabilistes qui sont fortement liées aux MMC. Par la suite, nous nous intéresserons à la théorie ainsi que les algorithmes des MMC que nous avons choisis pour concevoir le BPAM. Nous introduirons alors, dans la section 3.1.2.1, les paramètres des MMC nécessaires pour définir et initialiser les modèles de comportements de navires. Ces paramètres doivent être reestimés, par la suite, en utilisant les pistes systèmes afin d'améliorer la précision sur ces modèles de comportements. Nous présenterons alors, dans la section 3.1.2.4, l'algorithme Baum-Welch qui permet d'estimer les paramètres des MMC. Par ailleurs, les modèles de comportements de navires sont ceux d'un modèle de Bakis (voir section 3.1.2.2). Etant donnée que ce modèle impose l'utilisation de plusieurs séquences d'observations, nous présenterons, dans la section 3.1.2.4, une nouvelle forme des paramètres estimés par Baum-Welch. Au cours de la programmation de notre BPAM, la normalisation de certains paramètres de nos modèles s'est avérée indispensable. Cette phase de normalisation sera également présentée dans 3.1.2.4.

Les processus d'identification, d'initialisation et de reestimation de ces paramètres

constituent la phase d'apprentissage (ou la construction) des modèles de MMC associés aux comportements de navires.

Ces modèles construits seront utilisés par la suite pour reconnaître en temps-réel les comportements des gros bateaux dans le port. Cette reconnaissance sera effectuée en évaluant et en comparant les probabilités des observations sur les navires au vu de ces modèles construits. Pour évaluer ces probabilités, nous avons utilisé l'algorithme forward-backward que nous introduirons dans la section 3.1.2.4. L'ensemble des deux phases d'apprentissage et de reconnaissance constitue le BPAM.

Dans la deuxième partie de ce chapitre (section 3.2), nous nous intéresserons à l'approche réactive synchrone que nous avons principalement choisie pour modéliser et reconnaître les comportements des petites embarcations. Par cette approche, nous avons conçu le BRAM qui est un ensemble de modules programmés en ESTEREL où chaque module est un scénario de comportements de bateaux. Chaque module prend en entrée un ensemble d'informations sur le bateau et renvoie des alarmes qui traduisent des niveaux de reconnaissance du comportement du navire. Puisque ESTEREL est un langage réactif synchrone, nous présenterons, tout d'abord dans la section 3.2.1, l'approche réactive synchrone puis nous nous intéresserons, dans la section 3.2.2, à ce langage impératif. Nous introduirons ainsi ce langage en présentant ces instructions principales, leur sémantique intuitive qui explique le comportement de ces instructions puis nous présenterons leur sémantique comportementale sous forme de sémantique opérationnelle, ce qui permet d'interpréter un programme de façon non ambiguë. Nous terminerons cette partie par un petit module programmé en ESTEREL qui est celui d'un comportement dangereux d'un bateau afin de mettre en pratique l'approche réactive.

L'application de ces approches dans le cadre de notre problématique sera présentée dans le chapitre 4.

3.1 L'approche probabiliste par MMC

3.1.1 Généralités

La théorie des probabilités propose un cadre mathématique pour représenter quantitativement l'incertain. La probabilité peut-être inférée de deux manières : on parle d'*inférence statistique classique* et d'*inférence bayésienne*.

Au cours d'une *inférence statistique*, on part d'un échantillon aléatoire issu d'une population donnée afin d'induire les caractéristiques inconnues sur cette population. Une fois donnée, les caractéristiques de l'échantillon, reflètent avec une certaine marge d'erreur possible celles de la population (par exemple, la recherche des personnes

aux yeux bleus en France, l'estimation des intentions de vote au cours d'une élection présidentielle, etc). Il s'agit alors dans ce cas de tirer des conclusions fiables sur la population à partir des données statistiques d'un échantillon.

L'*inférence bayésienne*, quant à elle, cherche à réviser la probabilité d'un événement à partir d'un ensemble de connaissances. Dans la perspective bayésienne, une probabilité n'est pas interprétée comme le passage à la limite d'une fréquence, mais comme la simple traduction numérique d'un état de connaissance. Les méthodes bayésiennes se distinguent des méthodes statistiques dites standards par l'application systématique de règles formelles de raffinement des probabilités (lois conditionnelles, règles de Bayes, etc).

Plusieurs approches probabilistes qui sont basées sur la notion d'inférence bayésienne et qui permettent de modéliser un système à base de connaissance ont vu le jour. Une telle modélisation par approche probabiliste permet de réduire l'incertitude entre un phénomène réel et le modèle associé.

Dans ce contexte, on peut citer les Réseaux Bayésiens, les filtres de Kalman, les Chaînes de Markov et les Modèles de Markov Cachés :

- Un Réseau Bayésien (RB) [?] est un modèle probabiliste graphique formulé à partir de la théorie des graphes et de la théorie des probabilités. Un modèle de RB est un ensemble de variables aléatoires reliées entre elles par des dépendances conditionnelles. A titre d'exemple, la notion de RB est très utilisée en diagnostic médical [14] pour représenter la relation qui existe entre des maladies et leurs symptômes. De cette manière, on part des symptômes et on utilise le réseau pour calculer les probabilités d'avoir plusieurs types de maladies.
- Un filtre de Kalman [6] est un filtre qui estime les états d'un système dynamique à partir d'une série de mesures incomplètes ou bruitées. Les filtres de Kalman sont très utilisés en traitement du signal afin de réduire les perturbations sur les mesures de positions et de vitesses d'une cible issues d'un capteur comme un Radar par exemple. Ces perturbations étant dues aux erreurs de mesure.
- Un Modèle de Markov Caché (MMC) [110] est une approche probabiliste qui modélise un système par un ensemble de deux processus stochastiques. Un premier processus stochastique va représenter l'évolution temporelle des observations sur le modèle. Il est sous-jacent à un autre processus stochastique qui représente l'évolution d'un ensemble d'états non observables. Un MMC obéit à l'hypothèse de Markov d'ordre 1. Cette approche probabiliste a largement été utilisée en Intelligence Artificielle (voir section 2.2) pour la reconnaissance de comportements. La reconnaissance de la parole [110] et la reconnaissance de formes (reconnaissance de visages [96], reconnaissance de lettres manuscrites

[79] , etc), qui sont aussi des branches de l'IA, ont été traitées par les MMC et ont montré des résultats prometteurs en matière de reconnaissance.

Plusieurs études menées par différents chercheurs sur les approches probabilistes montrent que toutes ces approches dépendent les unes des autres. Julien Diard a établi, dans son rapport de thèse [49], un graphe qui explique les liens existant entre ces différentes approches. Nous avons repris une partie de ce graphe dans la figure (3.1) que nous développerons par la suite. Le but est d'avoir une idée d'ensemble sur les principales approches probabilistes ainsi que les liens qui les associent.

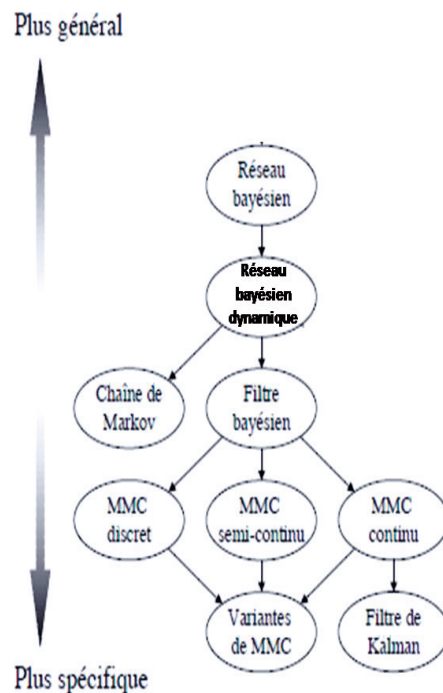


Figure 3.1: Les approches probabilistes du plus général (en haut) au plus spécifique (en bas) [49].

Pour traiter l'hypothèse de généralité, Diard [49] s'est principalement concentré sur la capacité de modélisation de chaque approche sans traiter les problèmes d'inférence et d'apprentissage associés.

D'après Diard [49], la mesure de « généralité » est basée sur le nombre de contraintes imposées par chaque approche : une approche sera plus générale qu'une autre si elle

fait moins d'hypothèses, ce qui lui donne la possibilité de modéliser des phénomènes différents et elle est plus spécifique si elle propose des solutions algorithmiques plus performantes en terme d'inférence et d'apprentissage. Par exemple, les filtres de Kalman dans le cas Markovien Gaussien permettent une inférence probabiliste exacte grâce à une solution analytique. Cette spécificité en terme d'inférence le place en dernier dans le graphe de la figure (3.1). Par ailleurs, les MMC sont très spécifiques grâce à une phase d'apprentissage très performante par l'algorithme Baum-Welch [11].

Pour mieux comprendre ce passage du général au spécifique, nous regarderons le lien entre les formalismes : Réseaux Bayésiens (RB), Réseaux Bayésiens Dynamiques (RBD), Filtres Bayésiens (FB), les Modèles de Markov et les Modèles de Markov Cachés (MMC).

3.1.1.1 Réseau Bayésien

Un RB est défini par un graphe acyclique orienté qu'on notera $G = (S, E)$ où S est l'ensemble des noeuds et E est l'ensemble des arcs de G . Il est également défini sur un espace de probabilité fini (Ω, B, P) (voir annexe A) et porte sur un ensemble de variables aléatoires $(S_1 \dots S_n)$ associées aux noeuds du graphe et définies par (Ω, B, P) . La loi conjointe $P(S_1 \dots S_n)$ (voir annexe A) associée à S s'écrit :

$$P(S_1 \dots S_n) = \prod_{i=1}^n P(S_i / C(S_i)) \quad (3.1)$$

où $C(S_i)$ est l'ensemble des causes (parents) de C dans le graphe.

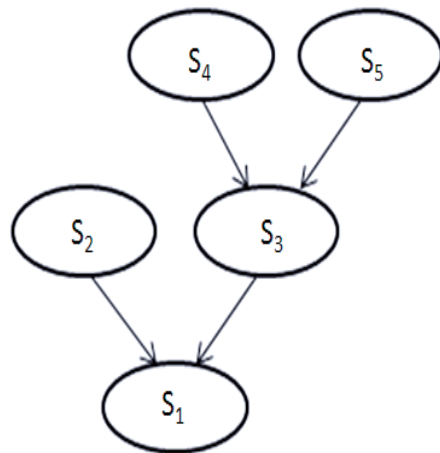


FIGURE 3.2 – Graphe d'un Réseau Bayésien

Par exemple, si on considère le RB de la figure (3.2), la loi conjointe associée est simplifiée grâce à la formule (3.1) en :

$$P(S_1 S_2 S_3 S_4 S_5) = P(S_1/S_2 S_3) P(S_3/S_4 S_5) \quad (3.2)$$

Il s'agit alors ici d'apporter une contrainte sur le type de décomposition de la loi conjointe. Il n'existe aucune autre contrainte sur les types de variables (qui peuvent être discrètes ou continues) ainsi que le type de la distribution de probabilité associée.

3.1.1.2 Réseau Bayésien Dynamique

Un RBD est l'évolution dans le temps d'un RB en respectant à la fois l'hypothèse de stationnarité (voir annexe A) ainsi que l'hypothèse de Markov d'ordre 1 (voir annexe A).

Pour passer du formalisme RB au formalisme RBD, deux hypothèses ont été rajoutées. Pour cette raison, le formalisme RB est considéré comme plus général que le formalisme RBD car il présente moins d'hypothèses.

3.1.1.3 Chaîne de Markov

Une Chaîne de Markov est un cas particulier des RBD où chaque réseau d'une tranche temporelle est réduit à une variable d'état discrète unique qui évolue dans le temps. Cette hypothèse vient s'ajouter à l'hypothèse de Markov d'ordre 1 et à l'hypothèse de stationnarité pour définir un formalisme plus spécifique. Celui des Chaînes de Markov.

3.1.1.4 Filtre Bayésien

Un Filtre Bayésien peut-être vu comme un RBD ayant une seule variable d'état et une variable d'observation qui évoluent dans le temps. Les variables d'état et d'observation peuvent être discrètes ou continues.

3.1.1.5 Modèle de Markov Caché

Un MMC est alors une spécialisation des Filtres Bayésiens qui consiste à faire des hypothèses sur la nature des variables : les variables d'état sont obligatoirement discrètes et les variables d'observation peuvent être discrètes ou continues.

La figure (3.3) décrit les deux processus stochastiques sous-jacents des MMC (chacun est de taille T , $t \in [1, T]$).

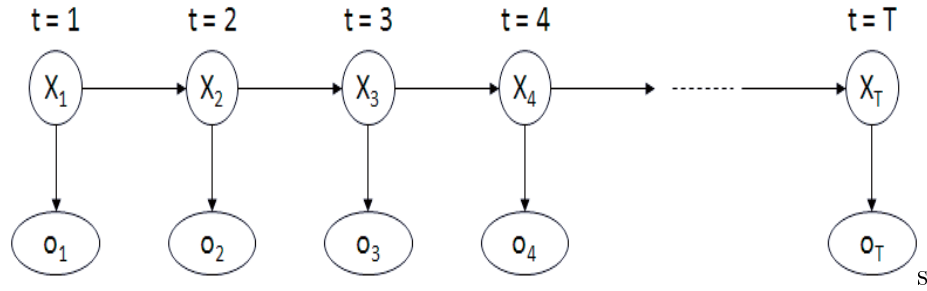


FIGURE 3.3 – Représentation d’un MMC comme une spécialisation des Filtres Bayésiens

Après avoir introduit brièvement certaines approches probabilistes liées entre elles et très utilisées en reconnaissance de comportements, nous allons nous concentrer, dans la section suivante, sur la théorie des MMC que nous avons utilisée pour traiter une partie de la problématique de cette thèse.

3.1.2 La théorie des MMC

Les Modèles de Markov Cachées (MMC) ou Hidden Markov Models (HMM) en anglais est une extension des Modèles de Markov. Ils ont été introduits par Baum et al [10, 9, 12, 11, 8] dans les années 1960-1970. Cette approche probabiliste qui obéit à l’hypothèse de Markov d’ordre 1 est basée sur deux types de variables aléatoires : les états et les observations. Les états, qui sont dits cachés et qui sont discrets dans le cas des MMC, ne sont pas observables. En revanche, il existe un ensemble d’observations qui peuvent être discrètes ou continues et qui s’expriment en fonction de ces états. Il s’agit alors de deux processus aléatoires sous-jacents.

La relation qui existe entre deux états est le terme de transition, celle qui existe entre l’état et l’observation est le terme d’émission.

Soit $(X_1 \dots X_t)$ l’évolution dans le temps d’un processus stochastique à valeurs dans l’espace fini $S = \{S_1, S_2, \dots, S_N\}$, et $(O_1 \dots O_t)$ est un processus stochastique sous-jacent à valeurs dans l’espace fini $V = \{V_1, \dots, V_M\}$. Ces deux suites de variables aléatoires (voir figure (3.4)) sont définies sur un espace de probabilité (Ω, B, P) .

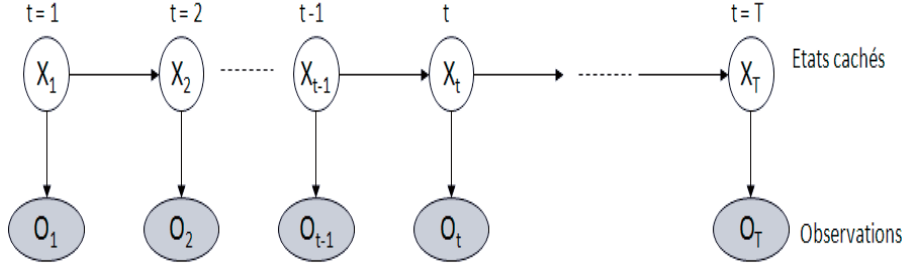


FIGURE 3.4 – Evolution d’un Modèle de Markov Caché dans le temps

Un modèle de MMC est défini sur l’ensemble des paramètres suivants :

- T : la longueur de la séquence d’observation ;
- N : le nombre d’états dans le modèle ;
- M : le nombre de symboles d’observation ;
- $S = \{S_1, \dots, S_N\}$: l’ensemble des états du modèle de MMC ;
- $V = \{V_1, \dots, V_M\}$: l’ensemble des observations possibles du MMC dans le cas discret et $O_t \in R^N$ dans le cas continu.

Ces notations seront conservées pour le reste du manuscrit.

3.1.2.1 Les trois paramètres d’un MMC

Un MMC λ est basé sur trois paramètres principaux : π , A et B qui représentent les connaissances à priori sur notre modèle. Le modèle λ s’écrit alors en fonction de ce triplet comme suit :

$$\lambda = (\pi, A, B) \tag{3.3}$$

- $\pi = \{\pi_i\}_{1 \leq i \leq N}$ est le vecteur des termes $\pi_i = P(X_1 = S_i)$. Ce terme représente la probabilité que la variable aléatoire X_t à l’instant initial $t=1$ prenne une certaine valeur S_i . Ce vecteur doit vérifier :

$$\begin{cases} \pi_i \geq 0, \\ \sum_{i=1}^N \pi_i = 1, \end{cases} \quad 1 \leq i \leq N \tag{3.4}$$

- $A = \{a_{ij}\}_{1 \leq i, j \leq N}$ avec $a_{ij} = P(X_t = S_j / X_{t-1} = S_i)$ est la matrice de transition d'états formée de $N \times N$ probabilités conditionnelles dites termes de transition. Chaque terme nous indique la probabilité de passage d'un état à un autre (c'est-à-dire le passage de S_i à S_j pour $1 \leq i, j \leq N$) entre deux instants consécutifs $t-1$ et t au sein d'un processus stochastique. Pour cette matrice A , les conditions suivantes sont vérifiées :

$$\begin{cases} a_{ij} \geq 0, \\ \sum_{j=1}^N a_{ij} = 1, \end{cases} \quad 1 \leq i, j \leq N \quad (3.5)$$

- $B = \{b_j(O_t)\}_{1 \leq j \leq N}$ avec $b_j(O_t) = P(O_t / X_t = S_j)$
- Dans le cas discret, $B = \{b_j(V_k)\}_{1 \leq j \leq N, 1 \leq k \leq M}$ avec $b_j(V_k) = P(O_t = V_k / X_t = S_j)$. B est une matrice d'émission formée de $N \times M$ probabilités conditionnelles dites termes d'émission. Chaque terme nous indique statistiquement la relation qui existe entre l'observation et l'état au même instant t au sein des deux processus stochastiques sous-jacents. De la même manière que pour la matrice A , la matrice B vérifie les conditions suivantes :

$$\begin{cases} b_j(V_k) \geq 0, \\ \sum_{k=1}^M b_j(V_k) = 1, \end{cases} \quad , \quad 1 \leq j \leq N, 1 \leq k \leq M \quad (3.6)$$

- Dans le cas continu, $B = \{b_j(O_t)\}_{1 \leq j \leq N}$ avec $b_j(O_t) = P(O_t \in R^N / X_t = S_j)$ est l'ensemble des N densités de probabilité. Dans ce cas, $b_j(O_t)$, qui est aussi dit terme d'émission, est une densité de probabilités quelconque. Puisque n'importe quelle densité de probabilité peut-être approximée par une somme de lois multinormales et afin de pouvoir estimer efficacement ces paramètres [110], cette densité peut alors être approximée par une somme de M lois multinormales \mathfrak{N} [110] (ou une somme de lois normales dans le cas où $O_t \in R$) de moyenne μ_{jm} et de covariance U_{jm} comme suit :

$$b_j(O_t) = \sum_{m=1}^M c_{jm} \mathfrak{N}(\mu_{jm}, U_{jm}; O_t) \quad (3.7)$$

où O_t est le vecteur des observations, \aleph est la loi multinormale et c_{jm} est le terme de pondération qui vérifie les conditions suivantes :

$$\begin{cases} c_{jm} \geq 0, \\ \sum_{m=1}^M c_{jm} = 1, \end{cases} \quad 1 \leq j \leq N, 1 \leq m \leq M, \quad (3.8)$$

Rappelons l'équation d'une loi multinormale \aleph de moyenne μ_{jm} et de covariance U_{jm} :

$$\aleph(\mu_{im}, U_{im}; O_t) = \frac{1}{(2\pi)^{\frac{N}{2}} |U_{im}|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (O_t - \mu_{im})^T U_{im}^{-1} (O_t - \mu_{im}) \right\} \quad (3.9)$$

Un modèle de MMC est formé par deux processus stationnaires. Un processus stationnaire possède des propriétés statistiques caractérisées par des espérances mathématiques qui ne dépendante pas du temps, ce qui nous permet d'écrire :

$$P(X_{t+1} = S_j / X_t = S_i) = P(X_t = S_j / X_{t-1} = S_i) = a_{ij}, \quad (3.10)$$

C'est-à-dire que la valeur de $\{a_{ij}\}_{1 \leq i, j \leq N}$ ne change pas au cours du temps.

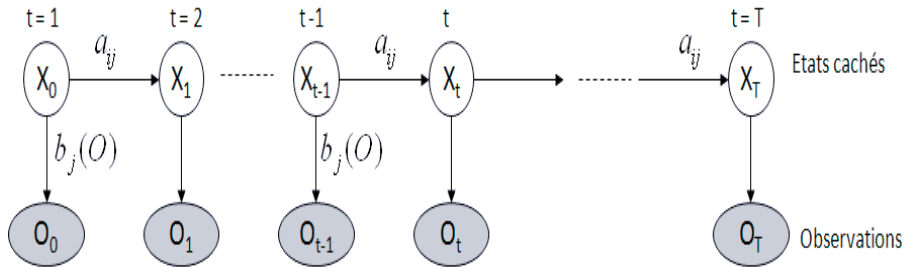


FIGURE 3.5 – Un Modèle de Markov Caché est stationnaire : $\{a_{ij}\}_{1 \leq i, j \leq N}$ évoluent de la même manière dans le temps

Les transitions entre les états et les transitions entre les états et les observations peuvent être représentées graphiquement. Ce qui nous permet de classifier les modèles

de MMC dans la section suivante en deux catégories selon la forme que peut prendre un graphe d'état [110].

3.1.2.2 Types de MMC

Modèle ergodique Au sein d'un modèle ergodique, n'importe quel état peut-être atteint depuis n'importe quel autre état du modèle comme le montre la figure (3.6)) :

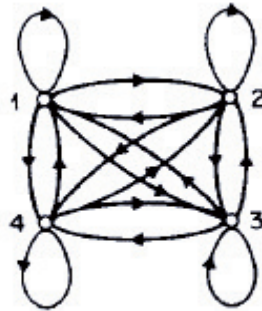


FIGURE 3.6 – Modèle ergodique

Par exemple, si on considère la matrice A telle que $A = \{a_{ij}\}_{1 \leq i, j \leq N}$ est de taille $N=4$ d'un modèle ergodique, tous les coefficients a_{ij} pour $1 \leq i, j \leq 4$ sont strictement supérieurs à θ .

Néanmoins, dans certaines applications telles que les applications sur la modélisation des comportements, d'autres types de MMC, comme le modèle de Bakis, permettent de mieux prendre en compte certaines propriétés observées sur le signal à modéliser.

Modèle de Bakis Connue sous le nom de left-right MMC, ce type de modèle considère que, quand le temps augmente, l'indice de l'état augmente à son tour ou reste le même. Les états vont alors évoluer de gauche à droite. D'où le terme left-right MMC.

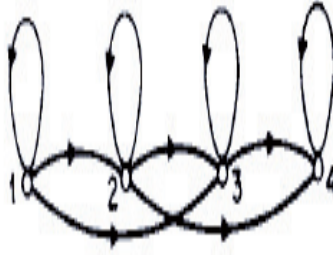


FIGURE 3.7 – Modèle de Bakis

La propriété principale d'un tel modèle est qu'aucune transition n'est permise pour les états ayant un indice inférieur à l'indice courant : $a_{ij} = 0$, pour $j < i$. Par ailleurs, un tel modèle a la propriété suivante :

$$\pi_i = \begin{cases} 1, & i = 1 \\ 0, & i \neq 1 \end{cases} \quad (3.11)$$

C'est-à-dire que la variable aléatoire à l'instant initial doit obligatoirement prendre la valeur de l'état initial S_1 .

D'autres contraintes ont aussi été rajoutées à ce type de modèle pour garantir que l'écart entre les indices de deux états en transition ne soit pas très grand :

$$a_{ij} = 0, \quad j > i + \Delta \quad (3.12)$$

Si on considère le modèle de Bakis de la figure (3.7), $\Delta = 2$, ce qui veut dire que l'écart entre i et j ne doit pas dépasser 2.

La matrice A précédente associée au graphe de la figure (3.7) s'écrit :

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{pmatrix}$$

Ainsi, les termes de transition associés au dernier état s'écrivent :

$$\begin{cases} a_{NN} & = 1 \\ a_{iN} & = 0, \quad i < N \end{cases} \quad (3.13)$$

Après avoir établi la structure et les paramètres des MMC, et afin de pouvoir utiliser en pratique un MMC dans différents domaines d'application, nous allons nous intéresser aux principales problématiques que cette approche permet de résoudre afin qu'elle soit utilisable en pratique. Ces problématiques concernent : l'évaluation du modèle, l'estimation des états cachés au vu des observations et la modélisation. Différentes solutions algorithmiques ont alors été proposées dans le cadre de cette approche.

Nous introduirons tout d'abord dans la section 3.1.2.3 ces problématiques, puis nous exposerons, dans la section 3.1.2.4, les solutions algorithmiques proposées.

3.1.2.3 Problématiques des MMC

L'évaluation du modèle La première problématique d'un MMC $\lambda = (\pi, A, B)$ concerne le calcul de la probabilité de la séquence d'observation $O = (O_1, \dots, O_t)$ produite par ce modèle λ , c'est-à-dire $P(O/\lambda)$.

On parle d'évaluation du modèle dans ce cas car ce problème peut aussi être vu comme étant un processus de calcul de l'adéquation de ce modèle λ avec la séquence d'observation O en question. Par exemple, si on dispose de n modèles $\lambda_1, \dots, \lambda_n$, la solution à ce problème nous permet de choisir le modèle $\lambda_{modèle}$ qui correspond au mieux à cette séquence d'observation O :

$$\lambda_{modèle} = \arg \max_{\lambda_i} (P(O/\lambda_i))_{1 \leq i \leq n} \quad (3.14)$$

Estimation des états La deuxième problématique consiste à dévoiler la partie cachée du modèle, c'est-à-dire la séquence d'états cachés qu'on notera $X^* = (X_1, \dots, X_t)$ associée à une séquence d'observation $O = (O_1, \dots, O_t)$. Étant donné que plusieurs séquences d'états peuvent être candidates, un critère d'optimalité doit être établi afin de trouver la séquence d'état la plus probable X^* associée à la séquence d'observation en question.

Modélisation (ou apprentissage) Cette problématique consiste à utiliser une séquence d'observation $O = (O_1, \dots, O_t)$ sur le modèle ainsi que le MMC λ initial afin d'ajuster ses trois paramètres π , A et B pour qu'il corresponde au mieux à la séquence d'observation O . Celle-ci est dite séquence d'entraînement ou séquence d'apprentissage car elle est utilisée pour apprendre le modèle. Ce problème est très important car il permet de mieux adapter le modèle initial à la séquence d'observation afin de créer un système plus proche du phénomène en question (réduction d'une certaine incertitude et incomplétude entre le modèle et la réalité).

$$\lambda_{opt} = \arg \max_{\lambda} (P(O/\lambda)) \quad (3.15)$$

Après avoir exposé les trois problématiques des MMC, nous regarderons dans la section suivante les différentes solutions apportées à ces problématiques.

3.1.2.4 Méthodes de résolution

L'algorithme forward-backward On veut calculer la probabilité de la séquence d'observation $O = (O_1, \dots, O_T)$ connaissant le modèle $\lambda : P(O/\lambda)$. Le calcul direct de cette probabilité se fait en énumérant tous les états possibles associés à chaque observation pour chaque pas temporel. On considère la séquence d'état $X = (X_1, \dots, X_T)$. Cette probabilité $P(O/\lambda)$ se fait en sommant la loi conjointe $P(O, X/\lambda)$ sur tous les états possibles $X_t \in \{S_1, \dots, S_N\}$, il s'agit de l'application d'une loi marginale (voir annexe A) sur X :

$$P(O/\lambda) = \sum_X P(O, X/\lambda) \quad (3.16)$$

La loi conjointe $P(O, X/\lambda)$ qui est la probabilité que O et X se produisent en même temps se décompose par la règle de Bayes (voir annexe A) comme suit :

$$P(O, X/\lambda) = P(O/X, \lambda)P(X/\lambda) \quad (3.17)$$

En insérant (3.17) dans (3.16), il vient :

$$P(O/\lambda) = \sum_X P(X/\lambda)P(O/X, \lambda) \quad (3.18)$$

$$= \sum_{X_1 \dots X_T} P(X_1 \dots X_T/\lambda)P(O_1 \dots O_T/X_1 \dots X_T, \lambda) \quad (3.19)$$

$$= \pi_{X_1} \sum_{X_2 \dots X_T} P(X_2/X_1, \lambda) \dots P(X_T/X_{T-1}, \lambda)P(O_1/X_1, \lambda) \dots P(O_T/X_T, \lambda) \quad (3.20)$$

$$= \pi_{X_1} \sum_X \prod_{t=1}^{T-1} a_{X_{t+1}X_t} \prod_{t=1}^T b_{X_t}(O_t) \quad (3.21)$$

Il est clair que le traitement direct de l'équation (3.21) est très compliqué et nécessite le calcul de $2TN^T$. En effet, pour chaque observation, il existe N états possibles. On a alors N^T séquences d'états possibles. Par ailleurs, pour chaque séquence d'état fixé (c'est-à-dire pour chaque terme de la somme de 3.21), on a besoin d'effectuer un calcul sur $2T$ termes. Ce type de calcul n'est pas faisable même pour une petite taille de N et de T . Pour cette raison, une procédure spécifique est nécessaire pour le calcul de $P(O/\lambda)$.

Cette procédure développée par Baum et al [9, 12] est dite l'algorithme du *forward-backward*.

L'algorithme du forward Cet algorithme doit son nom à la variable du forward $\alpha_t(i)$ qui est définie par : $\alpha_t(i) = P(O_1 \dots O_t, X_t = S_i/\lambda)$. La variable du forward est la probabilité partielle de la séquence d'observation jusqu'à l'instant t : O_1, \dots, O_t et de l'état X_t à l'instant t , connaissant le modèle λ . Le calcul de cette probabilité $\alpha_t(i)$ peut se faire de façon récursive comme suit :

1. Initialisation :

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (3.22)$$

2. Récursivité :

$$\alpha_{t+1}(i) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1, \quad 1 \leq j \leq N \quad (3.23)$$

3. Terminaison :

$$P(O/\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (3.24)$$

Au cours de l'étape 1, on initialise la variable du forward $\alpha_1(i)$ (équation (3.22)) qui est la probabilité jointe de l'état S_i et de l'observation initiale O_1 . La deuxième étape de récursivité (équation (3.23)) constitue le corps de l'algorithme, on peut rapidement démontrer cette récursivité :

$$\alpha_{t+1}(j) = P(O_1, \dots, O_{t+1}, X_{t+1} = S_j/\lambda) \quad (3.25)$$

$$= \sum_{i=1}^N P(O_1, \dots, O_{t+1}, X_{t+1} = S_j, X_t = S_i/\lambda) \quad (3.26)$$

$$= \sum_{i=1}^N P(O_1, \dots, O_t, X_t = S_j/\lambda) P(O_{t+1}/X_{t+1} = S_j, \lambda) \quad (3.27)$$

$$\begin{aligned} & \times P(X_{t+1} = S_j/X_t = S_i, \lambda) \\ & = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) \end{aligned} \quad (3.28)$$

L'équation (3.25) s'écrit en (3.26) en fonction de la loi marginale sur tous les états possibles S_i à l'instant t . Le passage de (3.26) à (3.27) s'est fait par application de la règle de Bayes. On retrouve ainsi l'équation (3.23).

Finalement, la troisième étape de terminaison nous donne la valeur finale de la probabilité de la séquence d'observation $O = (O_1, \dots, O_t)$ connaissant le modèle λ en effectuant la somme sur tous les termes finaux du forward, c'est-à-dire $\alpha_T(i)$.

Si on regarde dans ce cas le calcul de $\alpha_T(i)$, on peut voir qu'on a eu besoin d'effectuer un calcul sur N^2T termes à la place de $2TN^T$.

En réalité, l'algorithme du forward peut suffire pour résoudre le problème d'évaluation du modèle. Cependant, nous introduisons l'algorithme du backward car l'algorithme du forward-backward est indispensable pour la résolution du troisième problème, celui de la modélisation :

L'algorithme du backward De la même manière que pour l'algorithme du forward, cet algorithme doit son nom à la variable du backward $\beta_t(i)$ définie par $\beta_t(i) = P(O_{t+1}O_{t+2}\dots O_T/X_t = S_i, \lambda)$. La variable du backward est la probabilité partielle de la séquence d'observation de $t+1$ jusqu'à T sachant l'état X_t et le modèle λ . De la même manière que pour l'algorithme du forward, $\beta_t(i)$ peut-être résolu par récursivité selon les étapes suivantes :

1. Initialisation :

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \quad (3.29)$$

2. Récursivité :

$$\beta_t(i) = \sum_{j=1}^N a_{ij}b_j(O_{t+1})\beta_{t+1}(i), \quad t = T-1, T-2, \dots, 1, \quad 1 \leq i \leq N \quad (3.30)$$

A l'étape 1, $\beta_T(i)$ est arbitrairement défini à 1 pour toute valeur de i . La deuxième étape de récursion peut-aussi être démontrée de la même manière que dans (3.25)-(3.28).

On a

$$\beta_t(i) = P(O_{t+1}O_{t+2}O_{t+4}\dots O_T/X_t = S_i, \lambda) \quad (3.31)$$

$$= \sum_{j=1}^N P(O_{t+1}O_{t+2}O_{t+4}\dots O_T/X_{t+1} = S_j/X_t = S_i, \lambda) \quad (3.32)$$

$$= \sum_{j=1}^N P(O_{t+1}O_{t+2}O_{t+4}\dots O_T/X_t = S_i, X_{t+1} = S_j, \lambda) \quad (3.33)$$

$$\dots P(X_{t+1} = S_j/X_t = S_i, \lambda)$$

$$= \sum_{j=1}^N P(O_{t+1}O_{t+2}O_{t+4}\dots O_T/X_{t+1} = S_j, \lambda) \quad (3.34)$$

$$\dots P(X_{t+1} = S_j/X_t = S_i, \lambda)$$

$$= \sum_{j=1}^N P(O_{t+2}O_{t+4}\dots O_T/X_{t+1} = S_j, O_{t+1}, \lambda)P(O_{t+1}/X_{t+1} = S_j, \lambda) \quad (3.35)$$

$$\dots P(X_{t+1} = S_j/X_t = S_i, \lambda)$$

$$= \sum_{j=1}^N P(O_{t+2}O_{t+4}\dots O_T/X_{t+1} = S_j, \lambda)P(O_{t+1}/X_{t+1} = S_j, \lambda) \quad (3.36)$$

$$\dots P(X_{t+1} = S_j/X_t = S_i, \lambda)$$

$$= \sum_{j=1}^N \beta_{t+1}(j)b_j(O_{t+1})a_{ij} \quad (3.37)$$

L'équation (3.31) s'écrit en (3.32) en fonction de la loi marginale sur tous les états S_j à l'instant $t+1$. Les passages de (3.32) à (3.33) ainsi que de (3.34) à (3.35) se sont faits par la règle de Bayes. Le terme $O_{t+1}\dots O_T$ est indépendant de X_t connaissant X_{t+1} et $O_{t+1}O_{t+2}\dots O_T$ est indépendant de O_{t+1} connaissant X_{t+1} . D'où les simplifications respectives de (3.33) à (3.34) et de (3.35) à (3.36). On retrouve alors (3.37).

De la même manière que pour l'algorithme forward, l'algorithme backward permet un calcul de N^2T termes à la place de $2TN^T$.

Rappelons que l'algorithme du forward permet de calculer la probabilité $P(O/\lambda)$ et que nous avons introduit ici l'algorithme du forward-backward car il sera utilisé pour la résolution de la problématique de modélisation des MMC.

L'algorithme Viterbi La deuxième problématique associée au MMC est celle de l'estimation de la séquence d'état la plus probable $X^* = (X_1\dots X_t)$ associée à la

séquence d'observation $O = (O_1, \dots, O_t)$. Plusieurs critères d'optimalité de cette séquence d'état peuvent être envisageables. Par exemple, on peut choisir comme critère d'optimalité l'état le plus probable individuellement pour chaque pas temporel, c'est-à-dire :

$$X_t = \arg \max_{1 \leq i \leq N} P(X_t = S_i / O, \lambda) \quad (3.38)$$

Néanmoins, ce type de critère n'est pas toujours valide (voir [110]). Pour cette raison, dans quasiment toutes les applications, le critère du chemin d'état le plus probable a été choisi afin de maximiser $P(X_1 \dots X_T / O_1 \dots O_T, \lambda)$. La méthode utilisée dans ce cas est celle de l'algorithme Viterbi.

Il s'agit d'un algorithme issu de la programmation dynamique [55]. Il a été utilisé afin de déterminer le chemin le plus probable (l'ensemble des états $X^* = (X_1, \dots, X_t)$) associé à une séquence d'observation $O_1 \dots O_t$ d'un MMC. Cet algorithme se base sur un calcul itératif pour chaque état caché S_i du chemin le plus probable atteignant cet état à l'instant t de l'exécution du processus Markovien.

On note alors cette probabilité par :

$$\delta_t(i) = \max_{X_1 X_2 \dots X_{t-1}} P(X_1 X_2 \dots X_{t-1} = S_i, O_1 O_2 \dots O_t / \lambda) \quad (3.39)$$

où $\delta_t(i)$ représente la probabilité la plus élevée le long d'un chemin donné à l'instant t ayant donné cette séquence d'observation $O_1 \dots O_t$ en arrivant à l'état S_i . On obtient de façon récursive :

$$\delta_{t+1}(i) = \max_{S_j \in S} (\delta_t(j) a_{ij}) b_j(O_{t+1}) \quad (3.40)$$

Pour récupérer l'argument (l'état) qui maximise (3.40) à chaque pas temporel t et pour chaque j , on définit le vecteur $\psi_t(j)$ dans lequel on stockera cet argument.

Les différentes étapes de l'algorithme Viterbi qui nous permettent de trouver la séquence d'états la plus probable sont alors les suivantes :

1. Initialisation :

$$\delta_1(i) = \pi_i b_i(O_1) \quad 1 \leq i \leq N \quad (3.41)$$

$$\psi_1(i) = 0 \quad (3.42)$$

2. Récursivité :

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t) \quad 2 \leq t \leq T \quad 1 \leq j \leq N \quad (3.43)$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \quad 2 \leq t \leq T \quad 1 \leq j \leq N \quad (3.44)$$

3. Terminaison :

$$p^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (3.45)$$

$$X_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)] \quad (3.46)$$

4. Séquence d'état optimale :

$$X_t^* = \psi_{t+1}(X_{t+1}^*), \quad t = T - 1, T - 2, \dots, 1 \quad (3.47)$$

Cet algorithme prend en entrée le MMC λ et une séquence d'observation $O = (O_1 \dots O_t)$, et calcule le chemin de Viterbi X^* qui est le chemin d'état le plus probable par cette séquence d'observation O .

L'algorithme Baum-Welch Cette problématique de modélisation ou d'apprentissage est la problématique la plus importante pour les MMC. L'idée est de trouver une méthode permettant d'ajuster les paramètres d'un MMC $\lambda = (\pi, A, B)$ afin qu'il puisse maximiser la probabilité d'une séquence d'observation $(O_1 \dots O_t)$ associée à ce modèle.

Il n'existe aucune solution analytique permettant de trouver directement le modèle λ_{opt} qui maximise $P(O/\lambda)$ pour une séquence d'observation qui dépend d'une séquence d'état :

$$\lambda_{opt} = \arg \max_{\lambda} (P(O/\lambda)) \quad (3.48)$$

La solution qui a été proposée par Baum et al [11] était de choisir $\lambda = (\pi, A, B)$ tel que la fonction $P(O/\lambda)$ soit localement maximisée en utilisant une procédure itérative par un algorithme dit l'algorithme de Baum-Welch, ce qui revient à appliquer l'algorithme Espérance-Maximisation (EM). Etant donné la complexité des calculs analytiques, les différentes étapes d'estimation des paramètres d'une MMC λ par l'algorithme EM dans le cas discret et continu seront expliquées en annexe B.

Paramètres estimés par l'algorithme Baum-Welch Afin de comprendre la procédure de réestimation, nous définissons tout d'abord les deux termes ξ_t et γ_t , qui représentent respectivement la probabilité d'être conjointement à l'état S_i et à l'état S_j , et la probabilité d'être à l'état S_i , connaissant la séquence d'observation O et le modèle λ :

– Le premier terme :

$$\begin{aligned} \xi_t(i, j) &= P(q_t = S_i, q_{t+1} = S_j / O, \lambda) \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O/\lambda)} \end{aligned} \quad (3.49)$$

– Le deuxième terme :

$$\begin{aligned} \gamma_t(i) &= P(q_t = S_i / O, \lambda) \\ &= \sum_{j=1}^N \xi_t(i, j) \end{aligned} \quad (3.50)$$

Ces deux termes s'expriment en fonction des probabilités du forward et du backward : $\alpha_t(i)$ et $\beta_t(i)$.

Les termes estimés associés au vecteur d'initialisation π , à la matrice de transition d'états A et au terme d'émission B (dans le cas discret ou continu) seront exprimés en fonction de (3.49) et (3.50) comme suit :

1. La probabilité de transition d'états :

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad 1 \leq i, j \leq N, \quad (3.51)$$

2. La probabilité d'émission :

(a) le cas discret :

$$\bar{b}_j(O_t) = \frac{\sum_{t=1, O_t=v_k}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)}, \quad 1 \leq j \leq N, \quad 1 \leq k \leq M \quad (3.52)$$

(b) le cas continu :

Rappelons que la densité de probabilité des observations connaissant les états est une loi multinormale :

$$b_i(O) = \sum_{m=1}^M c_{im} N(\mu_{im}, U_{im}; O) \quad (3.53)$$

il s'agit alors de réestimer les valeurs de la moyenne μ_{im} , la covariance U_{im} et le coefficient c_{im} :

$$\bar{\mu}_{im} = \frac{\sum_{t=1}^T \gamma_t(i, m) O_t}{\sum_{t=1}^T \gamma_t(i, m)} \quad (3.54)$$

$$\bar{U}_{im} = \frac{\sum_{t=1}^T \gamma_t(i, m) (O_t - \mu_{im})(O_t - \mu_{im})^T}{\sum_{t=1}^T \gamma_t(i, m)} \quad (3.55)$$

$$\bar{c}_{im} = \frac{\sum_{t=1}^T \gamma_t(i, m)}{\sum_{t=1}^T \sum_{k=1}^M \gamma_t(i, m)} \quad (3.56)$$

3. La probabilité de l'état initial :

$$\bar{\pi}_i = \gamma_1(i), \quad 1 \leq i \leq N \quad (3.57)$$

Le calcul de ces termes (3.51)-(3.52), (3.54)-(3.57) est expliqué en annexe B.

L'algorithme Baum-Welch qui permet d'obtenir les probabilités estimées de l'état initial, les probabilité de transition d'état et les probabilités d'émission est un processus itératif qui peut se résumer par les étapes de la figure (3.8) :

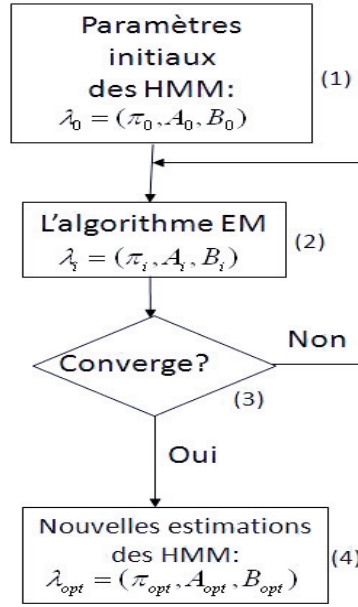


FIGURE 3.8 – L'algorithme Baum-Welch

- Etape (1) : Initialisation du MMC initial : $\lambda_0 = (\pi_0, A_0, B_0)$.
 - Etape (2) : Réestimation itérative des paramètres du MMC :
- Il a été prouvé par Baum et al [12] que cette réestimation par l'algorithme Baum-Welch des paramètres d'un MMC λ vérifie pour chaque itération :

$$P(O/\lambda_{i+1}) \geq P(O/\lambda_i) \quad (3.58)$$

avec i le numéro de l'itération, jusqu'à ce $P(O/\lambda_{i+1})$ devienne très proche de $P(O/\lambda_i)$:

$$P(O/\lambda_{i+1}) - P(O/\lambda_i) \approx 0 \quad (3.59)$$

- (3.58) exprime le fait qu'un modèle λ_{i+1} qui permet d'améliorer la probabilité de O a été trouvé. On est alors à l'étape (3) ; l'étape (2) sera réitérée.
- Si (3.59) est vérifiée, on dit que λ_{i+1} est un modèle optimal qui permet de maximiser $P(O/\lambda_{opt})$. Il s'agit de l'étape (4) et :

$$\lambda_{opt} = \arg \max_{\lambda} \log(P(O/\lambda)) \quad (3.60)$$

Notons que l'estimation des paramètres optimaux (3.51)-(3.52), (3.54)-(3.57) du MMC $\bar{\lambda}$ a été effectuée grâce à la fonction auxiliaire et les propositions introduites par Baum et ses collègues dans [11] :

- Fonction auxiliaire :

$$Q(\lambda, \bar{\lambda}) = \sum_X P(O, X/\lambda) (\log P(O, X/\bar{\lambda})) \quad (3.61)$$

- Proposition 1 :

Si la valeur de $Q(\lambda, \bar{\lambda})$ augmente, alors la valeur de $P(O/\bar{\lambda})$ augmente aussi, c'est-à-dire :

$$Q(\lambda, \bar{\lambda}) \geq Q(\lambda, \lambda) \implies P(O/\bar{\lambda}) \geq P(O/\lambda) \quad (3.62)$$

- Proposition 2 :

λ est un point critique de $P(O/\lambda)$ si et seulement il est un point critique de $Q(\lambda, \bar{\lambda})$ qui est une fonction de $\bar{\lambda}$, c'est-à-dire que :

$$\frac{\partial P(O/\lambda)}{\partial \lambda_i} = \frac{\partial Q(\lambda, \bar{\lambda})}{\partial \bar{\lambda}_i} \Big|_{\bar{\lambda}=\lambda}, 1 \leq i \leq D \quad (3.63)$$

où D est la dimension de λ et λ_i , $1 \leq i \leq D$, sont les éléments individuels de λ .

Normalisation des paramètres des MMC En effectuant les calculs de réestimation des paramètres d'un MMC par l'algorithme Baum-Welch, on a pu constater que la valeur de $\alpha_t(i)$ tend exponentiellement vers θ quand T augmente, ce qui provoque des dépassements mémoires au niveau informatique. Pour cette raison, une

phase de normalisation des paramètres du forward et du backward est indispensable. D'après [110], la variable $\alpha_t(i)$ serait alors normalisée par la somme sur i de tous les $\alpha_t(i)$:

$$\hat{\alpha}_t(i) = \frac{\alpha_t(i)}{\sum_{i=1}^N \alpha_t(i)} \quad (3.64)$$

Le paramètre de normalisation est alors :

$$c_t = \frac{1}{\sum_{i=1}^N \alpha_t(i)} \quad (3.65)$$

Et les nouvelles formules de normalisation de $\alpha_t(i)$ et de $\beta_t(i)$ seront :

$$\alpha_t(i) = \sum_{j=1}^N \hat{\alpha}_{t-1}(i) a_{ij} b_j(O_t) \quad (3.66)$$

$$\hat{\beta}_t(i) = c_t \beta_t(i) \quad (3.67)$$

Notons que les termes $\beta_t(i)$ seront normalisés par c_t .

Dans [123], il a été démontré que la formule (3.23) reste toujours vraie après normalisation. Nous allons dans la suite de cette section montrer que la probabilité $P(O/\lambda)$ va s'exprimer dans ce cas en fonction des termes de normalisation c_t .

On démontre tout d'abord par récurrence que :

$$\hat{\alpha}_t(i) = c_1 \dots c_t \alpha_t(i) \quad (3.68)$$

En effet, on considère que l'équation (3.68) est toujours vraie. Démontrons alors par récurrence que :

$$\hat{\alpha}_{t+1}(i) = c_1 \dots c_{t+1} \alpha_{t+1}(i) \quad (3.69)$$

On a :

$$\begin{aligned}
\hat{\alpha}_{t+1}(i) &= c_{t+1}\alpha_{t+1}(i) \\
&= c_{t+1} \sum_{j=1}^N \hat{\alpha}_t(i) a_{ij} b_j(O_t) \\
&= c_1 \dots c_t c_{t+1} \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_t) \\
&= c_1 \dots c_t c_{t+1} \alpha_{t+1}(i)
\end{aligned}$$

On retrouve alors (3.69). Par conséquent, (3.68) est toujours vraie.
En sommant tous les termes (3.64) sur j à l'instant T , on retrouve :

$$\sum_{j=1}^N \hat{\alpha}_T(j) = 1 \tag{3.70}$$

$$= \prod_{t=1}^T c_t \sum_{j=1}^N \alpha_T(i) \tag{3.71}$$

$$= \prod_{t=1}^T c_t P(O/\lambda) \tag{3.72}$$

Ainsi

$$P(O/\lambda) = \frac{1}{\prod_{t=1}^T c_t} \tag{3.73}$$

et

$$\log [P(O/\lambda)] = - \sum_{t=1}^T \log c_t \tag{3.74}$$

D'où la nouvelle formule du log-vraisemblance après normalisation.

Modélisation par un modèle de Bakis Dans le cas où une application donnée est modélisée par un modèle de Bakis, les contraintes (3.11), (3.12) et (3.13) sur ce modèle, citées dans la section 3.1.2.2 imposent l'utilisation de plusieurs séquences d'observations pour l'apprentissage de ce modèle. En effet, il n'est pas possible d'utiliser une séquence d'observation unique à cause du caractère transitoire du modèle de Bakis qui ne permet qu'un nombre limité d'observations pour chaque état.

Dans ce cas, le procédure de réestimation des paramètres par l'algorithme Baum-Welch est simple. Nous établirons alors dans ce cas les nouvelles formules.

On considère l'ensemble O de plusieurs séquences d'observation :

$$O = (O^{(1)}, O^{(2)}, \dots, O^{(K)}) \quad (3.75)$$

où $O^{(k)} = (O_1^{(k)}, O_2^{(k)}, \dots, O_{T_k}^{(k)})$ est la k^{eme} séquence d'observation.

On considère aussi que les séquences d'observations sont indépendantes les unes des autres, ce qui nous permet d'écrire :

$$P(O/\lambda) = \prod_{k=1}^K P(O^{(k)}/\lambda) \quad (3.76)$$

$$= \prod_{k=1}^K P_k \quad (3.77)$$

Ainsi, en tenant compte du caractère de normalisation décrit dans la section 3.1.2.4, les formules de réestimations de \bar{a}_{ij} et de \bar{b}_{jk} sont alors modifiées comme suit :

$$\bar{a}_{ij} = \frac{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_k-1} \hat{\alpha}_t^{(k)}(i) a_{ij} b_j(O_{t+1}^{(k)}) \hat{\beta}_{t+1}^{(k)}(j)}{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_k-1} \hat{\alpha}_t^{(k)}(i) \hat{\beta}_{t+1}^{(k)}(j)} \quad (3.78)$$

$$\bar{b}_{jk} = \frac{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_k-1} \hat{\alpha}_t^{(k)}(k) \hat{\beta}_t^{(k)}(k)}{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_k-1} \hat{\alpha}_t^{(k)}(k) \hat{\beta}_t^{(k)}(k)} \quad (3.79)$$

Conclusion

Nous avons introduit dans cette section la théorie des MMC que nous avons utilisé pour la construction du sous-module de reconnaissance de comportement par approche probabiliste (BPAM) du module global (BAM). Ce module est destiné à la reconnaissance des gros bateaux qui évoluent dans le port. L'utilisation de cette approche pour la construction de ce sous-module du BAM sera présentée dans la section 4.1 du chapitre 4.

3.2 L'approche événementielle par langage réactif synchrone

3.2.1 Généralités

3.2.1.1 Les systèmes réactifs synchrones

Un système réactif synchrone est un système en interaction constante avec son environnement. Le rythme de ses réactions est dicté par l'environnement : il reçoit des événements de son environnement auxquels il réagit instantanément en émettant des événements vers son environnement. Instantané signifie ici que les sorties sont fournies d'une façon absolument synchrone aux entrées [23, 97], à chaque fois que l'environnement va fournir de nouvelles entrées, le système sera toujours prêt à les traiter et aucune entrée ne pourra être perdue.

Les systèmes temps-réels sont des systèmes en interaction avec leur environnement qui se caractérisent par leur capacité à respecter des échéances, c'est à dire que les réponses doivent arriver en un temps fini et prédictible. Ce type de système informatique prend donc en compte des contraintes temporelles dont le respect est aussi important que l'exactitude des résultats (un résultat qui arrive trop tard sera considéré comme erroné même si il est le résultat de calculs justes). Ces systèmes sont largement présents dans de nombreux secteurs d'activités : dans l'industrie de production au travers des systèmes de contrôle de procédé (usines, centrales nucléaires, etc.), dans l'aéronautique et le spatial au travers des systèmes de pilotage embarqués, ou encore dans le secteur de la nouvelle économie au travers du besoin, toujours croissant, du traitement et de l'acheminement de l'information (vidéo, données, pilotage à distance, réalité virtuelle, etc). Le développement de systèmes temps-réel nécessite donc la prise en compte de contraintes temporelles de manière à garantir que les services et les résultats soient fournis aux délais souhaités. Leur temps d'exécution doit ainsi être prédictible. Les systèmes temps-réels sont des systèmes réactifs synchrones.

3.2.1.2 Les langages réactifs synchrones

Dans cet important contexte de développement de systèmes logiciels souvent embarqués et critiques, une famille de langages a vu le jour dans les années 80 : les langages formels réactifs synchrones. Notons qu'un langage informatique est dit formel quand il possède une sémantique mathématiquement fondée permettant que la signification des programmes ne soit jamais ambiguë. Les trois langages fondateurs de ce domaine sont :

- ESTEREL [22] qui est un langage impératif pour programmation des systèmes dominés par le contrôle. Il a été inventé et développé par notre centre de recherche, le Centre de Mathématiques Appliquées (CMA) ;
- LUSTRE [101] qui est un langage déclaratif pour la programmation des systèmes flot de données ;
- SIGNAL [17] qui est un langage déclaratif pour la programmation de systèmes multi-horloges.

Ces langages réactifs synchrones se sont révélés être des formalismes mathématiques de haut-niveau, robustes et expressifs pour la spécification et l'implémentation des systèmes réactifs. Ces langages et leurs environnements de programmation ont été utilisés de manière intensive dans l'industrie pour la programmation des sous parties réactives critiques d'applications importantes dans des domaines comme l'aéronautique, le spatial et les télécommunications. Par ailleurs, le langage LUSTRE a donné lieu à des outils de conception industrialisés comme la SCADE Suite [27].

Le terme *système réactif* a été introduit par Nicolas Halbwachs dans le document [63] pour désigner les systèmes en interaction permanente avec leur environnement. Le comportement d'un système réactif est similaire à celui d'une Machine à Etats Finis ou en anglais Finite State Machine. L'exécution est très efficace et des outils appelés *model checkers* [43] permettent de vérifier formellement leur comportement.

3.2.1.3 Les autres types de systèmes informatiques

Notons qu'en plus des systèmes réactifs, il existe deux autres catégories de systèmes informatiques :

- *Les systèmes transformationnels* qui s'assimilent au calcul d'une fonction. Une fois les entrées fournies au programme, celui-ci effectue son calcul et termine en renvoyant le résultat. On peut citer comme exemple un compilateur ou encore l'inversion d'une matrice. Le temps de réponse dépend de la complexité de l'algorithme de calcul implémenté.
- *Les systèmes interactifs* sont en interaction constante avec leur environnement dont ils reçoivent des requêtes auxquelles ils réagissent à leur propre rythme. Citons comme exemple un système d'exploitation classique ou encore une recherche sur Internet. Les temps de réponse sont indéterminés et imprédictibles.

3.2.1.4 Implantation d'un système réactif synchrones

Lors de son implantation dans une application complexe, un système réactif synchrone [?] est formé de deux parties :

- La *machine d'exécution* qui interagit avec l'environnement est chargée de la lecture des informations provenant de l'environnement (systèmes de détection, capteurs, etc) de manière à transformer tout événement physique externe en événement logique interne compréhensible par le noyau réactif et inversement.
- Le *noyau réactif* contient la partie logique du système. Il réagit aux entrées logiques, en déclenchant la réaction adéquate en terme de calcul et d'émission de sorties logiques.

3.2.2 Un langage événementiel : ESTEREL

ESTEREL est un langage de programmation pour les systèmes réactifs synchrones développé au cours des années 80 dans une équipe de recherche commune à l'Institut National de Recherche en Informatique et en Automatique (INRIA) et au Centre de Mathématiques Appliquées (CMA) de l'Ecole des Mines de Paris.

C'est un langage impératif orienté contrôle dont les opérateurs sont munis d'une sémantique mathématique.

3.2.2.1 Concepts de base du langage

Le langage ESTEREL peut se décrire de la manière suivante [16] :

- *Modularité* : l'unité de programmation d'ESTEREL est le *module*. L'approche modulaire a été introduite dans le langage ESTEREL par souci d'expressivité, c'est une facilité de programmation qui permet de découper un programme en sous-parties logiques qui seront ultérieurement combinées.
- *Réaction* : un module implémenté dans le langage ESTEREL réagit *instantanément*, à des événements d'entrée, en émettant des événements de sortie et redevient immédiatement prêt à réagir une prochaine fois. Ce comportement est appelé une *réaction* ; rappelons que la caractéristique principale d'un système réactif est que tous les calculs doivent impérativement être finis avant que d'autres signaux d'entrée ne se présentent.
- *Temps logique* : lors de l'exécution d'un système réactif synchrone, la suite des *réactions* définit l'*horloge logique* du système ; chaque *réaction* étant considérée comme un *instant* de ce *temps logique*. « tick » est le signal définissant ce temps logique, il est présent à chaque réaction du système. Notons que le temps physique ne joue aucun rôle privilégié dans l'approche réactive et sera simplement introduit au besoin comme n'importe quel autre signal d'entrée.
- *Les signaux* : les informations sont échangées entre le système et son environnement ou entre les sous-modules par le biais des signaux. Un système réactif synchrone reçoit des signaux d'entrée et émet des signaux de sortie. Au cours

d'une réaction, un signal peut-être *présent* ou *absent* mais il garde impérativement ce même statut tout au long de la réaction. Il peut transporter une valeur dont le type sera soit l'un des types prédéfinis dans le langage, soit un type abstrait défini par l'utilisateur.

- *Signaux locaux* : les signaux servent aussi de moyens de communication interne entre sous-parties du système. Un signal peut alors être local à un module, il sera possible de l'émettre mais aussi de le recevoir. Un signal local est caché à l'extérieur, il participe simplement à l'expression d'un comportement interne du système.
- *Communication par diffusion* : les statuts de présence et d'absence des signaux pendant une réaction sont visibles et cohérents dans toutes les sous-parties du programme. L'émission d'un signal est instantanément perçue par toutes les parties concurrentes du programme.
- *Approche synchrone* ou *Hypothèse de temps de réaction nul* : la réaction étant considérée comme instantanée, au cours d'une réaction, les signaux de sortie peuvent être considérés comme synchrones avec les signaux d'entrée.
- *La compilation d'un programme réactif synchrone* : pour un langage réactif synchrone, la phase de compilation est particulièrement délicate et a donné matière à plusieurs thèses au sein du CMA [75, 46, 59, 91]. Comme pour le signal local, le parallélisme est complètement résolu au cours de la compilation et on passe ainsi d'un code source parallèle à un code compilé séquentiel. Le code compilé d'un programme réactif synchrone est équivalent à une machine à états finis où les réactions sont les transitions entre les états. Pour cette raison, si le processus de compilation d'un programme réactif synchrone hiérarchique est fortement complexe, l'exécution d'un code compilée est très rapide (comparable au temps de transition d'un automate) : ceci constitue la force de l'approche réactive synchrone.
- *L'exécution* : un programme réactif synchrone est une séquence infinie de réactions.

Pour respecter l'hypothèse synchrone, c'est-à-dire pour assurer qu'un système réactif réagisse instantanément, certaines conditions ont été imposées :

- il n'y a pas d'allocation dynamique d'espace mémoire au cours de l'exécution du programme, tout l'espace requis existe avant l'exécution (allocation statique) ;
- dans un programme ESTEREL, la manipulation des données est réduite à des opérations simples sur ces données (comparaison, soustraction, addition, etc). Il n'est pas possible d'exécuter des boucles sur les données. Les fonctions et procédures utilisées dans un programme ESTEREL sont considérées comme ayant un comportement instantané ;

- l’exécution des opérateurs eux-mêmes comme le « ; » ainsi que le test ou l’émission d’un signal est instantané.

3.2.2.2 Les modules ESTEREL

L’unité de programmation en ESTEREL est la *module*. Un module est formé d’une partie *déclaration* et d’une partie comportement réduite à une *instruction*. Les types de données, les constantes, les fonctions et les procédures sont importés d’un langage hôte, et sont simplement déclarés comme des noms abstraits dans la partie *déclaration*. Le langage ne contient qu’un ensemble réduit de types (les entiers, les flottants, les doubles, les chaînes de caractères et les booléens) et de leurs opérateurs respectifs arithmétiques et logiques [97].

Dans cette section, nous nous intéresserons uniquement à une sous partie de la syntaxe du programme ESTEREL. Pour plus de détail, le lecteur trouvera une présentation complète de ce langage dans [24].

Les déclarations Dans la partie déclaration, on peut trouver les types, les constantes, les signaux, les fonctions et les procédures utilisées par le module qui seront définies dans le langage hôte. Voici un exemple de la partie déclaration d’un module :

```

module VersLavera : nom du module
input BateauCommercial, rapide, zoneInterdite : déclaration des signaux d’entrée
output alarm 1, alarm 2, alarm 3 : déclaration des signaux de sortie
...
end module

```

Les instructions Le langage ESTEREL se base sur les instructions suivantes :

- l’instruction vide : **nothing ;**
- l’arrêt : **halt ;**
- l’attente d’un signal ou d’une expression de signaux : **await ;**
- l’émission d’un signal : **emit ;**
- la levée d’une exception : **exit ;**
- la séquence entre deux instructions : « ; » ;
- la boucle infinie : **loop...end ;**
- le test de présence d’un signal ou d’une expression de signaux : **present...then...else...end ;**
- l’instruction conditionnelle sur les données : **if...then...else...end ;**

- la préemption forte d'une instruction par un signal ou une expression de signaux : **abort...when** ;
- la préemption faible d'une instruction par un signal ou une expression de signaux : **weak abort...when**
- le parallélisme entre deux instructions : « || » ;
- la déclaration d'exception : **trap...end** ;
- la déclaration du signal local : **signal...in...end** ;
- l'affectation d'une expression à une variable : « := » ;
- la déclaration de variable locale : **var...in...end** ;

D'autres instructions peuvent être dérivées des instructions listées ci-dessus comme par exemple : « **loop stat each R** » qui est équivalente à : « **loop abort stat ;halt when R end** ».

Nous pouvons classer ces instructions dans un tableau (voir tableau (3.1)) selon qu'elles manipulent les signaux ou non :

Instructions qui ne manipulent pas des signaux	Instructions qui manipulent des signaux
nothing	emit S
halt	emit $S(exp)$
var := exp	present S then $stat_1$ else $stat_2$
$stat_1;stat_2$	present $S(exp)$ then $stat_1$ else
loop $stat$ end	$stat_2$
$stat_1 stat_2$	abort $stat$ when S
trap var in	trap T end
exit var	await S
var var_dec in	signal $signal_dec$ in $stat$ end

TABLE 3.1 – Instructions manipulant ou non les signaux

En ESTEREL, certains programmes sont « *interdits* » c'est à dire qu'ils n'auront pas de signification dans la sémantique comportementale d'ESTEREL. A titre d'exemple, la boucle instantanée : « $X := 0$; **loop** $X := X+1$ **end** » n'a pas de sens puisque son comportement serait dans l'instant, où la boucle s'exécute, d'affecter une infinité de fois 1 à la variable X , de même le programme « **loop nothing end** » consiste, dans l'instant où il s'exécute, à exécuter une infinité de fois l'instruction **nothing**, ce qui ne revêt pas de sens non plus.

3.2.2.3 La sémantique intuitive

Nous allons dans cette section exprimer de manière simple et non formelle le comportement des opérateurs du langage ESTEREL (la sémantique intuitive) :

- **nothing** : cette instruction ne fait rien et termine instantanément ;
- **halt** : cette instruction ne fait rien et ne termine pas ;
- **emit** S : cette instruction émet le signal S dans l’environnement et termine. Le signal S devient présent dans toutes les parties actives du programme en cours d’exécution ;
- **exit** T : cette instruction lève l’exception T . L’instruction **exit** ne termine pas ;
- **stat1 ; stat2** : l’instruction $stat1$ est exécutée en premier. Lorsqu’elle termine, l’exécution de l’instruction $stat2$ commence immédiatement (l’exécution de l’opérateur « ; » lui-même ne prend pas de temps). Si $stat1$ ne termine pas, $stat2$ ne sera pas exécutée et la séquence ne terminera pas. Par exemple, dans le programme « **halt ; emit** O », le signal O n’est jamais émis ;
- **loop** $stat$ **end** : $stat$ commence à être exécutée, quand elle termine, on recommence à l’exécuter (comme si $stat$ était en séquence avec elle-même). L’instruction **loop** ne termine pas ;
- **present** S **then** $stat1$ **else** $stat2$ **end** : si le signal S est présent, l’instruction $stat1$ est exécutée sinon c’est $stat2$ qui le sera. Le test de présence du signal S ne prend pas de temps ;
- **abort** $stat$ **when** S : au premier instant l’instruction $stat$ commence à être exécutée quelque soit le statut (*présence* ou *d’absence*) de S . Aux instants suivants, le programme termine si $stat$ termine ou si le signal S est présent. Cette préemption est une préemption forte c’est à dire qu’à l’instant où S est présent, $stat$ n’est pas exécutée. Prenons par exemple le programme « **abort** **await** A ; **emit** O **when** S ». Au premier instant, où ce programme s’exécute, il se bloque en attente de A . Dans les instants suivants, si A est présent et S est absent alors O est émis et le programme termine. Si A et S sont présents, le programme termine instantanément sans émettre le signal O . Si on remplace **abort** par un **weak abort**, lorsque A et S sont présents au même instant, le signal O est quand même émis avant que le programme ne termine ;
- « $stat1$ || $stat2$ » : l’exécution des instructions $stat1$ et $stat2$ commence simultanément et terminera quand $stat1$ et $stat2$ auront tous les deux terminé. Si une des branches du parallèle ne termine jamais, le parallèle ne terminera pas ;
- **trap** T **in** $stat$ **end** : $stat$ commence à s’exécuter. A l’instant où l’exception T est levée, le programme finit l’exécution de son instant courant et termine. Au cas où des exceptions sont imbriquées, l’exception la plus englobante est

- prioritaire et détermine la fin du programme ;
- **signal** S **in** $stat$ **end** : la portée de S est réduite à l’instruction $stat$;
- $X := exp$: calcule instantanément la valeur de l’expression exp , met à jour la valeur de X dans la mémoire à cette valeur et termine instantanément ;
- **var** $X : integer := 0$ **in** $stat$ **end** : déclare une variable entière X dont la portée est réduite à $stat$ et dont la valeur initiale est 0. Comme dans le cas du signal local, cette variable n’est pas accessible à l’extérieur de $stat$;
- **await** S : au premier instant, cette instruction se bloque en attente de S même si S est présent. Aux instants suivants, si S est présent l’instruction termine sinon elle reste en attente de S . Notons que cette instruction peut s’exprimer en fonction d’instructions de base « **await** $S = \mathbf{abort\ halt\ when\ } S$ ».

Cette sémantique intuitive n’est qu’une explication dans un langage naturel, ici le français, de comment sont supposés se comporter les opérateurs. Elle peut prêter à confusion voire à erreur suivant la manière dont on interprète le français. C’est pourquoi, afin d’éviter toute confusion dans l’interprétation des programmes, le langage ESTEREL a été muni d’une sémantique exprimée dans un langage universel et formel : les mathématiques. Nous allons introduire, dans la section suivante, la sémantique comportementale du langage ESTEREL.

3.2.2.4 La sémantique comportementale

Les opérateurs précédents sont munis d’une sémantique sous forme de règles de sémantiques opérationnelles. Cette sémantique permet d’interpréter un programme de manière unique et d’éviter toute ambiguïté quant à son interprétation. La compilation des programmes, quant à elle, obéit à une sémantique d’exécution que nous ne détaillerons pas ici [20].

Nous présenterons les règles de réécriture de la sémantique comportementale pour les opérateurs précédemment définis dans la section 3.2.2.3, cette sémantique est présentée de manière complète dans [24].

Le format des règles : Comme dans [91], nous définissons l’exécution d’un programme ESTEREL M sur une suite d’instantants $t_1, \dots, t_{n+1}, \dots$ comme étant la production de l’histoire des ensembles des signaux de sortie $O_1, O_2, \dots, O_n, \dots$ correspondant à l’histoire de l’ensemble des signaux d’entrée $I_1, I_2, \dots, I_n, \dots$. Cette exécution peut-être représentée par une séquence de transitions de la forme :

$$M = M_1 \xrightarrow[I_1]{O_1} M_2 \xrightarrow[I_2]{O_2} M_3 \dots M_n \xrightarrow[I_n]{O_n} M_{n+1} \dots$$

où chaque M_n est un module ESTEREL ayant la même interface que M et calculant l'histoire (O_n, O_{n+1}, \dots) de l'ensemble des signaux de sortie de M à partir du pas n . Le passage d'un module M_n à un module M_{n+1} sera appelé transition car chaque module M_n peut-être associé à un état d'un automate représentant le comportement du module M .

Nous allons utiliser la même règle d'écriture (3.80) adoptée par [24] afin de décrire le comportement des différentes instructions citées dans la section 3.2.2.3.

$$\langle stat, \rho \rangle \xrightarrow{E', b, T} \langle stat', \rho' \rangle \quad (3.80)$$

où :

- ρ et ρ' sont les états de mémoires des variables associées à $stat$ et à $stat'$.
- E est l'ensemble des signaux d'entrée et des signaux locaux présents dans lequel $stat$ est exécuté ; un signal est présent dans l'environnement d'exécution s'il a été donné par l'extérieur (signaux d'entrée) ou émis par le programme lui même (signaux locaux).
- E' est l'ensemble des signaux émis après avoir exécuté $stat$ une seule fois.
- b est la *terminaison*. C'est une information booléenne qui nous renseigne sur l'état du programme lors de son passage de $stat$ à $stat'$: b prendra t (pour true) si le programme termine à l'instant ou f (pour false) sinon.
- T est l'ensemble des exceptions levées lors de l'exécution de $stat$.
- $stat'$ est le résultat d'exécution de $stat$.

La règle (3.80) nous indique que $stat$ s'exécute dans l'ensemble des événement E en allouant ou modifiant la mémoire ρ qui devient ρ' . L'exécution de $stat$ donne $stat'$ qui s'exécute ensuite. Cette exécution a levé les exceptions dans T et termine si b est vrai ($b = t$) ou en ne termine pas si b est faux ($b = f$).

En plus de la règle (3.80), nous définirons les règles suivantes :

$$\langle exp, \rho \rangle \rightarrow v \quad (3.81)$$

La formule (3.81) est la règle d'évaluation d'une expression si exp est une constante, nous écrirons :

$$\langle C, \rho \rangle \rightarrow c \quad (3.82)$$

Les règles d'écriture

- La règle pour **nothing** :

$$\langle \mathbf{nothing}, \rho \rangle \xrightarrow[E]{\phi, t, \phi} \langle \mathbf{nothing}, \rho \rangle$$

Le **nothing** termine instantanément et ne modifie pas la mémoire.

- La règle pour **halt** :

$$\langle \mathbf{halt}, \rho \rangle \xrightarrow[E]{\phi, f, \phi} \langle \mathbf{halt}, \rho \rangle$$

Le **halt** ne termine jamais et se réécrit en **halt**.

- La règle du **emit** S :

$$\langle \mathbf{emit} S, \rho \rangle \xrightarrow[E]{\{S\}, t, \emptyset} \langle \mathbf{nothing}, \rho \rangle$$

L'instruction **emit** S émet le signal S et termine instantanément.

- La règle du **exit** T :

$$\langle \mathbf{exit} T, \rho \rangle \xrightarrow[E]{\phi, f, \{T\}} \langle \mathbf{halt}, \rho \rangle$$

L'**exit** ne termine jamais et se réécrit en **halt**.

- La **séquence** comporte deux règles suivant que la première instruction termine à l'instant ou non : $stat_1; stat_2$

1. Si $stat_1$ termine dans l'instant, l'exécution de $stat_2$ est commencée dans le même instant

$$\frac{\langle stat_1, \rho_1 \rangle \xrightarrow[E]{E'_1, t, \phi} \langle stat'_1, \rho'_1 \rangle \quad \langle stat_2, \rho'_1 \rangle \xrightarrow[E]{E'_2, b_2, T_2} \langle stat'_2, \rho'_2 \rangle}{\langle stat_1; stat_2, \rho \rangle \xrightarrow[E]{E'_1 \cup E'_2, b_2, T_2} \langle stat'_2, \rho'_2 \rangle}$$

2. Si $stat_1$ ne termine pas, la séquence reste bloquée dans l'exécution de $stat_1$ et $stat_2$ n'est pas exécutée

$$\frac{\langle stat_1, \rho \rangle \xrightarrow[E]{E'_1, f, T_1} \langle stat'_1, \rho'_1 \rangle}{\langle stat_1; stat_2, \rho \rangle \xrightarrow[E]{E'_1, f, T_1} \langle stat'_1; stat_2, \rho'_1 \rangle}$$

– La règle pour **if** exp **then** $stat_1$ **else** $stat_2$ **end**

1. Si exp s'évalue à vrai, c'est la branche **then** $stat_1$ qui commence à être exécutée :

$$\frac{\langle exp, \rho \rangle \rightarrow true \quad \langle stat_1, \rho \rangle \xrightarrow[E]{E'_1, b, T_1} \langle stat'_1, \rho'_1 \rangle}{\langle \text{if } exp \text{ then } stat_1 \text{ else } stat_2 \text{ end}, \rho \rangle \xrightarrow[E]{E'_1, b, T_1} \langle stat'_1, \rho'_1 \rangle}$$

2. Si exp s'évalue à faux, c'est la branche **else** $stat_2$ qui commence à être exécutée :

$$\frac{\langle exp, \rho \rangle \rightarrow false \quad \langle stat_2, \rho \rangle \xrightarrow[E]{E'_2, b, T_2} \langle stat'_2, \rho'_2 \rangle}{\langle \text{if } exp \text{ then } stat_1 \text{ else } stat_2 \text{ end}, \rho \rangle \xrightarrow[E]{E'_2, b, T_2} \langle stat'_2, \rho'_2 \rangle}$$

– La règle pour **present** S **then** $stat_1$ **else** $stat_2$ **end** : elle est similaire à celle du **if...then...else...end** :

1. Si S est *présent* dans E , on exécute la branche **then**

$$\frac{S^+ \in E \quad \langle stat_1, \rho \rangle \xrightarrow[E]{E'_1, b_1, T_1} \langle stat'_1, \rho'_1 \rangle}{\langle \text{present } S \text{ then } stat_1 \text{ else } stat_2 \text{ end}, \rho \rangle \xrightarrow[E]{E'_1, b_1, T_1} \langle stat'_1, \rho'_1 \rangle}$$

2. Si S est *absent* dans E , on exécute la branche **else**

$$\frac{S^- \in E \quad \langle stat_2, \rho \rangle \xrightarrow[E]{E'_2, b_2, T_2} \langle stat'_2, \rho'_2 \rangle}{\langle \mathbf{present} S \mathbf{ then } stat_1 \mathbf{ else } stat_2 \mathbf{ end}, \rho \rangle \xrightarrow[E]{E'_2, b_2, T_2} \langle stat'_2, \rho'_2 \rangle}$$

– La règle du **await** S :

Remarquons qu'au premier tick même si le signal S est présent, la branche se bloque en attente de S :

$$\langle \mathbf{await} S, \rho \rangle \xrightarrow[E]{\phi, f, \phi} \langle \mathbf{present} S \mathbf{ then } \mathit{nothing} \mathbf{ else } \mathbf{await} S, \rho \rangle$$

– La règle du **parallèle** « \parallel » :

$$\frac{\langle stat_1, \rho \rangle \xrightarrow[E]{E'_1, b_1, T_1} \langle stat'_1, \rho'_1 \rangle \quad \langle stat_2, \rho \rangle \xrightarrow[E]{E'_2, b_2, T_2} \langle stat'_2, \rho'_2 \rangle}{\langle stat_1 \parallel stat_2, \rho \rangle \xrightarrow[E]{E'_1 \cup E'_2, b_1 \text{ et } b_2, T_1 \cup T_2} \langle stat'_1 \parallel stat'_2, \mathit{merge}(\rho, \rho'_1, \rho'_2) \rangle}$$

– La règle pour la **loop** $stat$ **end** :

La sémantique de la boucle traduit le fait qu'il est interdit à celle-ci de terminer dans l'instant où elle commence son exécution :

$$\frac{\langle stat; \mathbf{loop} stat \mathbf{end}, \rho \rangle \xrightarrow[E]{E', f, T} \langle stat', \rho' \rangle}{\langle \mathbf{loop} stat \mathbf{end}, \rho \rangle \xrightarrow[E]{E', f, T} \langle stat', \rho' \rangle}$$

– La règle pour **abort** $stat$ **when** S

le module $stat$ est exécuté en $stat'$ tant que le signal S n'est pas présent

$$\frac{\langle stat, \rho \rangle \xrightarrow[E]{E', b, T} \langle stat', \rho' \rangle}{\langle \mathbf{abort} stat \mathbf{when} S, \rho \rangle \xrightarrow[E]{E', b, T} \langle \mathbf{present} S \mathbf{ else } \mathbf{abort} stat' \mathbf{when} S, \rho' \rangle}$$

– La règle de la **Trap** U **in** $stat$ **end**

1. le **trap** termine si $stat$ termine ou si l'ensemble des exceptions levées est réduit à l'exception $\{U\}$.

– exemple1 : **trap** U **in** **emit** V **end** : le signal V est émis et le trap termine dans l'instant.

– exemple 2 : **trap** U **in** **exit** U ; **emit** O **end** : l'exception U est levée et le trap termine sans que O ne soit émis.

$$\frac{\langle stat, \rho \rangle \xrightarrow[E]{E', b, T} \langle stat', \rho' \rangle \quad b = true \text{ ou } T = \{U\}}{\langle \mathbf{trap} U \mathbf{in} stat \mathbf{end}, \rho \rangle \xrightarrow[E]{E', b, \phi} \langle \mathbf{nothing}, \rho' \rangle}$$

2. le **trap** U ne termine pas : ceci correspond au cas où soit $stat$ ne termine pas soit $stat$ lève des exceptions autres que U par exemple dans le cas où on a des trap emboîtées.

– exemple :

```

trap  $W$  in
  trap  $V$  in
    trap  $U$  in
      exit  $W$ 
    end
  end
end

```

$$\frac{\langle stat, \rho \rangle \xrightarrow[E]{E', f, T} \langle stat', \rho' \rangle}{\langle \mathbf{trap} U \mathbf{in} stat \mathbf{end}, \rho \rangle \xrightarrow[E]{E', f, T \setminus \{U\}} \langle \mathbf{trap} U \mathbf{in} stat' \mathbf{end}, \rho' \rangle}$$

avec $T \setminus \{U\}$ est l'ensemble des exceptions levées autre que U .

Pour la règle du signal local, nous définissons les notations suivantes :

$E \setminus \{S\}$: L'ensemble des événements E à l'exception du signal S quand il est présent.

S^+ : Le signal S considéré comme émis (présent).

S^- : Le signal S considéré comme non-émis (absent).

– La règle du **signal local : signal S in $stat$ end**

1. Cas où je considère S présent, il faut alors qu'il soit émis par l'exécution de $stat$ au cours du même instant :

$$\frac{\langle stat, \rho \rangle \xrightarrow[(E \setminus \{S\}) \cup \{S^+\}]{E', b, T} \langle stat', \rho' \rangle \quad S^+ \in E'}{\langle \mathbf{signal} \ S \ \mathbf{in} \ stat \ \mathbf{end}, \rho \rangle \xrightarrow[E]{E', b, T} \langle \mathbf{signal} \ S' \ \mathbf{in} \ stat' \ \mathbf{end}, \rho' \rangle}$$

2. Je considère S absent, il ne faut pas qu'il soit émis par l'exécution de $stat$ au cours de ce même instant :

$$\frac{\langle stat, \rho \rangle \xrightarrow[(E \setminus \{S\}) \cup \{S^-\}]{E', b, T} \langle stat', \rho' \rangle \quad S^- \in E'}{\langle \mathbf{signal} \ S \ \mathbf{in} \ stat \ \mathbf{end}, \rho \rangle \xrightarrow[E]{E', b, T} \langle \mathbf{signal} \ S' \ \mathbf{in} \ stat' \ \mathbf{end}, \rho' \rangle}$$

3.2.2.5 Compilation du programme ESTEREL en automate

D'après [24], il a été montré que tout programme ESTEREL correct peut se compiler en un automate à états finis qui réalise à chaque instant une transition d'un état à un autre. Pour calculer l'automate équivalent à un programme ESTEREL, il y a eu plusieurs générations de compilateurs :

La compilation d'un programme ESTEREL par le compilateur ESTEREL actuel consiste en la traduction des fichiers sources en formats intermédiaires successifs, puis au codage de l'automate obtenu dans le langage de programmation hôte (langage C, ADA, LISP, etc). L'automate obtenu implémente la partie réactive de l'application développée et peut alors être intégré au reste de cette application.

La possibilité de compiler les programmes ESTEREL en automates est un atout majeur de ce langage car les automates sont des objets mathématiques bien connus et sur lesquels des preuves de propriétés peuvent être menées.

Etant donné que ce langage a été développé par le CMA, nous possédons sa chaîne de compilation. Nous avons donc la possibilité de l'utiliser, la modifier, l'adapter et l'étendre comme nous le souhaitons.

Example Soit, le petit programme suivant qu'on notera *prog* par la suite.

```
Module M:  
  abort  
    await Cap;  
    emit OK;  
    when ZoneR  
end Module
```

Ce petit programme est un exemple d'un certain type de comportement de bateau dans le port. On veut s'intéresser au comportement du bateau chaque fois qu'il change brutalement de cap dans n'importe quelle zone du port, et ceci en lançant une alarme. En revanche, si le bateau est dans une zone restreinte, on ne s'intéresse plus à son changement de cap.

Dans *prog*, nous avons exprimé le changement brutal de *Cap* par la variable *Cap*, le déclenchement d'une alarme par la variable *Ok* et la présence dans une zone restreinte par la variable *ZoneR*.

On considérera par la suite que ce programme ESTEREL constitue le module *M*. En s'exécutant, à l'instant suivant, ce module *M* se transformera par les règles de sémantiques en un module ESTEREL, on notera ce module par M_1 (voir section (3.2.2.4)). On indiquera la fin de l'exécution du programme par *term*.

La sémantique intuitive de ce programme Au premier instant, même si *Cap* et *ZoneR* sont présents, le programme se bloque.

A partir du deuxième instant, deux cas sont possibles :

- Si *ZoneR* est présent, le programme termine dans l'instant sans transmettre *Ok* même si *Cap* est présent.
- Si *ZoneR* est absent, on a aussi deux cas possibles :
 - Si *Cap* est présent, le programme termine en émettant *Ok*.
 - Si *Cap* est absent, le programme ne termine pas.

Ce comportement évolue de la même manière dans le temps.

Premier instant :

$$\begin{array}{c}
\langle \mathbf{await} \text{ } Cap, \rho \rangle \xrightarrow[\phi]{\phi, f, \phi} \langle \mathbf{present} \text{ } Cap \text{ else } \mathbf{await} \text{ } Cap \text{ end}, \rho \rangle \\
\hline
\langle \mathbf{await} \text{ } Cap; \mathbf{emit} \text{ } \mathcal{OK}, \rho \rangle \xrightarrow[\phi]{\phi, f, \phi} \langle \mathbf{present} \text{ } Cap \text{ else } \mathbf{await} \text{ } Cap \text{ end}; \mathbf{emit} \text{ } Ok, \rho \rangle \\
\hline
\langle \mathbf{abort} \text{ } prog \text{ when } ZoneR, \rho \rangle \xrightarrow[\phi]{\phi, f, \phi} \langle \mathbf{present} \text{ } ZoneR \text{ else } \mathbf{abort} \text{ } prog' \text{ when } ZoneR, \rho \rangle
\end{array}$$

avec $prog' = \mathbf{present} \text{ } Cap \text{ else } \mathbf{await} \text{ } Cap \text{ end}; \mathbf{emit} \text{ } Ok$

Ce qui correspond à :

$$M \xrightarrow[\phi]{\phi} M_1 \quad (3.83)$$

Deuxième instant :

On part du module M_1 , au prochain instant :

Si $ZoneR$ est présent : $ZoneR^+ \in \{ZoneR^+\}$ et $E = \{ZoneR^+\}$, Quelque soit la valeur du signal $Cap : Cap \in \{Cap^+, Cap^-\}$:

$$\langle \mathbf{present} \text{ } ZoneR \text{ else } \mathbf{abort} \text{ } prog' \text{ when } ZoneR, \rho \rangle \xrightarrow[\{ZoneR^+\}]{\phi, t, \phi} \langle \mathbf{nothing}, \rho \rangle$$

Ceci correspond à l'écriture :

$$M_1 \xrightarrow[\{ZoneR^+\}]{\phi, t, \phi} term \quad (3.84)$$

Si $ZoneR$ est absent et Cap est présent : $E = \{ZoneR^-, Cap^+\}$

$$\begin{array}{c}
\text{Cap} \in \{\text{Cap}^+\} \langle \text{present Cap else await Cap}, \rho \rangle \xrightarrow[\{\text{ZoneR}^-, \text{Cap}^+\}]{\phi, t, \phi} \langle \text{nothing}, \rho \rangle \langle \text{emit Ok}, \rho \rangle \xrightarrow[\{\text{ZoneR}^-, \text{Cap}^+\}]{\{\text{Ok}\}, t, \phi} \langle \text{nothing}, \rho \rangle \\
\hline
\langle \text{present Cap else await Cap; emit Ok}, \rho \rangle \xrightarrow[\{\text{ZoneR}^-, \text{Cap}^+\}]{\{\text{Ok}\}, t, \phi} \langle \text{nothing}, \rho \rangle \\
\hline
\text{ZoneR}^- \in \{\text{ZoneR}^-\} \langle \text{abort prog' when}, \rho \rangle \xrightarrow[\{\text{ZoneR}^-, \text{Cap}^+\}]{\{\text{Ok}\}, t, \phi} \langle \text{nothing}, \rho \rangle \\
\hline
\langle \text{present ZoneR else abort prog' when ZoneR}, \rho \rangle \xrightarrow[\{\text{ZoneR}^-, \text{Cap}^+\}]{\{\text{Ok}\}, t, \phi} \langle \text{nothing}, \rho \rangle
\end{array}$$

Ce qui correspond à :

$$M_1 \xrightarrow[\{\text{Cap}^+, \text{ZoneR}^-\}]{\{\text{Ok}\}, t, \phi} \text{term} \quad (3.85)$$

sinon Cap et ZoneR peuvent être absents tous les deux, c'est-à-dire $E = \{\text{Cap}^-, \text{ZoneR}^-\}$ dans ce cas, on retrouve :

$$\langle \text{present ZoneR else abort prog' when ZoneR}, \rho \rangle \xrightarrow[\{\text{Cap}^-, \text{ZoneR}^-\}]{\phi, f, \phi} \langle \text{present ZoneR else abort prog' when ZoneR}, \rho \rangle$$

Ce qui correspond à :

$$M_1 \xrightarrow[\{\text{Cap}^-, \text{ZoneR}^-\}]{\phi, f, \phi} M_1 \quad (3.86)$$

Compilation en automate à états finis La sémantique comportementale de tout programme permet de construire l'automate à états finis de celui-ci. Les équations (3.83), (3.84), (3.85) et (3.86) représentent exactement les transitions et les états de l'automate à états finis de la figure (3.9) associé à Prog .

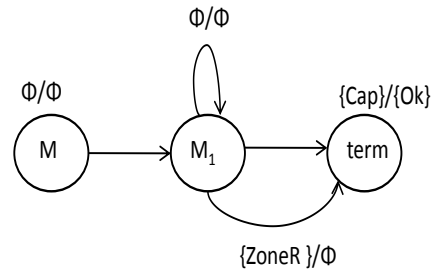


FIGURE 3.9 – L'automate à états fini associé au module M

Conclusion

Dans cette section nous avons introduit l'approche réactive synchrone qui nous a permis de concevoir le sous-module de reconnaissance de comportement par approche réactive (BRAM) du système global de reconnaissance de comportement de navire (le BAM). Nous rappelons que le BRAM a été principalement conçu pour reconnaître les comportements des petites embarcations.

La conception détaillée par approche réactive synchrone de ce sous-module sera traitée dans la section 4.2 du chapitre 4.

Chapitre 4

Application à la reconnaissance de comportements de navires

Ce travail de thèse est une contribution qui s'inscrit dans le cadre du projet SECMAR que nous avons introduits en section 1.5. Notre contribution a consisté à développer un système autonome permettant d'apporter une assistance à l'équipe de surveillance du port concernant le caractère menaçant ou nominal du comportement de bateaux. Nous avons ainsi conçu un module d'analyse de comportement BAM qui a été ensuite implanté dans le système de surveillance portuaire temps-réel multi-capteurs développé par l'ensemble des partenaires industriels et académiques de ce projet SECMAR sous la maîtrise d'oeuvre de TUS. Le module BAM prend en entrée l'ensemble des informations, décrivant les bateaux présents dans la zone sous surveillance à un instant donné, obtenues à la suite du processus de fusion décrit dans la section 1.5.3. Au cours du temps, les informations concernant un même bateau sont reliées dans ce que nous avons appelé dans la section 1.5.3 une *piste système*. Nous rappelons qu'une *piste système* est une suite temporelle d'informations datées donnant la position en longitude et en latitude du bateau, sa vitesse instantanée, son cap, éventuellement son type, sa taille et son chargement lorsqu'il s'agit d'un navire marchand ; notons que le temps qui s'écoule est calculé à partir des dates contenues dans les pistes systèmes.

Le BAM (voir figure (4.1)) est formé de deux sous-modules :

- Le BPAM ou *Behavioral Probabilistic Analysis Modulus* est basé sur l'approche probabiliste MMC. Il prend en charge la reconnaissance et la classification des comportements des bateaux qui doivent suivre des chenaux d'accès imposés par la législation. Il s'agit principalement des gros bateaux marchands dont la cargaison peut être dangereuse comme le cas des méthaniers, des pétroliers et

des chimiques ou non dangereuse comme les porte-conteneurs. Ce module a été construit en trois phases successives : une phase d'*initialisation* des Modèles de Markov Cachés associés aux comportements des bateaux dans le port, une phase de *réestimation* des paramètres des MMC et la phase à proprement dit de *reconnaissance* des comportements de bateaux.

- Le BRAM ou *Behavioral Analysis Reactive Modulus* est basé sur l'approche réactive. Ce module décrit un ensemble de comportements prédéfinis de bateaux par des programmes ESTEREL. Notons que l'ensemble des comportements de bateaux qu'on a appelé *scénarios* a été établi en collaboration avec des experts de la sûreté portuaire du GPMM. Ainsi, chaque scénario pour ce module est un système réactif synchrone décrit par un programme ESTEREL. Ce module est particulièrement pertinent pour la reconnaissance de comportements transgressifs (excès de vitesse, entrée sans autorisation dans une zone à accès interdit, etc.) ainsi que pour les comportements dangereux des petits bateaux rapides (a changé de cap brutalement avant de faire cap vers un point sensible, etc). Enfin, très peu de réglementations étant imposées par la législation à ces bateaux, ils sont libres de circuler dans le port, ce qui rend leurs comportements difficile à modéliser par des modèles probabilistes. Notons qu' ESTEREL permet aussi de spécifier des scénarios de comportements nominaux.

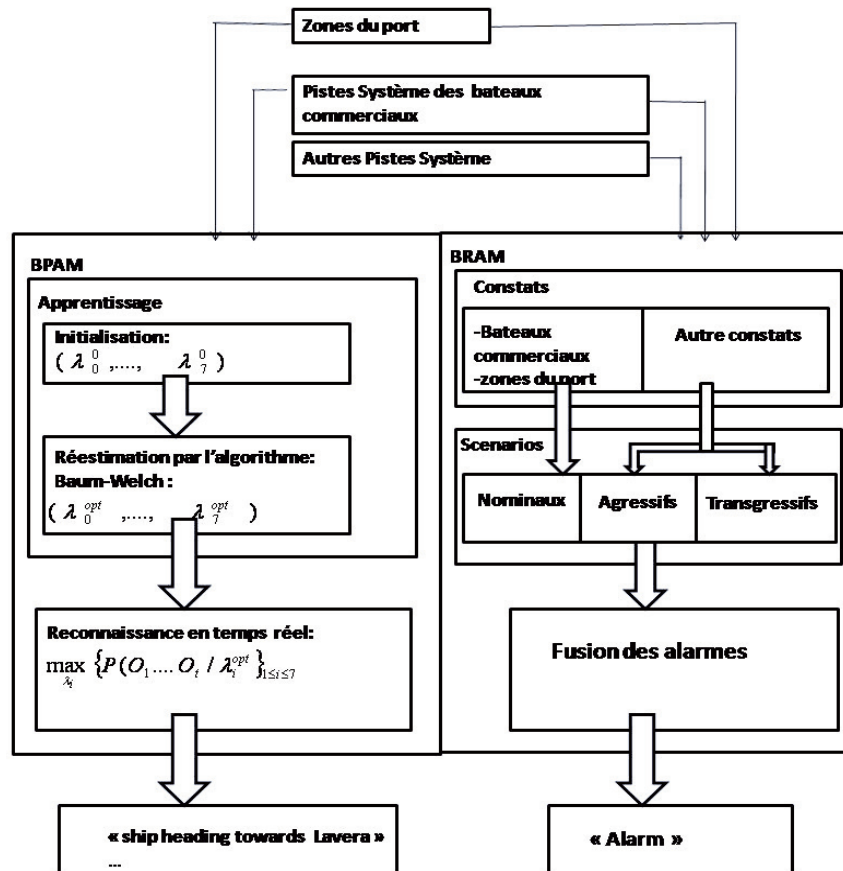


FIGURE 4.1 – Behavior Analysis Modulus.

4.1 Reconnaissance de comportements de bateaux par MMC

4.1.1 Fonctionnement général du module BPAM

Le BPAM est uniquement dédié aux gros bateaux qui se déplacent dans leurs chenaux d'accès conformément aux règles de la navigation en suivant des comportements réguliers et récurrents. Grâce à ces informations régulières, il est possible d'apprendre les comportements de navires. Pour ceci, nous avons principalement utilisé les informations sur les positions en longitude et en latitude (initialement en coordonnées géographiques et converties en coordonnées cartésiennes), la vitesse en

m/s et le temps en seconde.

Pour concevoir ce module, nous avons tout d'abord commencé par définir le comportement d'un navire dans un port. Nous avons alors décidé de définir le comportement d'un navire comme étant « *le déplacement de celui-ci dans des zones fonctionnelles successives définies par la législation afin d'atteindre la zone finale qui lui est destinée* ». Pour une traduction mathématique par MMC de cette définition, nous avons considéré que ceci correspond à l'évolution de deux processus stochastiques sous-jacents. Le premier processus stochastique nous renseigne sur le *déplacement* du bateau dans le port à travers les pistes systèmes et plus particulièrement les positions. Ce processus est sous-jacent à un deuxième qui, lui, va représenter l'intention du commandant du bateau de se diriger vers *une zone fonctionnelle*. La position du bateau représente l'observation et l'intention du bateau de se diriger vers une zone constitue l'état caché. Cet état est inféré à partir de la position du bateau. Un comportement de bateau est alors un MMC où les états représentent les intentions du commandant du bateau et les observations sont celles des longitudes et latitudes des bateaux au cours de leurs déplacements.

Après avoir établi la liste des MMC associés à tous les comportements possibles des gros bateaux dans le port, le comportement de tout nouveau bateau entrant dans le GPMM sera évalué en temps-réel.

En somme, le BPAM que nous avons conçu et que nous avons illustré par la figure (4.2) est principalement basé sur deux phases :

- Une phase de construction des modèles au cours de laquelle les modèles associés à tous les comportements de bateau évoluant dans le port seront initialisés (étape (b)) puis réestimés grâce aux pistes systèmes (étape (c)).
- Une phase de reconnaissance (étape (d)) où les modèles préalablement construits au cours de la phase (a) et (b) seront utilisés afin de reconnaître en temps-réel le comportement de tout nouveau bateau entrant dans le port.

Le BPAM prendra alors en entrée l'ensemble des zones du port définies par la législation ainsi que l'ensemble des pistes systèmes sur les gros bateaux (étape (a)). En sortie, ce module fournira le comportement du bateau le plus probable (étape (e)).

Dans la section suivante, nous détaillerons les étapes (b), (c) et (d). C'est-à-dire les phases d'initialisation, de réestimation et de reconnaissance des modèles de comportements. Puis, nous expliquerons à partir d'exemples concrets comment nous avons pu reconnaître les comportements de bateaux à partir des modèles construits.

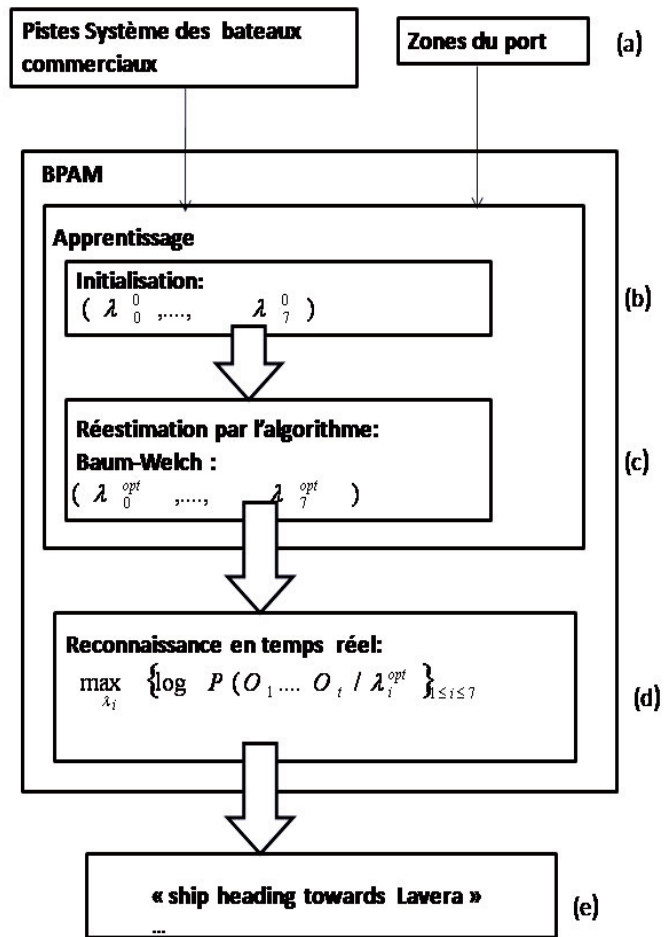


FIGURE 4.2 – Module de reconnaissance de comportements par approche probabiliste : le BPAM

4.1.2 Construction des modèles de comportements

4.1.2.1 Initialisation des modèles de comportements

Rappelons qu'un MMC λ est formé de trois paramètres principaux : le vecteur d'initialisation π , la matrice de transition d'état A et les densités de probabilités connaissant les états B . Nous allons introduire nos connaissances préliminaires afin

d'établir la matrice de transition d'état et les densités de probabilités des observations connaissant les états de nos modèles de comportements.

Nous disposons d'une suite temporelle de pistes systèmes de gros bateaux ayant des propriétés qui évoluent dans le temps. Ces pistes systèmes constituent les observations de nos modèles de MMC associés aux comportements de bateaux dans le port. Elles ont été enregistrées au cours du déplacement de chaque bateau dans les différentes zones du port de manière à ce qu'il traverse successivement les zones du port sans revenir en arrière ni sauter d'une zone à une autre. Autrement dit, le bateau va, au cours de son évolution, soit rester dans sa zone soit se diriger vers la zone suivante. Pour cette raison, il est clair qu'un modèle de Bakis décrit dans 3.1.2.2 permet de mieux intégrer ce type d'information dans les modèles de comportements.

Nous avons alors initialisé les paramètres des MMC des comportements de navires de la manière suivante :

La matrice de transition d'état Dans notre modèle de comportement, la variable aléatoire X_t qui représente l'état qu'on notera S_i est la succession des intentions de passer d'une zone à une autre dans le port. Le terme de transition d'état $P(X_t = S_j / X_{t-1} = S_i)$ désigne alors la probabilité d'avoir l'intention de transiter d'une zone à une autre.

Comme nous l'avons expliqué dans la section 3.1.2.2, un modèle de Bakis décrit au mieux notre modèle de comportement de bateaux. Il a alors été utilisé pour modéliser l'intention de passer d'une zone à une autre :

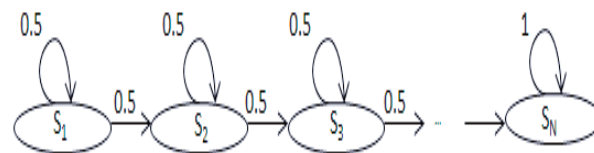


FIGURE 4.3 – Graphe de transition d'états modélisant l'intention de passer d'une zone à une autre

Au cours de son évolution dans le port, nous avons considéré que le bateau a 50% de chance de rester dans sa zone et 50% de chance d'aller dans la zone suivante. Ceci est traduit par le diagramme d'état de la figure (4.3). D'où la forme suivante de la matrice de transition d'état :

$$\begin{aligned} \{a_{ij}\}_{1 \leq i, j \leq N} &= 0.5 \quad \text{si } i = j \text{ ou } j = i + 1 \\ a_{NN} &= 1 \end{aligned} \quad (4.1)$$

où N qui est la dimension de l'espace d'état représente le nombre de zones par lesquelles passera chaque bateau. La valeur de N dépendra du comportement modélisé.

Le vecteur d'initialisation Etant donné qu'il s'agit d'un modèle de Bakis, l'initialisation de π est la suivante :

$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases}, \quad 1 \leq i \leq N \quad (4.2)$$

Ce qui illustre le fait qu'à l'instant initial, tous les bateaux se trouvent obligatoirement dans la zone A_1 (voir figure (4.4a)).

Les densités de probabilité des observations connaissant les états Les observations prises pour notre modèle de comportement sont celles des positions, longitude et latitude du bateau, au cours de son déplacement dans le port. Nous noterons cette observation par O_t et $O_t \in R^2$.

Nous avons considéré que la loi qui relie l'observation à l'état $\{b_i(O_t)\}_{1 \leq i \leq N}$ est celle d'une loi multinormale (4.3)(voir section 3.1.2.1)

$$b_i(O_t) = \frac{1}{(2\pi)^{\frac{k}{2}} \det(\Sigma_i)^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (O_t - \mu_i) \Sigma_i^{-1} (O_t - \mu_i)^T \right\}_{1 \leq i \leq N} \quad (4.3)$$

dont les paramètres associés doivent également être initialisés, c'est-à-dire la moyenne μ_i et la covariance Σ_i .

Ces paramètres ont été initialisés à partir des différentes zones fonctionnelles délimitées par la législation (figure (4.4a)).

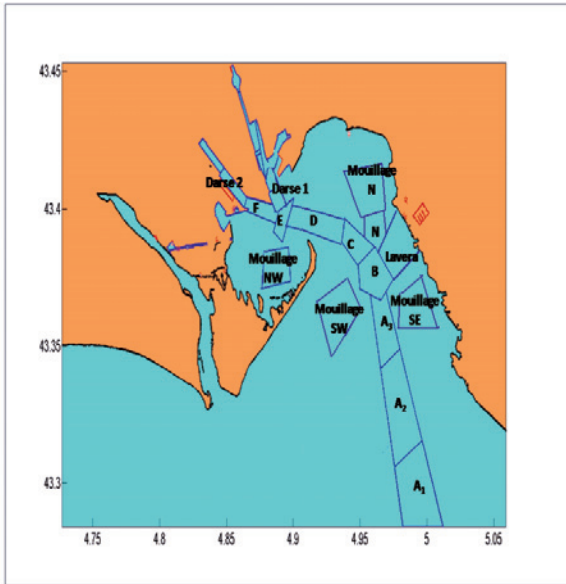
En effet, chaque zone de la figure (4.4a) est estimée par une ellipse de la figure (4.4b) qui contient les paramètres (la moyenne μ_i et la covariance Σ_i) de chaque densité de probabilités $\{b_i(O_t)\}_{1 \leq i \leq N}$.

La moyenne μ_i de la gaussienne (4.3) représente le centre de l'ellipse et nous avons choisi de calculer la covariance Σ_i à partir des valeurs du grand axe a_i , du petit axe b_i et l'angle θ_i que fait l'ellipse avec le repère orthonormé. La relation entre la covariance et ces paramètres est donnée par la formule :

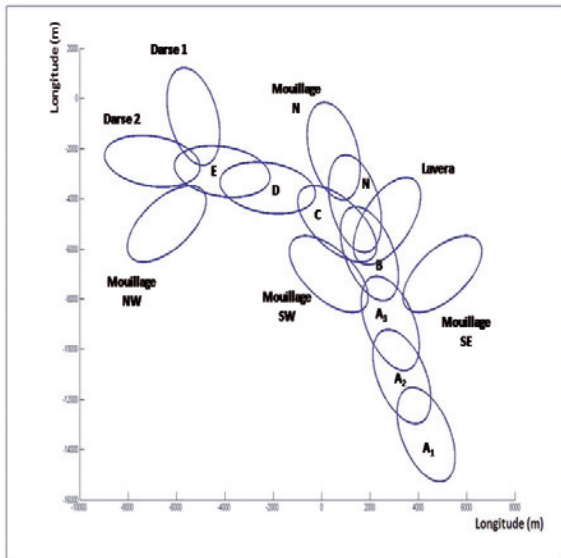
$$\Sigma_i = U_i A_i U_i^T \quad (4.4)$$

$$\text{où } U_i = \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{pmatrix} \text{ et } A_i = \begin{pmatrix} a_i^2 & 0 \\ 0 & b_i^2 \end{pmatrix}$$

Le meilleur comportement *nominal* est alors celui qui passe par les centres de ces ellipses puisque c'est là où la probabilité $b_i(O_t)$ est maximale.



(a) Les zones fonctionnelles du port délimitées par la législation



(b) Les ellipses estimant les zones du port

FIGURE 4.4 – Représentation de chaque zone fonctionnelle par une ellipse

Considérons le comportement du bateau qui se dirige vers Lavera (voir figure (4.5)). Le bateau doit prendre le chenal d'accès qui mène à Lavera tout en passant par les zones A_1 , A_2 , A_3 et B .

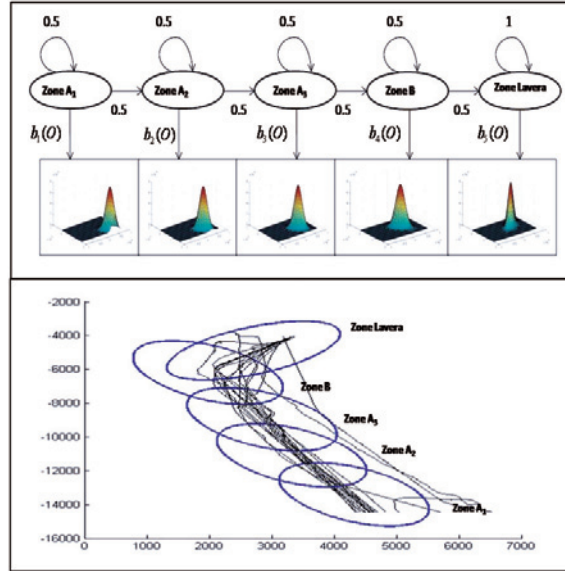


FIGURE 4.5 – Représentation des paramètres du modèle initial de MMC associé à un comportement de bateaux se dirigeant vers Lavera.

La dimension de l'espace d'état N qui correspond au nombre de zones est alors égale à 5.

Nous avons représenté sur la figure (4.5) le diagramme de transitions entre les états qui est celui d'un modèle de Bakis vérifiant les propriétés (4.1). A chaque état est associé une densité de probabilité $b_i(O_t)$ qui est approximée par une loi gaussienne de moyenne μ_i représentant le centre de cette ellipse et de covariance Σ_i obtenue à partir du grand axe et du petit axe de la même ellipse. Par exemple $b_1(O_t)$ est une densité de probabilité associée au premier état : « intention d'être dans la zone A_1 » et ses paramètres sont la moyenne μ_1 et la covariance Σ_1 . La moyenne μ_1 est celle du centre de la première ellipse de la figure (4.5) et la covariance Σ_1 s'obtient à partir du grand axe et du petit axe de la même ellipse qui s'écrit suivant l'équation (4.4).

A partir des chenaux d'accès au port définis par la législation que nous avons représentés dans la figure (4.4a), nous avons établi la liste de tous les comportements nominaux possibles dans le GPMM comme suit :

1. *Bateau dans son chenal d'accès se dirigeant vers la zone de mouillage Sud Est.*
2. *Bateau dans son chenal d'accès se dirigeant vers la zone de mouillage Sud Ouest.*
3. *Bateau dans son chenal d'accès se dirigeant vers Lavera.*
4. *Bateau dans son chenal d'accès se dirigeant vers la zone de mouillage Nord.*
5. *Bateau dans son chenal d'accès se dirigeant vers la zone de mouillage Nord Ouest.*
6. *Bateau dans son chenal d'accès se dirigeant vers Darse 1.*
7. *Bateau dans son chenal d'accès se dirigeant vers Darse 2.*

Nous avons alors initialisé ces 7 comportements selon le principe précédemment décrit. Notons que le nombre d'états pour chaque modèle de comportements varie selon le nombre de zones définies pour chaque comportement. C'est-à-dire que pour le premier comportement de la liste, pour aller à la zone de mouillage Sud Est, chaque bateau doit traverser 4 zones. Le nombre d'états est alors égal à 4. D'autre part, pour aller vers Lavera, 5 zones sont traversées par chaque bateau. Le nombre d'états est alors égal à 5.

Finalement, 7 modèles de comportements initiaux d'un bateau dans le GPMM ont été établis. On notera le MMC associé à chaque type de comportement par $\{\lambda_i^{(0)}\}_{1 \leq i \leq 7}$. **Dans tout ce qui suit, l'indice i des modèles de comportements de bateaux λ_i est exactement celui du numéro du comportement dans la liste précédente** (par exemple λ_1 est associé au comportement 1 : *Bateau dans son chenal d'accès se dirigeant vers la zone de mouillage Sud Est*).

4.1.2.2 Réestimation des modèles de comportements de bateaux dans le port

Nous avons décrit précédemment l'initialisation des paramètres d'un MMC $\lambda = (\pi, A, B)$ associé à chaque type de comportement de bateau.

Sachant que la vitesse du bateau au cours de son déplacement dans le port n'est pas uniforme et afin de préserver le principe de stationnarité des MMC, nous avons pris comme référence temporelle la notion de distance parcourue. La distance parcourue s'exprime en fonction de la vitesse instantanée et le temps comme suit : $\Delta S = V \Delta T$ où V est la vitesse instantanée en m/s et T est le temps écoulé entre deux pistes système en seconde.

Les valeurs de la vitesse V et du temps écoulé ΔT sont aussi données par les pistes systèmes. De cette manière, a_{ij} peut s'écrire :

$$a_{ij} = P(X_{S+\Delta S} = j / X_S = i) \quad (4.5)$$

Après avoir initialisé les comportements de bateaux, on a obtenu la base $\Lambda = (\lambda_1^{(0)}, \dots, \lambda_7^{(0)})$ des MMC associés aux comportements des gros bateaux qui ont des comportements réguliers dans le port. Les paramètres π , A et B de ces modèles ont été réestimés en utilisant les positions (longitude et latitude des bateaux en mètre) afin d'obtenir des modèles plus précis et plus proche du modèle réel de comportement. Il s'agit alors de l'*apprentissage* des modèles.

Puisqu'il s'agit d'un modèle de Bakis, la réestimation des paramètres selon l'algorithme Baum-Welch a été effectuée en utilisant des séquences d'observation multiples (principe du paragraphe 3.1.2.4). Les positions des bateaux commerciaux que nous avons choisies comme observation ont été échantillonnées chaque fois que le bateau parcourt $\Delta S = 1000m$ (rappelons que le but est de préserver le principe de stationnarité des MMC). Notons que le nombre d'observations et de séquences d'observations pris pour l'apprentissage était différent d'un comportement à un autre.

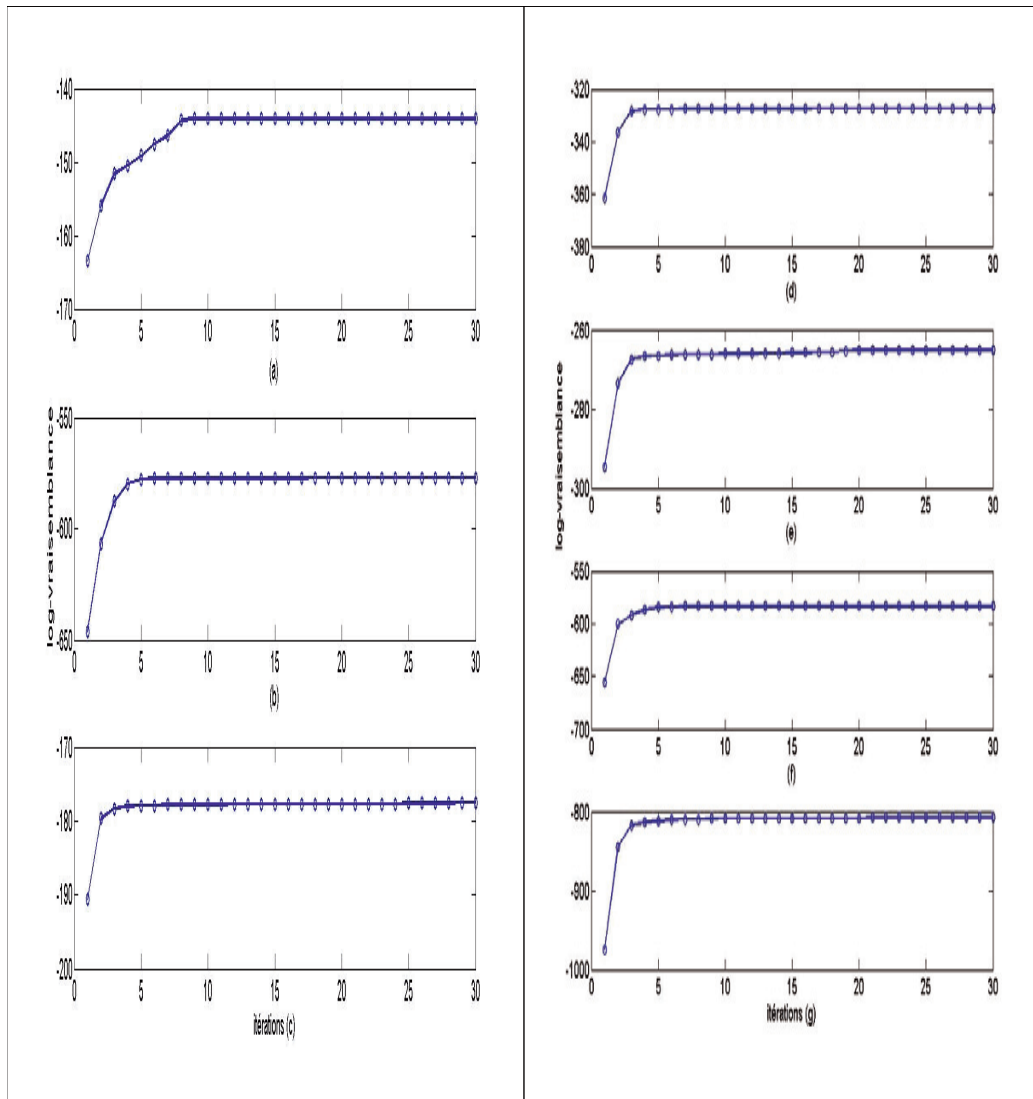


FIGURE 4.6 – Les log-vraisemblances $\log \{P(O^i/\lambda_i)\}_{1 \leq i \leq 7}$ calculées respectivement pour les courbes (a), (b), (c), (d), (e), (f) et (g) en utilisant l’algorithme Baum-Welch.

Suite à l’apprentissage, nous avons représenté sur la figure (4.6), les valeurs des log-vraisemblances $\log(P(O^i/\lambda_i))$ obtenues pour chaque modèle de comportement λ_i pour $i \in [1, 7]$. Ces valeurs ont été obtenues suite à l’application de l’algorithme

Baum-Welch et en utilisant les pistes système afin de mieux optimiser les modèles initiaux de comportements. Ce processus de réestimation permet de réduire l'incertitude entre le modèle réel et le modèle représenté par MMC. Un modèle plus précis permet d'améliorer les performances de la reconnaissance.

On constate qu'au bout de 5 itérations (10 itérations pour la figure (4.6)-(a)), l'algorithme Baum-Welch converge vers un modèle optimal : λ_{opt} .

À la suite de cette estimation, les propriétés (3.11), (3.12) et (3.13) du modèle de Bakis seront toujours conservées.

Donc, les valeurs des vecteurs d'initialisation restent les mêmes après la réestimation : le bateau part toujours de la première zone au cours de son déplacement. Néanmoins, les valeurs des matrices de transition d'état ne sont plus équiprobables. De même, les paramètres des gaussiennes (μ_i, Σ_i) ont aussi changé et par conséquent la représentation des zones par des ellipses a elle aussi été réévaluée (voir figure (4.7))

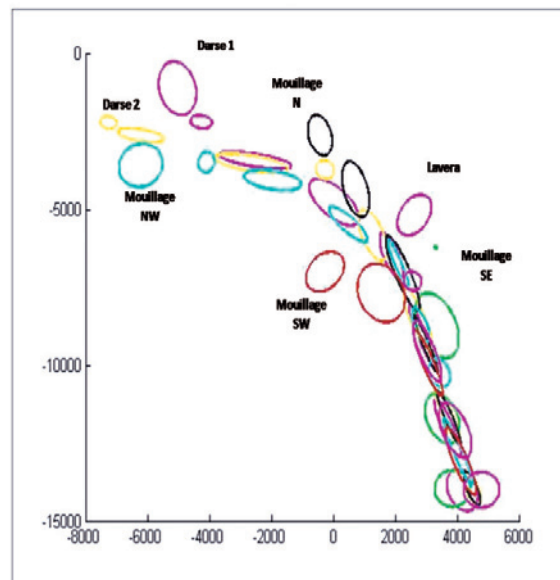


FIGURE 4.7 – Les ellipses associées aux différentes zones du port obtenue suite à leurs réestimation par l'algorithme Baum-Welch

Si on reprend l'exemple de la figure (4.5), la réestimation de ce comportement par l'algorithme Baum-Welch donne le modèle de la figure (4.8) :

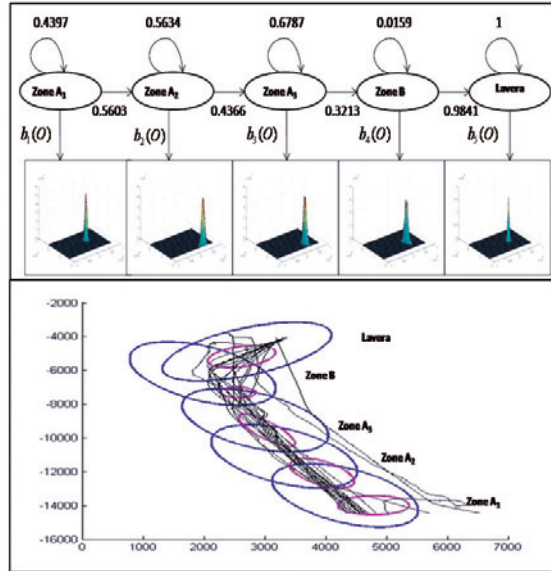


FIGURE 4.8 – Représentation des paramètres du modèle initial du comportement de bateaux qui se dirigent vers Lavera après apprentissage.

Nous avons alors représenté dans la figure (4.8) le nouveau diagramme d'état ainsi que les nouvelles zones du port obtenues après réestimation des modèles de comportements par l'algorithme Baum-Welch.

4.1.3 Reconnaissance de comportements de bateaux

Après avoir construit la base Λ de tous les comportements $\{\lambda_i\}_{1 \leq i \leq 7}$ possibles dans le port, le comportement de chaque nouveau bateau entrant dans le port sera évalué chaque fois que ce dernier parcourt une distance de $t = \Delta S$. Ce calcul sera fait par une comparaison à chaque pas temporel des valeurs du log-vraisemblance par rapport à tous les MMC.

L'idée est d'enregistrer les différentes positions du bateau en longitude et en latitude chaque fois que celui-ci parcourt exactement ΔS . Cette observation sera utilisée pour le calcul des vraisemblances (équation (4.7)) pour obtenir une estimation du comportement le plus probable à cet instant.

Le comportement le plus probable est celui qui a donné la probabilité la plus élevée :

$$\max \{\log P(O_1 \dots O_t / \lambda_i)\}_{1 \leq i \leq 7} \quad (4.6)$$

Le calcul de cette probabilité est basée sur les termes du forward de la section 3.1.2.4 :

$$P(O_1 \dots O_t / \lambda) = \sum_{i=1}^N \alpha_t(i) \quad (4.7)$$

Nous rappelons que $\alpha_t(i) = P(O_1 \dots O_t, X_t = S_i / \lambda)$ est la variable du forward. Cette valeur de $\alpha_t(i)$ dépend, à chaque pas temporel ΔS , à la fois de l'observation courante, de la valeur de l'état courant ainsi que de toutes les observations passées. A chaque instant, le processus de reconnaissance va alors inférer le comportement le plus approprié à partir de l'observation courante et des anciennes observations.

Afin de mieux analyser les résultats de reconnaissance, nous allons expliquer dans la section 4.1.4 trois types d'exemples :

4.1.4 Exemples de comportements de navires qui se déplacent dans le port

Exemple 1 : reconnaissance d'un comportement nominal de bateau se dirigeant vers Darse1 Dans l'exemple de la figure (4.9), il s'agit d'un comportement de bateau préalablement connu : celui d'un bateau se dirigeant vers Darse 1 représenté par λ_6 . Nous allons décrire nos résultats à travers les figures (4.9), (4.10) et le tableau (4.1).

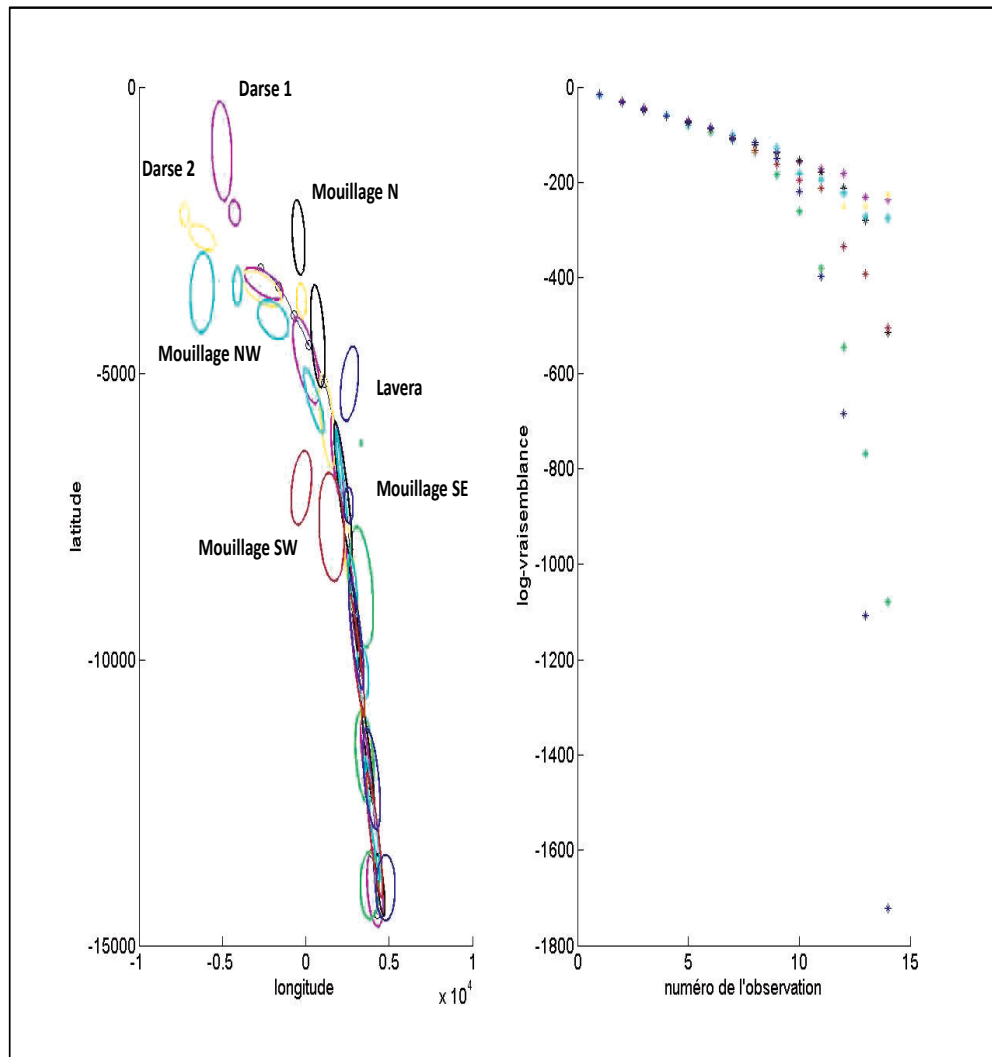


FIGURE 4.9 – Les valeurs du log-vraisemblance $\{P(O_1 \dots O_t / \lambda_i)\}_{1 \leq i \leq 7}$ calculées pour chaque pas de distance parcourue de 1000m - Le bateau se dirige vers Darse 1.

$t = \Delta S$	1	2	3	4	5	6	7
$P(O/\lambda_1)$	-15.15	-31.48	-47,11	-61,75	-81,59	-95,78	-112,70
$P(O/\lambda_2)$	-15.52	-30.15	-45,04	-60,01	-71,51	-86,26	-107,18
$P(O/\lambda_3)$	-15.25	-31.30	-46,03	-62,23	-74,44	-87,47	-108,08
$P(O/\lambda_4)$	-15.96	-31.71	-48,05	-57,71	-74,36	-85,63	-106,60
$P(O/\lambda_5)$	-18.28	-32.48	-45,02	-59,95	-75,32	-85,57	-100,76
$P(O/\lambda_6)$	-14.78	-30.87	-44.15	-59.08	-74.72	-84.47	-102.36
$P(O/\lambda_7)$	-15.34	-31.21	-46.39	-57.85	-74.88	-91.72	-103.8

8	9	10	11	12	13	14
-138,58	-184,32	-259.96	-380.54	-546.17	-769.69	-1078.98
-132,43	-160,89	-195.06	-213.27	-335.35	-392.05	-504.1
-117,49	-149,88	-219.51	-397.02	-684.05	-1106.99	-1722.52
-120,54	-137,15	-155.6	-177.96	-211.63	-278.46	-514.92
-116,31	-129,16	-180.52	-196.57	-221.13	-273.17	-275.39
-122.96	-136.19	-156.56	-172.24	-181.38	-230.54	-235.8
-123.82	-143.58	-155.22	-183.79	-251.46	-250.67	-225.99

TABLE 4.1 – L'évolution des valeurs du log-vraisemblance de la séquence d'observations $P(O_1 \dots O_t / \lambda_i)_{1 \leq i \leq 7, 1 \leq t \leq 14}$ par rapport à tous les modèles de comportements pour une séquence d'observation d'un comportement de bateau se dirigeant vers Darse 1.

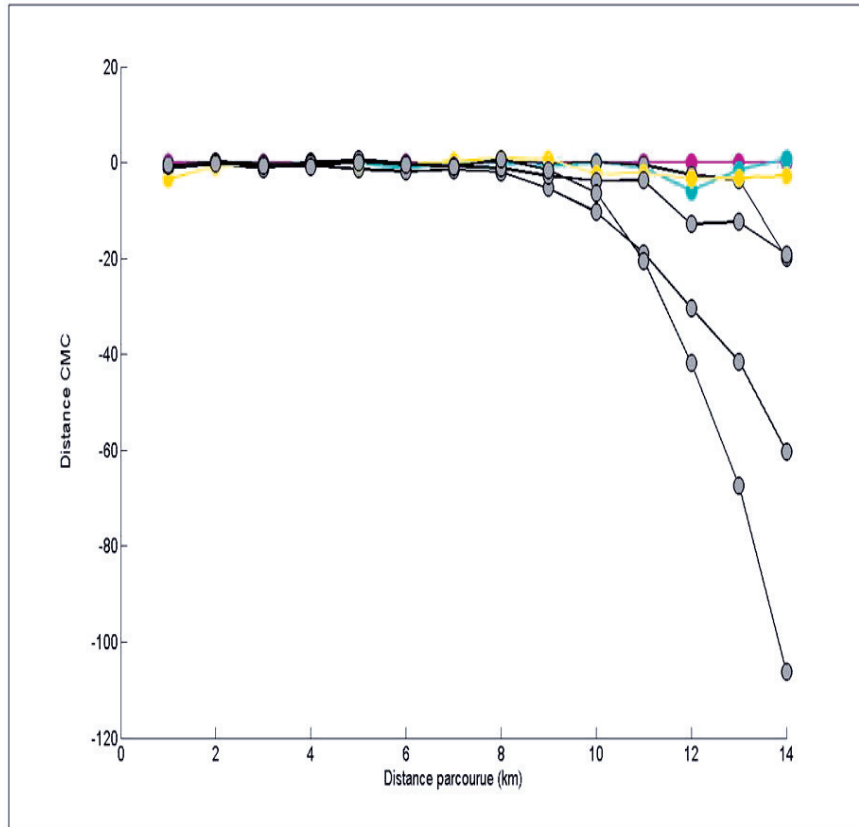


FIGURE 4.10 – Evolution de la distance $D(\lambda_i, \lambda_6)$. Après avoir parcouru 11km, D est proche de 0 pour les comportements de bateaux se dirigeant vers Darse 2 et Mouillage Nord Ouest (représentée respectivement en jaune et bleue pour $D(\lambda_5, \lambda_6)$ et $D(\lambda_5, \lambda_7)$). La valeur de D est petite pour le reste des modèles de comportements (pour $i \neq \{5, 6, 7\}$).

Dans la figure (4.9), nous avons représenté toutes les zones estimées du port par l'algorithme Baum-Welch, la trajectoire préalablement connue d'un bateau se dirigeant vers Darse 1 ainsi que l'évolution des différentes valeurs du log-vraisemblance au cours du passage du bateau par les différentes zones du port. Ces valeurs se trouvent entièrement dans le tableau 4.1. Afin d'étudier et comparer ces valeurs nous avons utilisé la distance entre deux MMC définie dans [110] par :

$$D(\lambda_1, \lambda_2) = \frac{1}{T} (\log (P(O^{(2)}/\lambda_1)) - \log (P(O^{(2)}/\lambda_2)))$$

avec λ_1 et λ_2 deux modèles quelconques de MMC, $O^{(2)} = (O_1, \dots, O_T)$ la séquence d'observation obtenue par λ_2 et T est sa longueur. Cette distance permet d'évaluer le degré de correspondance de la séquence $O^{(2)}$ au modèle λ_1 et au modèle λ_2 dont elle est issue. En se basant sur cette définition, nous allons calculer à chaque pas temporel la distance D définie par :

$$D(\lambda_i, \lambda_c) = \frac{1}{t} (\log (P(O^{(c)}/\lambda_i)) - \log (P(O^{(c)}/\lambda_c))) \quad (4.8)$$

où $\{\lambda_i\}_{1 \leq i \leq 7, i \neq c}$ est l'ensemble de tous les comportements et $O^{(c)}$ est la séquence d'observation issue du modèle λ_c et t est le pas temporel courant. Cette distance nous fournira trois résultats possibles : $D(\lambda_i, \lambda_c) = 0$ pour des comportements similaires, $D(\lambda_i, \lambda_c) < 0$ pour un modèle λ_c qui correspond mieux que λ_i à la séquence d'observation $O^{(c)}$ et inversement pour $D(\lambda_i, \lambda_c) > 0$.

Dans notre cas, on sait préalablement que le comportement en cours est celui d'un bateau se dirigeant vers Darse 1 ($c = 6$). Nous voulons évaluer comment évolue cette distance associée à une séquence d'observation issu d'un comportement préalablement connue. Les valeurs de la figure (4.10) ont été enregistrées pour un pas temporel $t = \Delta S = 1000$. Le calcul de la distance à travers cette figure illustre alors trois résultats intéressants :

1. Avant d'atteindre une distance parcourue de 8 km, la distance $D(\lambda_i, \lambda_6) \approx 0$ pour l'ensemble des modèles : les observations sont en adéquation avec tous les MMC. Ce résultat nous montre que quand le bateau passe par les zones communes (zone A_1 , zone A_2 , zone A_3 , zone B), le bateau a les mêmes chances d'effectuer tous les comportements de la liste définie par la législation. C'est-à-dire que le bateau peut se diriger vers n'importe quelle destination (Lavera, Dase1, Zone de Mouillage N...). Néanmoins, en regardant de plus près le tableau (4.1), il est possible d'avoir une approximation du comportement du bateau le plus probable et le moins probable à chaque pas temporel. Néanmoins, le niveau de pertinence de cette approximation est faible car les distances sont proches les unes des autres.

2. Après avoir parcouru 8 km et en dépassant la zone B, l'écart entre les distances D commence à augmenter. Cependant, $D(\lambda_i, \lambda_6)$, reste toujours proche de 0 pour $i = 5, 6$ et 7 alors que sa valeur est de plus en plus faible et tend vers $-\infty$ pour le reste des comportements. En plus du modèle λ_6 , les modèles λ_5 et λ_7 correspondent aussi à la séquence d'observation O plus que $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$. Par conséquent, le bateau est en train de se diriger vers Darse 1, Darse 2 ou la zone de Mouillage N.
3. L'écart D est plus significatif à partir de 11 km et le comportement définitif est donné selon notre modèle. Cette distance D reste toujours proche de 0 pour les comportements de bateau se dirigeant vers Darse 1, Darse 2 et Mouillage N alors qu'elle tend vers $-\infty$ pour le reste des modèles de comportement. Ceci permet de conclure définitivement que le bateau se dirige probablement vers l'une des zones : Darse1, Darse 2 ou la zone de mouillage N.

En conclusion, quand le bateau passe par les zones communes, le niveau de pertinence sur la prédiction du comportement du bateau est faible. En arrivant à la zone B et à partir des valeurs de la distance, on voit que le bateau se dirige vers Darse 1, Darse 2 et Mouillage Nord. Finalement, ce résultat se confirme car l'écart est considérable à la fin du processus de reconnaissance. Nos résultats de reconnaissance peuvent alors se résumer en trois phases :

- une phase d'attente avec de faibles estimations sur les comportements en cours.
- un phase transitoire de début de reconnaissance où certains comportements sont plus vraisemblables que d'autres.
- une phase de reconnaissance définitive du ou des comportements les plus probables.

Ce module de reconnaissance de comportements par approche probabiliste a été implémenté en MATLAB (voir figure (4.11)) de façon à afficher les informations sur les comportements des bateaux en cours pour chaque pas temporel (distance parcourue de 1000 m).

```
Command Window
bateau 1*****
distance parcourue 0:bateau se dirige vers Darse1
distance parcourue 1000:bateau se dirige vers MouillageSW
distance parcourue 2000:bateau se dirige vers Darse1
distance parcourue 3000:bateau se dirige vers MouillageN
distance parcourue 4000:bateau se dirige vers MouillageSW
distance parcourue 5000:bateau se dirige vers Darse1
distance parcourue 6000:bateau se dirige vers MouillageNW
distance parcourue 7000:bateau se dirige vers MouillageNW
distance parcourue 8000:bateau se dirige vers MouillageNW
distance parcourue 9000:bateau se dirige vers Darse2
distance parcourue 10000:bateau se dirige vers Darse1
distance parcourue 11000:bateau se dirige vers Darse1
distance parcourue 12000:bateau se dirige vers Darse1
distance parcourue 13000:bateau se dirige vers Darse2
*****
```

FIGURE 4.11 – Le résultat de la reconnaissance du comportement de la figure (4.9) obtenu par la fenêtre de commande en MATLAB.

La figure (4.11) nous montre qu'en décrivant mathématiquement le comportement d'un bateau évoluant dans le port, le module de reconnaissance de comportement se comporte exactement comme un être humain dans sa prise de décision : un être humain (le surveillant du port) peut observer le comportement du bateau de l'exemple 1 en donnant son avis approximatif sur le comportement en cours. Sa décision sera définitive quand le bateau va se diriger vers une zone bien déterminée. De la même manière, le module récupère les positions de bateaux (observations) puis estime leurs comportements les plus probables pour chaque pas temporel (algorithme forward). En reproduisant les expériences précédentes avec plusieurs types de comportements de bateau, le BPAM s'est avéré être une aide parfaite à la décision pour les comportements nominaux.

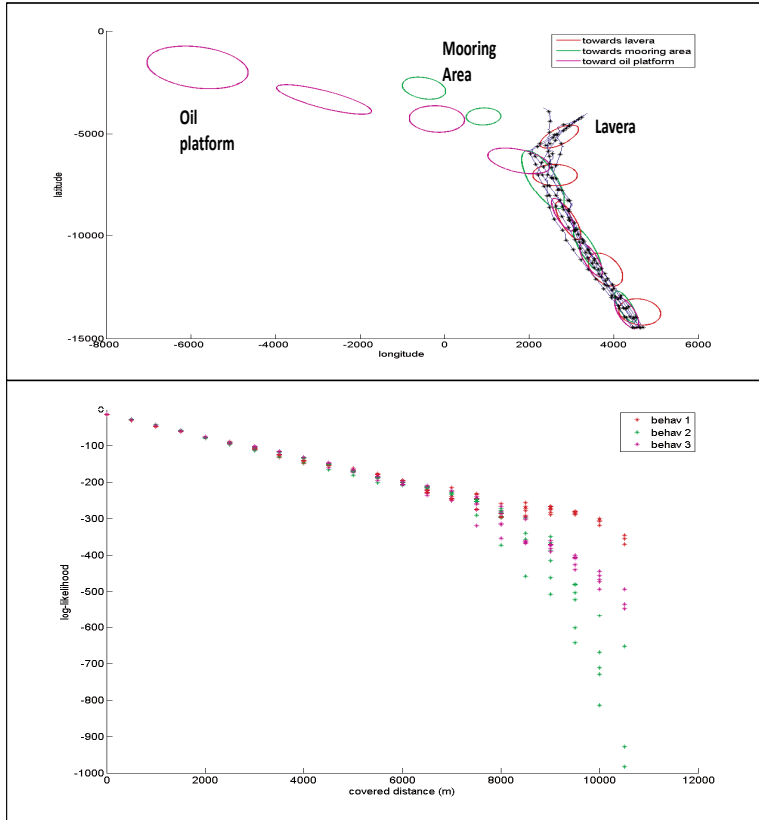


FIGURE 4.12 – Figure (a) Ensemble de trajectoires de bateaux se dirigeant vers Lavera.(b) Les valeurs du log-vraisemblance $P(O_1 \dots O_t / \lambda_i)_{1 \leq i \leq 7}$ pour des observations enregistrées tous les 500 m.

Exemple 2 : reconnaissance d'un ensemble de comportements nominaux de bateau se dirigeant vers Lavera

On considère à travers l'exemple de la figure (4.12) un ensemble de trajectoires de bateaux se dirigeant vers Lavera. Chaque fois que le bateau parcourt une distance de 1000 m, les valeurs des log-vraisemblances des observations sont calculées par rapport à trois modèles de comportements de bateau :

- comportement de bateaux se dirigeant vers Lavera (λ_3),
- comportement de bateaux se dirigeant vers Darse 1 (λ_6)
- comportement de bateaux se dirigeant vers Mouillage Nord (λ_4).

On voit que $P(O/\lambda_3) \approx P(O/\lambda_4) \approx P(O/\lambda_6)$ quand le bateau passe par les zones communes : A_1 , A_2 , A_3 et B . En dépassant la zone B , $P(O/\lambda_3)$ augmente considé-

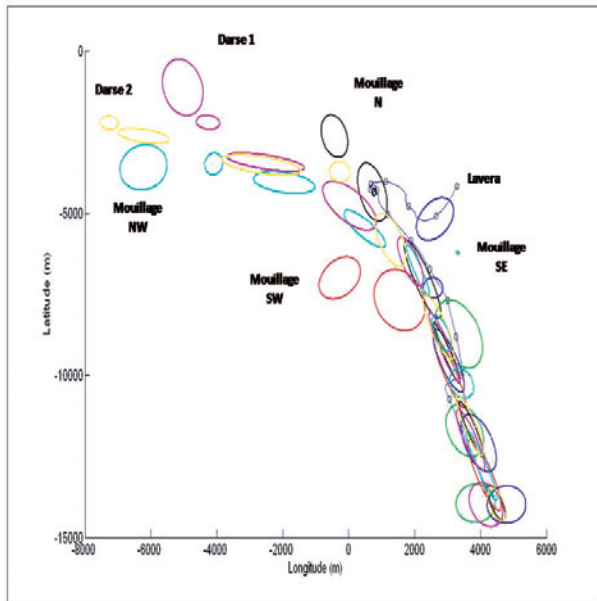
ablement par rapport à $P(O/\lambda_4)$ et $P(O/\lambda_6)$.

En effet, lors du passage par les zones communes, les valeurs des log-vraisemblances sont très proches les unes des autres. Le processus de reconnaissance est de plus en plus pertinent quand le bateau quitte les zones communes, ceci confirme ce qui a été démontré par l'exemple 1.

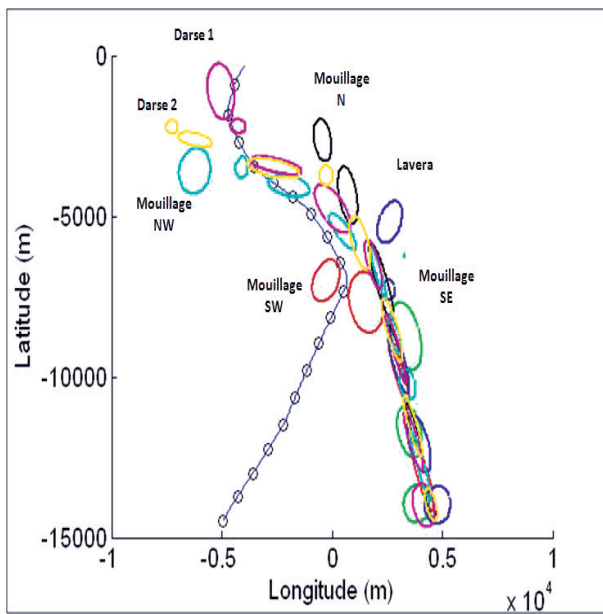
En somme, en utilisant une approche probabiliste par MMC, nous avons pu définir, modéliser et reconnaître les comportements nominaux de bateaux (c'est-à-dire ceux qui se déplacent en respectant les chenaux d'accès au port).

En se basant alors sur cette définition, comment peut-on définir mathématiquement un *comportement anormal* ?

comportement anormal Nous avons défini un comportement anormal d'un bateau par *celui qui change brutalement de direction lors de son déplacement dans le port* (figure (4.13a)) ou *celui qui ne respecte pas les règles de la navigation* (figure (4.13b)).



(a) Un exemple de comportements de bateaux qui change de direction et se dirige vers Lavera après avoir atteint la zone de Mouillage Nord.



(b) Un exemple de comportements de bateau qui ne respecte pas les règles de la navigation.

FIGURE 4.13 – Deux exemples de comportements de bateaux

- La figure (4.13a) est un exemple d'un bateau qui change de direction et se dirige vers Lavera après avoir atteint la zone de Mouillage N. Comme nous l'avons vu dans les exemples des paragraphes précédents, le module de reconnaissance de comportements BPAM va nous décrire exactement le comportement de ce bateau (c'est-à-dire son passage de la zone de mouillage Nord vers Lavera). Ce type de comportement peut-être considéré comme dangereux si le bateau change *brutalement de cap en accélérant* lors de son passage de la zone du mouillage Nord vers Lavera. Néanmoins, nos MMC sont construits à partir des positions du bateau et les informations concernant le changement de cap et l'accélération ne sont pas intégrées dans le modèle de comportement. Dans ce cas, le module peut juste nous dire que le bateau dévie de sa trajectoire et on ne dispose d'aucune information sur son changement de vitesse (accélération entre autres). Nous nous sommes alors demandés si un simple changement de trajectoire du bateau pourrait nous amener à considérer son comportement comme anormal.

Afin de pouvoir intégrer dans nos modèle de comportements un autre type d'observation tel que la vitesse (en plus des positions du bateau), la structure des MMC devient insuffisante. Dans [100, 99], l'auteur utilise un modèle CHMM (Coupled Hidden Markov Models) ou CCMC (Chaînes Couplées de Markov Cachées) afin de pouvoir intégrer plus d'un type d'observation dans son modèle associé à l'activité de conduite des conducteurs. En essayant d'utiliser cette approche (CCMC), nous nous sommes aperçus que l'application des algorithmes forward, backward et Baum-Welch est d'une très haute complexité algorithmique. En outre, il est possible d'avoir une réponse à ce type de problème de manière plus simple. D'où notre choix d'une approche complémentaire, celle de l'approche réactive, qui a fait l'objet de notre étude dans la section 3.2.

- La figure (4.13)-(b) est un deuxième exemple d'un comportement anormal d'un bateau. Il s'agit d'un bateau qui ne respecte pas les règles de la navigation en se déplaçant en dehors de son chenal. En calculant le log-vraisemblance $\log P(O_1 \dots O_t / \lambda_i)$ des positions de ce comportement par rapport à l'ensemble Λ , on trouve des valeurs très faibles : $\log \{P(O / \lambda_i)\}_{1 \leq i \leq 7} = -\infty$. La définition probabiliste d'un comportement anormal de bateau, dans ce cas, peut-être associée à des valeurs de log-vraisemblances très faibles. Ces valeurs seront en dehors d'un certain intervalle I qu'il va falloir définir : $\log P(O_1 \dots O_t / \lambda_i) \notin I$. Et I sera alors l'intervalle qui contient l'ensemble des valeurs qui permettent de considérer un comportement comme habituel. Cet intervalle peut-être défini suite à une phase d'expérimentation.

Conclusion

En conclusion, la force de ces approches en général à savoir, les MMC dépend fortement du processus d'apprentissage qui va déterminer les résultats de la reconnaissance par la suite. Puisqu'un comportement anormal n'a pas été appris et intégré dans le BPAM, il est difficile de le reconnaître. En outre, les petits bateaux circulent librement dans le GPMM. Par conséquent, les pistes systèmes relatives à leurs déplacements présentent de fortes irrégularité, ce qui rend difficile l'apprentissage de leurs comportements.

Nous avons alors opté pour l'approche réactive synchrone que nous avons présenté dans la section 3.2 afin de traiter cette difficulté. Tous les scénarios de comportements de bateaux que nous n'avons pas pu traiter par le BPAM seront pris en charge par le BRAM.

4.2 Reconnaissance de comportement par approche événementielle

4.2.1 Fonctionnement général du module BRAM

Le BRAM (Figure (4.14)) a principalement été construit pour reconnaître les comportements anormaux de tous types de bateaux.

La législation a établi pour les zones à surveiller des caractéristiques en limite de vitesse par type de véhicule, en limitation d'accès, en restriction de mouvement et de mouillage. A partir de ces caractéristiques et des pistes systèmes, un ensemble d'informations pertinentes qu'on a appelé *constats instantanés* a été construit. Ces constats instantanés sont basés sur la présence dans des zones de différents types, sur les règles relatives à ces zones, sur le type de cargaison ainsi que sur le type du véhicule (tableau (4.2)).

Tous les scénarios sont des programmes réactifs synchrones qui ont été implémentés en ESTEREL et les constats forment les événements en entrée de ces *scénarios* c'est à dire les signaux d'entrée des programmes ESTEREL.

La période d'exécution de ce module est dictée par l'environnement c'est à dire par le système SECMAR global. En effet, ce temps dépend des temps d'acquisition des informations des différents capteurs et du temps de leur fusion en de nouvelles entrées de pistes systèmes. Le comportement du module BRAM est le suivant, à chaque pas temporel :

- le module reçoit de l'extérieur, pour chacune des nouvelles entrées des pistes systèmes, l'ensemble des constats instantanés (une centaine) ;

- ces constats sont présentés, comme signaux d'entrée, à l'ensemble des scénarios ESTEREL prédéfinis. Les scénarios réagissent en parallèle, produisant chacun une information de sortie ;
- les informations en sortie des scénarios sont analysées et une phase de fusion réalise un mixage des résultats de chaque scénario pour élaborer une note de menace unique pour chaque bateau. Un limiteur permet d'éviter de surcharger le système avec des alarmes de même nature et assure la prise en compte de la décision opérateur dans la note finale de menace.

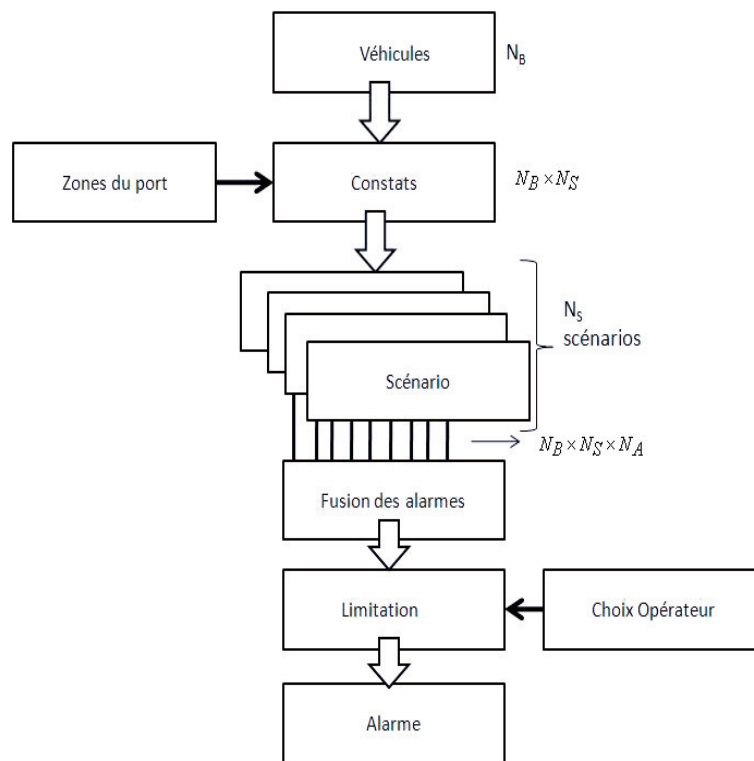


FIGURE 4.14 – Module de reconnaissance de comportements par approche réactive : le BRAM

La relation entre la durée du pas temporel et le temps de réaction des programmes ESTEREL est un élément primordial : la durée du pas temporel du BAM dictée par le système global SEC MAR est suffisamment grande devant les temps d'exécution de tous les instants de tous les programmes ESTEREL du BRAM.

4.2.2 Les scénarios de comportements

4.2.2.1 Trois types de scénarios

Nous avons déterminé trois types de scénarios à reconnaître : des scénarios *agressifs*, *transgressifs* et *nominaux*. Les trois types de scénarios conduisent à la génération d'une alarme à chaque pas temporel. Les scénarios nominaux ont été introduits dans le BRAM pour avoir une vue globale sur les deux approches (probabiliste par MMC et réactive synchrone par ESTEREL). En effet, le fait que chacune des deux approches détectent les mêmes comportements nominaux, nous permet de comparer les qualités relatives de détection des deux approches, ce qui sera détaillé dans la section (4.3).

Les trois types de scénario que nous avons déterminés sont :

- Les *scénarios nominaux* : c'est l'ensemble des comportements des bateaux commerciaux qui passent exactement par les zones fonctionnelles délimitées par la législation. Les comportements nominaux de bateaux ont donc été implémentés dans le but de comparer les résultats obtenus par le BPAM.
- Les *scénarios agressifs* : c'est l'ensemble des scénarios qui détectent les comportements anormaux et donc potentiellement menaçants des bateaux.
- Les *scénarios transgressifs* : c'est l'ensemble des comportements de bateaux qui ne respectent pas les règles de la navigation ou les règles du port.

4.2.2.2 Scénarios et constats instantanés implémentés pour le GPMM

La table (4.2) montre un échantillon significatif des constats instantanés représentant les signaux d'entrée de ces scénarios ESTEREL. Le tableau (4.3) donne l'ensemble des scénarios que nous avons imaginés avec l'équipe de sécurité du GPMM et implémentés en ESTEREL.

Présence dans une zone	<p>est_en_zone_A1 ... est_en_zone_N est_en_Darse1 ... est_en_DarseSud est_a_Lavera est_en_Mouillage_N est_en_Mouillage_NW est_en_Mouillage_SE est_en_Mouillage_SW</p>
Infraction	<p>est_en_zone_interdite est_en_zone_restreinte va_trop_vite</p>
Cinématique	<p>change_cap_brutal change_vitesse_brutal change_position_brutal est_en_contact_dangereux est_en_contact_10min, est_en_contact_dangereux_10min est_en_contact,est_en_contact_dangereux est_arrete est_trop_pres_de</p>
Cargaison	<p>est_avec_cargaison est_avec_cargaison_dangeureuse est_sans_cargaison</p>
Type de navire	<p>est_vehicule_rapide est_un_bateau_de_peche est_vehicule_petit est_vehicule_commercial</p>

TABLE 4.2 – Liste des constats instantanés

Scénarios nominaux	vers Darse1 vers Darse2 vers Lavera vers Mouillage N vers Mouillage NW vers Mouillage SE vers Mouillage SW
Scénarios agressifs	Abordage de navire commercialArrete Abordage de navire commercial Change cap brutal dans zone restreinte Change cap brutal dans zone interdite Change vitesse brutal dans zone restreinte Change vitesse brutal dans zone interdite Change vitesse brutal en contact Entre en collision Entre en collision dangereux Small boat en contact Zig zag dangereux Menace JetSki
Scénarios transgressifs	Cargo reste trop pres de Va trop vite en zone interdite Va trop vite en zone restreinte Peche en zone interdite Peche en zone restreinte Reste en zone interdite Reste en zone restreinte FastBoat en Darse 1 FastBoat en Darse 2 SmallBoat en Darse 1 SmallBoat en Darse 2

TABLE 4.3 – Liste des scénarios

4.2.3 Les différents niveaux d’alarme détectés par la reconnaissance

A la fin de leur exécution, chaque scénario ESTEREL délivre une alarme. Il s’agit du signal de sortie de chaque programme ESTEREL qui établit le niveau de recon-

naissance du scénario en question.

Nous avons considéré 5 niveaux d'alarme :

- **standby** : le scénario est initialisé mais la présence des signaux en entrée n'a pas permis à la reconnaissance de se déclencher ;
- **running** : étant donnée la présence de certains signaux en entrée, le processus de reconnaissance de comportement a commencé ;
- **pre_matching** : étant donné la présence de certains signaux en entrée, le scénario commence à correspondre au comportement en cours du bateau ;
- **non_matching** : le scénario ne correspond plus au comportement en cours ;
- **matching** : le scénario du comportement en cours est reconnu ;

A la fin du processus de reconnaissance, une unique alarme globale doit être obtenue pour chaque bateau alors que nous disposons d'une alarme pour chaque scénario. A titre d'exemple, un comportement de bateau peut-être considéré comme dangereux par un scénario (va trop vite en zone interdite) et non menaçant par un autre (est trop près de). Pour cette raison, une phase de consolidation est nécessaire pour obtenir une réponse unique. Cette alarme est alors utilisée pour colorier à l'écran de l'IHM le bateau à chaque pas temporel. Notons que l'opérateur (le surveillant du port) peut modifier ou ignorer l'alarme. Il peut, à titre d'exemple, ignorer un niveau d'alarme indiquant un danger par le scénario quand il est sûr que le comportement du bateau n'est pas du tout dangereux.

4.2.4 Spécification de la phase de reconnaissance des scénarios ESTEREL

La phase de reconnaissance est basée sur les constats instantanés obtenus à partir des pistes systèmes (table (4.2)). A chaque pas temporel, chaque scénario reçoit l'ensemble des constats instantanés c'est à dire qu'à chaque réaction, chaque programme ESTEREL connaît le statut de présence ou d'absence de tous les constats du tableau (4.2). Notons qu'à chaque pas temporel, tous les programmes ESTEREL concernant le même bateau vont recevoir les mêmes informations sur ce bateau.

4.2.4.1 L'algorithme du module

Soient :

- $N_B(t)$ le nombre de bateau présent à l'instant t dans la zone couverte par les capteurs ;
- N_C le nombre de constats instantanés obtenus à partir des pistes systèmes pour chaque bateau. Ces constats correspondent aux inputs des scénario ;

- N_S le nombre de scénarios (systèmes réactifs implémentés en ESTEREL) à reconnaître ;
- N_A le nombre des alarmes ;
- $[S_1, \dots, S_{N_S}]$ le vecteur des N_S scénarios prédéfinis qui représentent tous les comportements possibles à reconnaître ;
- $\left[S_1^{e^{i,1 < i < \text{card}(S_1)}}(b_i)(t), S_2^{e^{j,1 < j < \text{card}(S_2)}}(b_i)(t), \dots, S_{N_S}^{e^{k,1 < k < \text{card}(S_{N_S})}}(b_i)(t) \right]$ est le vecteur des scénarios, chacun étant considéré dans son état courant, $S_k^{e^{i,1 < i < \text{card}(S_k)}}$ est le scénario S_k dans l'état $e^{i,1 < i < \text{card}(S_k)}$ pour le bateau b_i à l'instant t , $\text{card}(S_k)$ étant le nombre des états possibles du scénario S_k et e^1 est son état initial.

L'algorithme du BRAM est le suivant :

1. Pour initialiser le BRAM, le BAM lui fournit le vecteur des N_S scénarios ; chaque scénario est un programme ESTEREL dont le code compilé a été traduit dans le langage hôte choisi par SECMAR (en l'occurrence le langage Java).
2. A l'instant t ou un bateau b_i , $1 \leq i \leq N_B(t)$ apparaît pour la première fois dans la zone surveillée du port, on lui fournit le vecteur $\left[S_1^{e^1}(b_i)(t), S_2^{e^1}(b_i)(t), \dots, S_{N_S}^{e^1}(b_i)(t) \right]$ des scénarios ESTEREL, chacun dans son état initial.
3. Pour chaque pas temporel t du BAM et pour chaque bateau b_i , $1 \leq i \leq N_B(t)$, l'algorithme suivant est exécuté :
 - (a) Le BAM calcule le statut de présence ou d'absence des N_C constats instantanés et range ces statuts dans le tableau $[c_1(b_i)(t), c_2(b_i)(t), \dots, c_{N_C}(b_i)(t)]$ qui décrit tous les événements concernant le comportement du bateau b_i à l'instant t . Notons que ces statuts sont des valeurs booléennes.
 - (b) Le vecteur $[c_1(b_i)(t), c_2(b_i)(t), \dots, c_{N_C}(b_i)(t)]$, représentant les valeurs booléennes (*vrai* pour présent et *faux* pour absent) de tous les signaux d'entrée. Ces signaux sont donnés en entrée à chacun des scénarios $\left[S_1^{e^{i,1 < i < \text{card}(S_1)}}(b_i)(t), S_2^{e^{j,1 < j < \text{card}(S_2)}}(b_i)(t), \dots, S_{N_S}^{e^{k,1 < k < \text{card}(S_{N_S})}}(b_i)(t) \right]$ (le scénario S_1 étant dans l'état e^i , S_2 dans l'état e^j, \dots, S_{N_S} dans l'état e^k), les scénarios exécutent chacun leur réaction, passent dans un nouvel état en émettant une alarme. Pour chaque scénario S_j , $1 \leq j \leq N_S$, étant à l'état e^i , pour le bateau b_i et à l'instant t :

$$S_j^{e^i} \xrightarrow[a_j]{c_1, \dots, c_{N_C}} S_j^{e^{i'}}$$

où a_j est le signal émis à l'instant t lors de la réaction du scénario $S_j^{e^i}$ pour le bateau b_i et $S_j^{e^{i'}}$ l'état atteint après la réaction.

- (c) Les N_S alarmes $[a_{S_1}^{m,1 < m < N_A}(b_i)(t), \dots, a_{S_{N_S}}^{m,1 < m < N_A}(b_i)(t)]$ (une par scénarios) du bateau b_i à l'instant t sont fusionnées en une alarme unique retournée par le BAM au système global SECMAR qui l'affiche sur l'écran des opérateurs de surveillance, ceux-ci pouvant alors en modifier le statut.

Nous allons par la suite évaluer une borne supérieure à la complexité en espace de l'algorithme. Soit N_{B_M} le nombre maximum de bateaux présents en même temps dans le port, pour chaque bateau :

- une matrice d'entiers de taille $N_{B_M} \times N_S$ pour stocker les entiers constituant les états courants des N_S scénarios ;
- une matrice de booléens de taille $N_{B_M} \times N_C$ pour stocker les booléens constituant les valeurs des N_C constats instantanés ;
- une matrice de booléens de taille $N_{B_M} \times N_S$ pour stocker les alarmes de sorties des N_S scénarios avant fusion.
- le nombre total d'états de l'ensemble des modules ESTEREL stockés : $N_{total} = \sum_{k=1}^{N_S} Card(S_k)$

Dans le GPMM, le nombre maximal de bateau est $N_{B_M} = 500$ pendant la période estivale étant donné l'affluence des bateaux de plaisance. Par contre, ce nombre est inférieur à 30 pendant les autres saisons de l'année excepté lors de conditions particulières (en hiver 2010 on a dénombré plus de 70 pétroliers dans la zone portuaire). Dans l'implémentation actuelle $N_C \cong 100$ et $N_S \cong 40$.

Concernant les contraintes temporelles, pour le projet SECMAR, la durée du pas temporel a été estimée à 3 secondes. (Cette estimation temporelle dépend des traitements spécifiques faits par les capteurs et ne dépend pas des programmes ESTEREL). Ce temps est très largement supérieur au temps limite requis pour le calcul de 100 constats instantanés par piste système (c'est-à-dire par bateau), pour évaluer 40 scénarios par bateau et pour fusionner tous les niveaux d'alarmes pour chaque bateau. Par conséquent, l'hypothèse synchrone est correctement respectée.

4.2.5 Analyse de scénario ESTEREL

Pure Inputs	Pure Outputs
<code>= is_on_area_A_1</code>	<code>standby</code>
<code>= is_on_area_A_2</code>	<code>running</code>
<code>= is_on_area_A_3</code>	<code>pre_matching</code>
<code>= is_on_area_B</code>	<code>non_matching</code>
<code>= is_on_area_C</code>	<code>matching</code>
<code>= is_on_area_D</code>	<code>ended</code>
<code>= is_on_area_E</code>	
<code>= is_on_area_HB_1</code>	
<code>= is_on_area_HB_2</code>	
<code>= is_on_Lavera</code>	
<code>= is_on_mooring_SE</code>	
<code>= is_on_mooring_SW</code>	
<code>= is_on_mooring_N</code>	
<code>= is_on_mooring_NW</code>	
<code>= is_on_forbidden_area</code>	
<code>= is_on_restricted_area</code>	
<code>= change_heading_violent</code>	
<code>= change_position_violent</code>	
<code>= change_speed_violent</code>	
<code>= is_with_cargo</code>	
<code>= is_very_fast</code>	
<code>= is_small_boat</code>	
<code>= is_with_dangerous_cargo</code>	
<code>= is_without_cargo</code>	
<code>= is_an_oil_tanker</code>	
<code>= is_a_commercial_boat</code>	
<code>= is_a_small_boat</code>	

`tick`

FIGURE 4.15 – simulateur XES

Le simulateur ESTEREL XES de la figure (4.15) nous permettra de visualiser les résultats de reconnaissance. Il est constitué de trois parties :

1. Le tableau des signaux d'entrée où on peut modifier le statut des constats.
2. Le tableau des signaux de sortie où les niveaux d'alarme sont visualisés.
3. le bouton de déclenchement d'une réaction. Dans ce simulateur, ce bouton est appelé « tick » qui représente l'instant de l'horloge logique.

Dans cette section, nous allons détailler trois types de scénarios ESTEREL : agressif, transgressif et nominal.

4.2.5.1 Un scénario agressif

```
module ChangeCapBrutalDansZoneInterdite:
input est_en_zone_interdite, change_cap_brutal, timeOut;
output standby, running, matching;
loop
  emit standby;
  await est_en_zone_interdite;
  emit running;
  abort
  await change_cap_brutal;
  emit matching;
  await timeOut
  when [not est_en_zone_interdite]
end loop
end module
```

FIGURE 4.16 – Scénario agressif

Le scénario de la figure (4.16) est un scénario de comportement agressif de bateau qui change brutalement de cap au cours de son déplacement dans une zone interdite :

- au premier instant, le module est activé quelles que soient les valeurs des constats. L’alarme **standby** est envoyée. Le module est considéré comme simplement *initialisé* ;
- ensuite, à l’instant où le signal **est_en_zone_interdite** est présent dans les entrées du scénarios, le bateau est considéré comme étant entré dans une zone interdite et l’alarme **running** est émise. Le scénario commence à être *reconnu* ;
- par la suite, si le bateau, tout en étant toujours dans la zone interdite, change brutalement de cap (c’est-à-dire que le signal **est_en_zone_interdite** est toujours *présent* et le signal **change_cap_brutal** devient *présent*, le module émet l’alarme **matching**). Le scénario est *reconnu* et le module se met à attendre **deltaT** ;
- a l’instant où le bateau sort de la zone interdite, c’est-à-dire à la réaction où le signal **est_en_zone_interdite** est *absent*, le module envoie l’alarme **standby** et ce quelle que soit la valeur du signal **change_cap_brutal** ;
- quand le bateau a effectué un changement de cap brutal dans une zone à accès interdit et est resté dans cette zone, on attend l’intervalle de temps *deltaT* (qui

- sera indiqué par la présence dans les signaux d'entrée du signal `deltaT`) avant de réinitialiser le scénario et le remettre en attente de la présence dans une zone à accès interdit ;
- le comportement du bateau n'est plus considéré comme agressif lorsqu'il quitte la zone interdite. Par conséquent, le scénario n'est plus reconnu, l'alarme `standby` est envoyée puis le module se met à attendre que le bateau retourne dans une zone interdite ;
 - à la fin de la temporisation `deltaT`, nous avons choisi de réinitialiser le scénario, le module envoie l'alarme `standby` et se remet en attente de sa présence en zone interdite). Le module retourne ainsi à son état initial. Ceci est effectué en levant l'exception `T` par l'instruction « `exit T` ». Nous avons considéré qu'au cours de cette phase de temporisation, le scénario a suffisamment eu le temps d'avertir l'équipe de sécurité du port du danger que présente ce bateau.

4.2.5.2 Un scénario transgressif

```

Module vaTropViteDansZoneRestreinte:
input est_vehicule_petit, est_en_zone_restreinte, va_trop_vite, deltaT;
output standby, running, pre_matching, matching;
loop
  await est_en_zone_restreinte;
  emit running;
  weak abort
    await va_trop_vite;
    emit pre_matching;
    weak abort
      await deltaT;
      emit matching;
      halt
    when [not va_trop_vite];
    emit running;
    halt
  when [not_est_en_zone_restreinte];
  emit standby;
end loop
end module

```

FIGURE 4.17 – Scénario transgressif

Le scénario de la figure (4.17) est celui d'un bateau qui est reconnu comme *petit* (par un module de classification des bateaux par reconnaissance d'image intégré dans le projet SECMAR) qui va trop vite pendant un laps temps prédéfini (ici δT) dans une zone restreinte. La structure de ce scénario est très proche du scénario agressif de la figure (4.16).

Dans ce scénario :

- il faut attendre que deux conditions soient présentes avant de pouvoir commencer le processus de reconnaissance, tout d'abord que le bateau soit petit, ensuite que le bateau soit entré dans une zone à accès restreint (respectivement présences des signaux `est_vehicule_petit` et `est_en_zone_restreinte`);
- à l'instant où le bateau commence à accélérer tout en restant dans la zone restreinte (le signal `est_en_zone_restreinte` reste *présent* et le signal `va_trop_vite` devient *présent*), comme le scénario commence à correspondre au comportement du bateau en cours, l'alarme `pre_matching` est envoyée;
- le scénario ne sera considéré comme reconnu qu'au bout du laps de temps δT pendant lequel le bateau est toujours dans la zone et va toujours trop vite. L'alarme `matching` est alors envoyée. L'instruction « halt » a été utilisée pour que le processus de reconnaissance reste activé jusqu'à ce que la vitesse excessive du bateau s'arrête. A ce moment, le scénario émet l'alarme `running` et se bloque (deuxième opérateur « halt »);
- au cours de la phase de reconnaissance du scénario (quand l'alarme `matching` est envoyé), si le bateau quitte la zone restreinte, le scénario est réinitialisé quelle que soit la vitesse du bateau et l'alarme `standby` est envoyée. Le comportement du bateau n'est alors plus considéré comme transgressif jusqu'à ce qu'il retourne dans une zone restreinte;
- l'opérateur « loop » a ici été utilisé pour exécuter indéfiniment le programme afin donc de recommencer une nouvelle exécution à l'instant où la dernière exécution se termine.

Notons que ce scénario concerne uniquement les petits bateaux contrairement au scénario agressif précédent qui concerne tous types de bateaux.

4.2.5.3 Les scénarios nominaux

```
scenario versDarse1
include_files
input_signals.str11
output_signals.str11

begin_gardes_list is_a_commercial_boat

nominal_sequence_list
est_en_zone_A_1
est_en_zone_A_2
est_en_zone_A_3
est_en_zone_B
est_en_zone_C
est_en_zone_D
est_en_zone_E

end_gardes_list est_en_Darse1

end scenario
```

FIGURE 4.18 – Scénario nominal

Les bateaux commerciaux ont un comportement précis et bien-défini. Le comportement de la figure (4.18) est celui d'un comportement nominal d'un bateau commercial se dirigeant vers Darse 1.

Les scénarios des comportements nominaux ont la structure suivante :

- attendre de détecter un bateau commercial ;
- attendre que ce bateau passe par une séquence de zones préalablement définie décrivant le chenal d'accès du bateau ;
- tolérer que le bateau soit *perdu* au cours de son déplacement (par exemple les capteurs ne l'ont pas vu pendant son passage dans une zone) puis réapparaisse dans l'une des zones suivantes ;
- ne lever un niveau d'alarme rouge exprimant un danger qu'au cas où le bateau entre dans des zones qui lui sont strictement interdites.

Comme tous les scénarios de comportements nominaux de bateaux commerciaux ont la même forme très régulière, nous avons conçu un outil logiciel appelé *SCE* (pour

*SCE*nario) pour générer automatiquement dans le langage ESTEREL les comportements de différents types de bateaux commerciaux (pétrolier vers Lavera, porte-conteneur vers Darse 2, etc).

Ce générateur se base sur quatre types d'information :

- une liste de *gardes* d'entrée optionnelle représentant l'ensemble des conditions qui doivent être satisfaites avant que la phase de reconnaissance ne commence ;
- la séquence ordonnée non vide des zones par lesquelles les bateaux visés par ce scénario sont obligés de passer ;
- une liste de *zones interdites*. Notons que quand cet ensemble est vide, on considère que toutes les zones sont interdites à l'exception de celles formant la séquence ordonnée de présence dans des zones ;
- une liste de *gardes* de sortie optionnelles représentant l'ensemble des conditions à satisfaire avant de considérer que la phase de reconnaissance est terminée.

Un programme SCE inclut la liste de tous les signaux d'entrée et de sortie que nous avons considéré et retourne le même type d'alarme que les autres scénarios (agressifs et transgressifs) :

- `standby` pour dire que le scénario vient d'être initialisé ;
- `running` pour dire que le scénario a satisfait ses gardes d'entrée ;
- `matching` pour dire que le scénario a procédé correctement le long de la séquence de zones et a satisfait à ses gardes de sorties ;
- `missing_zone` pour dire que le bateau est toujours dans une zone autorisée mais que par manque d'information capteur, le bateau est considéré comme ayant manqué une ou plusieurs zones au cours de son déplacement ;
- `non_matching` pour dire que le bateau est entré dans une zone interdite.

Ce générateur, implémenté en C++, traduit tout scénario exprimé en format *sce* (comme celui de la figure 4.18) en un programme ESTEREL. Tous les scénarios nominaux de comportements de bateaux commerciaux ont été implémentés grâce à ce générateur.

La figure (4.18) montre un comportement de bateau qui se dirige vers Darse 1, nous pouvons remarquer que :

1. ce programme inclut des fichiers d'entrée et de sortie (`input_signals_.strll` et `output_signals_.strll`) ;
2. la garde qui représente la condition nécessaire à la reconnaissance du scénario est que le bateau soit commercial (c'est-à-dire `est_bateau_commercial`) ;
3. la liste de la séquence nominale est que le bateau passe dans l'ordre par les zones : *A*₁, *A*₂, *A*₃, *B*, *C*, *D*, *E* avant d'atteindre *Darse1* ;

La description du programme ESTEREL associé à ce programme est la suivante :

- au premier tick, le scénario est initialisé (`standby` alarme est *présent*);
- par la suite, si le signal d'entrée `est_bateau_commercial` est reçu par le module, l'alarme `matching` est envoyée;
- si le module reçoit des informations traduisant que le bateau a manqué une ou plusieurs zones au cours de son déplacement tout en passant par des zones autorisées, l'alarme `missing_zone` est émise. Notons que le concepteur du système SECMAR ayant considéré que cette alarme ne transmettait qu'une information de faible intérêt, elle a été remplacée par l'alarme `matching`;
- quand le bateau entre dans une zone interdite, l'alarme `non_matching` est envoyée jusqu'à ce que le bateau retourne dans une zone autorisée (et dans ce cas l'alarme `matching` est envoyée).

Conclusion

Contrairement au BPAM, où on utilise directement les données brutes des pistes systèmes, le BRAM travaille sur des constats instantanés calculés à partir des pistes système et qui constituent l'ensemble des informations pertinentes sur notre système sur son lieu d'implantation. Cette méthode qui consiste à cumuler des analyses instantanées puis à intégrer les aspects temporels au travers de programmes réactifs s'est avéré être très performante que ce soit en terme de qualité, d'efficacité (cette méthode par systèmes réactifs synchrone est rapide en temps d'exécution et a très bien pu être intégrée dans SECMAR), d'adaptabilité et de fonctionnalité (les trois types de scénarios sont suffisants pour une bonne couverture fonctionnelle de notre problème de reconnaissance par méthode synchrone). Le découplage des données techniques et des données de « data-mining » qui est fait par rapport aux scénarios permet une grande richesse d'expression en langage naturel.

Les résultats obtenus lors des essais sur site des comportements nominaux, agressifs et transgressifs furent concluants. Néanmoins, les résultats obtenus sur les comportements nominaux sont en redondance avec la méthode probabiliste. Nous avons rencontré certains problèmes que nous avons résolus, comme le maintien des alarmes de certains scénarios agressifs, ou des scénarios agressifs trop sensibles. Nous avons aussi eu des difficultés liées au système lui-même, des données AIS qui ne sont pas toujours fiables et peuvent conduire à des fausses alertes, ou encore des *fausses pistes* encore présentes après la fusion et qui, par leur comportement totalement erratique, conduisaient aussi à de fausses alarmes.

Notons aussi qu'à cause de problèmes logiciels et matériels (spécifiques aux capteurs) rencontrés lors du projet SECMAR, la phase de *réglage* des scénarios fut malheureusement beaucoup trop courte et n'a été effectuée que pendant la phase

d'intégration du système global, certains capteurs et le module de fusion présentant encore des problèmes et des limitations.

Dans la section suivante, nous discuterons, des différents résultats obtenus par ce module construit par l'approche probabiliste MMC et le langage réactif synchrone ESTEREL. Par ailleurs, nous mettrons en évidence leur complémentarité.

4.3 Discussion et complémentarité des deux modules

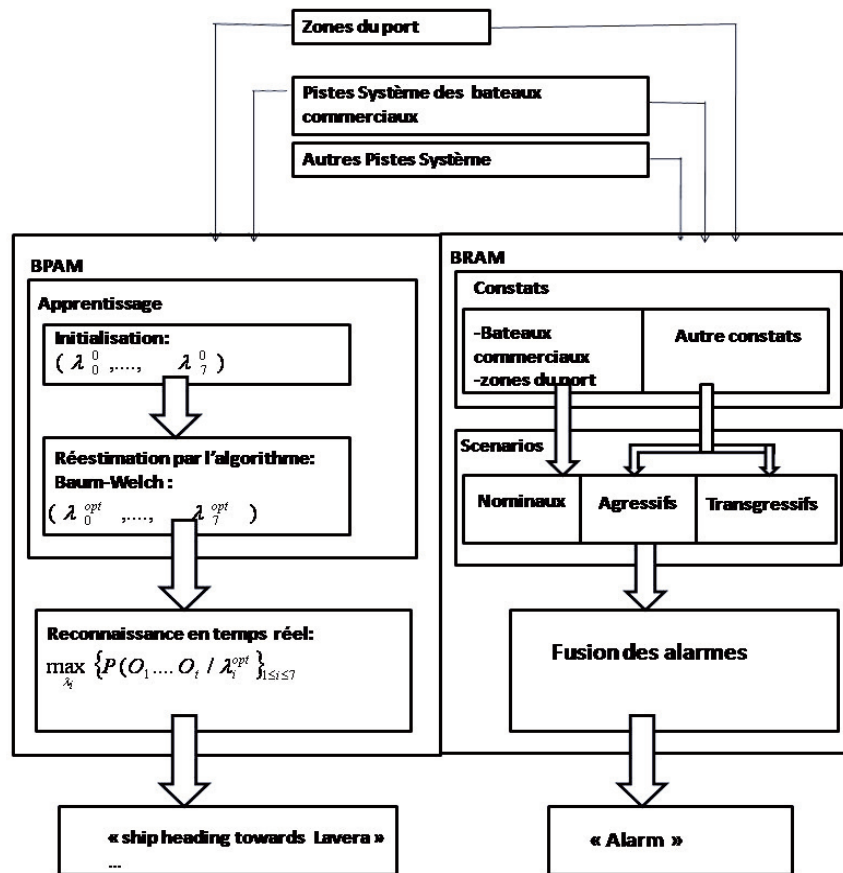


FIGURE 4.19 – Le BAM

Le système d'analyse de comportement (BAM) que nous avons conçu (voir figure (4.19)) est formée de deux sous-modules basés sur deux approches : le BPAM qui est

basé sur l'approche MMC et le BRAM qui est basé sur le langage réactif synchrone ESTEREL.

La construction du BPAM par les MMC a été effectuée en passant par deux phases : une phase d'apprentissage hors-ligne des comportements de navires et une phase de reconnaissance en temps-réel du comportement de tout bateau pénétrant dans le port.

Lorsqu'un nouveau bateau de type commercial, à titre d'exemple, arrive dans le port, le BPAM prédit son comportement courant à chaque pas temporel en calculant et en comparant les log-vraisemblances de ces positions par rapport à tous les modèles de comportements de bateaux dans le port obtenus par MMC.

Néanmoins, l'utilisation des valeurs des log-vraisemblances pour détecter des comportements menaçants n'était pas très efficace. Il était alors difficile d'utiliser le BPAM pour affirmer en temps-réel si le comportement du bateau est menaçant ou non. Par ailleurs, nous avons considéré que cette reconnaissance de comportement menaçant était difficile car il n'a pas été appris et intégré dans le BPAM.

Nous rappelons alors les deux définitions pour lesquelles nous avons opté :

- La première définition concerne *le comportement d'un bateau qui change brutalement de direction lors de son déplacement dans le port*, ce qui veut dire statistiquement qu'en quittant les zones communes à tous les comportements, les valeurs des log-vraisemblances calculées sur les positions de bateaux par rapport aux modèles de comportements nominaux se mettent à fluctuer de façon anormale. Cette fluctuation de trajectoires ne traduit pas forcément un comportement anormal. Par ailleurs, le calcul statistique du log-vraisemblance est uniquement basé sur les positions du bateau et *le changement brutal de cap* par exemple, qui est une information importante pour considérer un comportement comme dangereux, n'a pas été pris en compte par nos MMC, ce qui rend cette définition de comportement anormal insuffisante.
- La deuxième définition concerne *le comportement d'un bateau qui ne respecte pas les règles maritimes et se déplace en dehors de ces chenaux d'accès*. Les valeurs des log-vraisemblances sont très faibles par rapport à nos attentes. Dans ce cas, il s'agit d'un comportement improbable.

Pour récapituler, les modèles de comportements que nous avons construits par MMC sont uniquement basées sur les positions du bateau et n'intègre pas d'autres types d'informations à savoir, le type de bateau, le type de cargaison, les conditions météorologiques, le changement brutal de cap et l'accélération dans des zones dangereuses.

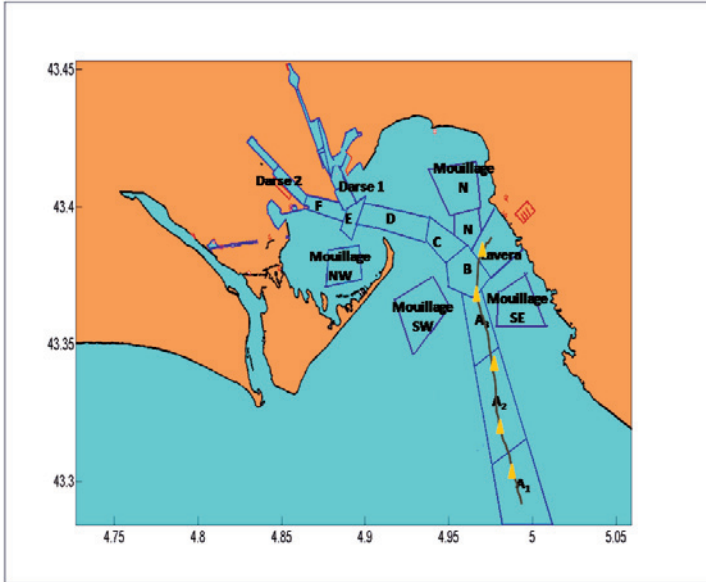
Etant donnée la limite de cette définition et sachant que l'utilisation des MMC n'était possible que pour des bateaux commerciaux ayant un comportement régulier, l'adoption d'une approche complémentaire s'est imposée.

Le BRAM conçu par l'approche réactive synchrone a alors été principalement utilisé pour traiter le cas des comportements anormaux (agressifs et transgressifs) de tout type de bateau à savoir, ceux qui circulent librement dans le port. Un comportement de bateau ou *scénario* a été préalablement défini comme étant un système réactif synchrone programmé en ESTEREL. Ce scénario prend en entrée un ensemble de constats instantanés construit à partir des pistes systèmes et fournit un niveau d'alarme décrivant le niveau de reconnaissance du scénario en cours. Ces constats qui constituent les entrées du BRAM (constats instantanés) sont plus riches que les BPAM (pistes système) puisqu'elles concernent des informations précises sur le bateau (le type de bateau, le type de sa cargaison, etc), ce qui rendra plus fiable le niveau de reconnaissance du scénario de comportement ; par exemple, « un changement brutal de vitesse » est intégré dans le scénario construit par l'approche réactive alors qu'il n'était pas pris en compte par les MMC qui se basent uniquement sur les positions pour le processus de reconnaissance.

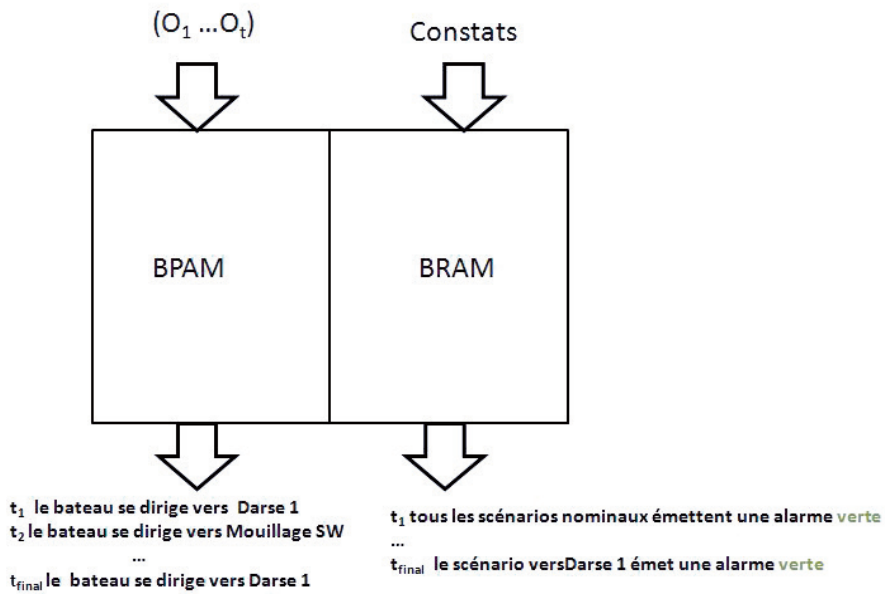
Par ailleurs, nous avons également construit des scénarios nominaux en utilisant l'approche réactive synchrone mais les résultats n'étaient pas très concluants. En effet, quant le bateau commercial est dans une zone commune du port (zone A_1 par exemple) à tous les scénarios de comportements, tous les scénarios envoient l'alarme *matching* et donc le bateau a exactement les mêmes chances d'effectuer n'importe quel comportement préalablement défini. Néanmoins, pour le même bateau dans la zone A_1 , l'utilisation de nos MMC permet de donner une approximation, bien qu'elle ne soit pas toujours correcte, du comportement le plus probable.

Pour conclure, les deux modules BPAM et BRAM sont complémentaires pour analyser et reconnaître les comportements de bateau dans le port. Le BPAM a montré des résultats intéressants pour la reconnaissance de comportements nominaux alors que le BRAM s'est montré pertinent pour le traitement des comportements transgressifs et agressifs.

Pour récapituler la complémentarité des deux modules, nous regarderons la mise en parallèle du BPAM et du BRAM pour deux exemples de comportements d'un navire commercial (rappelons que le BPAM n'est effectif que pour les gros bateaux). Pour les deux exemples suivants, nous considérerons que l'IHM affichera trois types d'alarmes : une alarme verte pour un comportement non menaçant, une alarme rouge pour un comportement suspect et une alarme grise pour un comportement inconnu.



(a) trajectoire d'un bateau se dirigeant vers Lavera en passant par les zones : A_1, A_2, A_3, B et *Lavera*

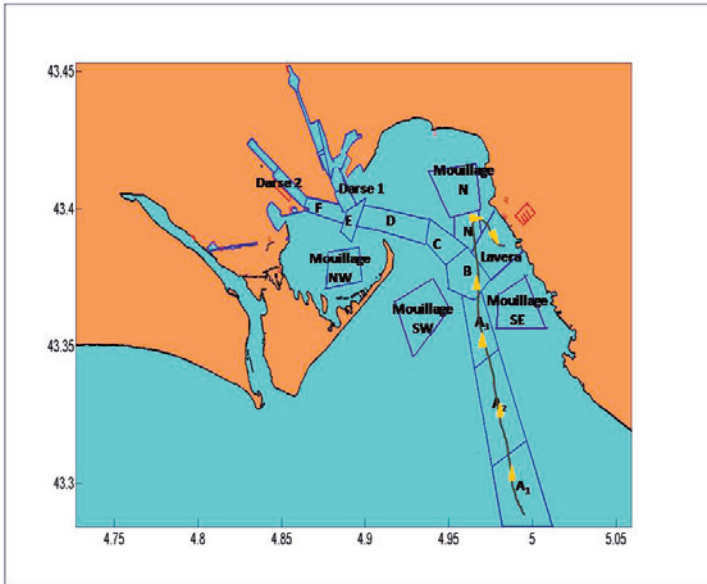


(b) Fonctionnement du BAM associé à (4.20a)

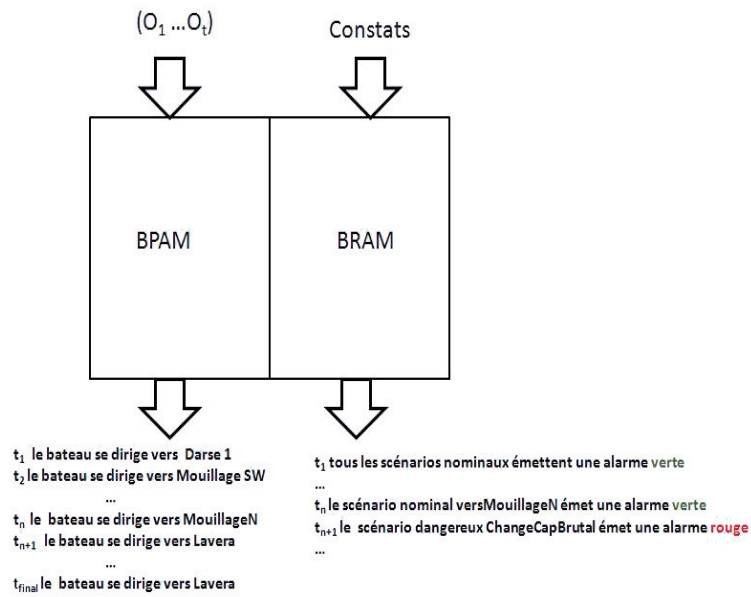
FIGURE 4.20 – Comportement habituel d'un navire se dirigeant vers Lavera.

Comportement nominal de bateau qui se dirige vers Darse1 (voir figure (4.20)) Dans le cas d'un comportement nominal de bateau qui se dirige vers Lavera. Le BRAM et le BPAM vont fonctionner en parallèle de la manière suivante :

- Quand le bateau passe par les zones communes à tous les comportements (zone A_1 , zone A_2 , zone A_3) :
 - Le BPAM renvoie à chaque pas temporel une estimation du comportement le plus probable en cours en calculant $\arg \max \{ \log P(O_1 \dots O_t / \lambda_i) \}_{1 \leq i \leq n}$ avec $n=7$ le nombre total de comportements de bateaux dans le port.
 - Tous les scénarios nominaux du BRAM renvoient une alarme de couleur verte pour exprimer que tous les scénarios de comportements de ce bateau sont possibles. Les autres scénarios (transgressifs et agressifs) émettent des alarmes grises.
- Quand le bateau dépasse la zone B :
 - De la même manière, le BPAM nous retourne toujours une estimation du comportement le plus probable qui est celui d'un bateau qui se dirige vers Lavera.
 - Pour le BRAM, les scénarios de comportements nominaux suivants (vers Lavera, vers MouillageN, versDarse1, versDarse2, versMouillageNW) émettent des alarmes vertes alors que tous les autres vont émettre des alarme grises.
- Quand le bateau est dans la zone Lavera :
 - Le scénario ESTEREL vers Lavera du BRAM est le seul à émettre une alarme verte.
 - Dans le BPAM, le log-vraisemblance $\log P(O_t / \lambda_3) \geq \log P(O_t / \lambda_i)_{1 \leq i \leq 7, i \neq 3}$ où le λ_3 est associé au comportement de bateau se dirigeant vers Lavera.



(a) trajectoire d'un bateau qui passe par A_1, A_2, A_3, B, N et *Lavera*



(b) Fonctionnement du BAM associé à (4.21a)

FIGURE 4.21 – Comportement anormal d'un bateau qui se dirige vers *Mouillage N* puis change brutalement de cap vers *Lavera*.

Comportement anormal de bateau (cas de la figure (4.21)) Dans le cas d'un comportement anormal d'un bateau qui se dirige vers MouillageN puis change brutalement de cap pour aller vers Lavera :

- Les deux modules BRAM et le BPAM fonctionnent de la même manière que le cas de l'exemple 1 quand le bateau passe respectivement par les zones A_1 jusqu'à la zone N .
- Quand le bateau quitte la zone N :
 - Le BPAM nous renvoie une estimation du comportement le plus probable.
 - Le scénario ESTEREL dangereux ChangeCapBrutal du BRAM renvoie une alarme rouge et tous les autres modules émettent des alarmes grises.

Dans ce cas, le BPAM reconnaît un changement de trajectoires de bateaux et le BRAM complétera cette information en nous indiquant un changement brutal du cap du navire lors de son passage de la zone N à la zone *Lavera*. Le comportement global de ce navire sera considéré comme dangereux par le BAM.

Comparaison des deux méthodes Nous avons récapitulé brièvement dans le tableau (4.4), les résultats obtenus par les deux modules. Dans le tableau (4.5), nous avons évalué les points forts et les points faibles obtenus par les deux approches ayant servies à construire les deux sous-modules du BAM.

	Méthode	Résultats		
		Scénarios nominaux	Scénarios agressifs	Scénarios transgressifs
Reconnaissance par approche probabiliste (BPAM)	MMC	intéressants	non-performants	non-performants
Reconnaissance par approche formelle (BRAM)	Le langage réactif synchrone ESTEREL	non-performants	intéressants	intéressants

TABLE 4.4 – Récapitulation des différents résultats obtenus par les BPAM et le BRAM.

	Points forts	Points faibles
BPAM par MMC	<ul style="list-style-type: none"> – précision – robustesse des modèles de comportements nominaux. – estimation probabiliste. – apprentissage rendant le modèle plus précis. 	<ul style="list-style-type: none"> – fausses alertes. – complexité des algorithmes des MMC.
BRAM par approche réactive synchrone	<ul style="list-style-type: none"> – rapidité. – richesse des informations en entrée. – richesse des comportements anormaux définis. – définition directe des scénarios sans apprentissage. – implémentation simple. 	<ul style="list-style-type: none"> – fausses alertes. – modélisation déterministe.

TABLE 4.5 – Points forts et points faibles des deux sous-modules : BPAM et BRAM

Conclusion et perspectives

Conclusion

Actuellement et dans la plupart des ports mondiaux, les responsables de la sécurité portuaire utilisent les données qui s'affichent à l'IHM et qui sont issues de plusieurs systèmes de détection pour analyser et reconnaître les comportements de tous les navires dans le port. Nous avons alors conçu le BAM qui est un système d'analyse automatique de ces comportements afin d'aider le personnel du GPMM à identifier tout comportement suspect.

En observant l'évolution des différents navires dans le port, nous avons classé leur mode de déplacement en deux catégories : les navires commerciaux et à cargaisons dangereuses se déplacent de manière régulière alors que les petits navires ont des déplacements non-réguliers.

Par conséquent, notre BAM a été conçu en deux sous-modules complémentaires, afin de traiter séparément ces deux catégories de comportements. Il s'agit du BPAM et du BRAM.

Le BPAM construit à partir de l'approche probabiliste MMC permet de prédire les comportements nominaux des navires commerciaux et ceux à cargaisons dangereuses. Ces bateaux se déplacent de façon récurrente car ils doivent suivre des routes imposées par la législation, ce qui a permis l'apprentissage de leurs comportements. En revanche, d'autres types de bateaux tels que les petits bateaux ne sont pas restreints à suivre des routes et peuvent ainsi circuler librement et aléatoirement dans le port. Cette catégorie de comportements a été traitée par le BRAM qui a été construit par le langage réactif synchrone ESTEREL. Contrairement au BPAM qui est fondé sur une phase d'apprentissage hors-ligne des modèles de comportements obtenus à partir des pistes systèmes, le BRAM traite ces pistes en temps-réel en les convertissant en informations logiques dites *constats*.

Dans le cas du BPAM, les zones fonctionnelles formant les routes, les positions issues des pistes systèmes ainsi que les algorithmes des MMC ont permis de construire

les modèles probabilistes associés à tous comportements nominaux de navires. Par la suite, par un calcul probabiliste, le BPAM compare en temps-réels les valeurs des log-vraisemblances des positions par rapport à tous les modèles et ce pour chaque pas temporel. Le comportement le plus probable est celui ayant donné la valeur la plus élevée du log-vraisemblance, contrairement au comportement anormal qui peut-être défini statistiquement par un comportement improbable, c'est-à-dire, celui ayant donné de très faibles valeurs de vraisemblances calculées à partir de modèles probabilistes. Nous avons également défini un comportement anormal d'un navire comme étant celui qui change de trajectoires, c'est-à-dire celui qui possède des valeurs de vraisemblances très variables.

Néanmoins, cette définition probabiliste d'un comportement anormal nous a paru insuffisante. D'où le rôle du BRAM, qui s'est montré complémentaire au BPAM pour traiter les comportements suspects de gros et de petits bateaux. Nous avons intégré dans le BRAM deux types de comportements anormaux de navires : les comportements transgressifs qui ne respectent pas les règles de la navigation et les comportements agressifs qui sont dangereux. Ce module prend en entrée l'ensemble des constats à partir desquels nous avons conçu un ensemble de modèles de comportements suspects dits *scénarios*. Il s'agit de modules programmés en ESTEREL qui renvoient des alarmes traduisant des niveaux de reconnaissance en temps-réel des comportements des bateaux au cours de leur évolution dans le GPMM. En effet, pour un même bateau et à chaque pas temporel, tous les modules sont actifs et renvoient des niveaux d'alarmes correspondant aux niveaux de reconnaissance. L'ensemble de ces alarmes a été fusionné afin d'obtenir une alarme unique correspondant à un seul niveau de reconnaissance. Notons aussi que nous avons intégré dans le BAM des scénarios nominaux de comportements de navires.

La mise en parallèle de ces deux modules, dans le cas de la reconnaissance en temps-réel des comportements de gros bateaux, a permis de tirer les deux conclusions suivantes :

- Dans le cas d'un comportement nominal de navires appris par MMC, les résultats de la reconnaissance par le BPAM se sont avérés plus intéressants que ceux du BRAM. En effet, si le navire est dans des zones communes à tous les modèles de comportements, le BPAM permet une prédiction de son comportement au cours de son évolution dans le port alors que le BRAM donne tous les comportements vraisemblables.
- Dans le cas d'un comportement anormal d'un navire, la reconnaissance sera principalement donnée par le BRAM sauf dans certains cas tels que le changement brutal de cap dans des zones du port, où la reconnaissance par le BPAM donnera un changement de trajectoires qui sera exprimé par des changement en

valeur des log-vraisemblances. Dans ce cas, cette description de changement de trajectoires viendra consolider le scénario de changement brutal de cap obtenu par le BRAM.

Notons également, que dans le cas des petits embarcations à comportements aléatoires, la reconnaissance sera intégralement effectuée par le BRAM.

De cette manière, l'utilisation des deux approches, probabiliste et logique, nous a permis de bénéficier de leurs avantages respectifs. L'approche réactive synchrone apporte une richesse d'expression des informations sur les navires en convertissant en temps-réel les pistes systèmes en événements logiques. D'où la richesse des scénarios de comportements construits qui s'exécutent en renvoyant instantanément des niveaux de reconnaissance sous forme d'alarmes. Cette rapidité est assurée grâce à l'hypothèse synchrone de cette approche. L'approche probabiliste par MMC, pour sa part, permet de construire les modèles de comportements nominaux de navires grâce à l'apprentissage qui donne des modèles précis. Cette précision dépend de la taille de la base de donnée utilisées et contribue fortement aux résultats de la reconnaissance. La phase d'apprentissage qui fait intervenir, à la fois, nos connaissances (les zones fonctionnelles) et de vraies informations sur les navires (les pistes systèmes) permet de construire ces modèles de comportements tout en minimisant les incertitudes entre les modèles réels et les modèles construits.

Perspectives

Une nouvelle conception du BAM

Bien que nous ayons opté pour l'utilisation de deux approches différentes pour la construction de notre module, le choix de plusieurs chercheurs d'un raisonnement purement probabiliste dans le domaine de la reconnaissance de comportement nous a laissé perplexe. A ce stade de nos travaux, il nous a paru possible de construire notre BAM entièrement par approche probabiliste. En effet, en regardant de plus près la structure des nos scénarios de comportements construits par ESTEREL, nous avons constaté que la relation qui existe entre les *constats* (*est_en zone interdite*, *va_trop_vite*, etc) et les *alarmes* (*matching*, *running*, etc) est une relation de *cause* à *effet* : **await** *est_en_zone interdite* ; **emit** *standby* ce qui est proche de l'algorithme : **Si** *est_en_zone interdite* **alors** *standby*, cette relation de causalité nous a fait penser aux Réseaux Bayésiens.

Si nous considérons l'algorithme du comportement agressif (4.1), il peut parfaitement être décrit par le graphe causal de la figure (4.22). Nous cherchons dans ce cas à déterminer $P(\textit{matching})$.

Algorithme 4.1 Comportement agressif

```
await est_en_zone_interdite ;  
weak abort  
await change_cap_brutal ;  
emit matching ;  
...  
when [not est_en_zone_interdite]
```

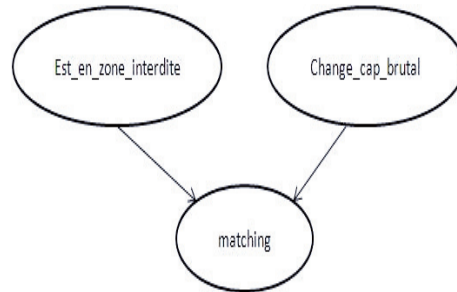


FIGURE 4.22 – Graphe causal associé à 4.1

L'utilisation des RB permettra de bénéficier des avantages précédemment cités à la fois des MMC et de l'approche réactive, et ceci en utilisant l'apprentissage, connu pour donner des modèles assez précis, sur les constats extraits des pistes systèmes qui sont riches en informations sur les navires.

Par ailleurs, puisque notre modèle de comportement évolue dans le temps, nous aurons probablement besoin du formalisme RBD.

A la fin de cette phase de nos travaux, la piste des RB nous a paru très intéressante à exploiter dans le cadre de travaux à venir.

Généralisation du BAM

Le BAM que nous avons conçu peut-être généralisé à plusieurs ports mondiaux. Dans notre cas, nous avons utilisé la réglementation (les chenaux d'accès et les zones fonctionnelles) du port pour construire ce module. Néanmoins et sachant qu'on n'a pas forcément accès à cette réglementation dans tous les ports mondiaux, il est alors indispensable d'effectuer des réglages sur les paramètres de notre module. De plus, il est possible d'utiliser uniquement les données AIS (qui sont actuellement les seules données disponibles dans tous les ports mondiaux) comme données d'entrées du BAM. Dans le cas du BRAM, les constats seront fabriqués à partir des données

AIS. Pour le BPAM, étant donné que les zones du port qui étaient délimitées par la législation ont été utilisées pour initialiser les MMC associés aux comportements de bateaux dans le port, il est possible de les estimer en utilisant les trajectoires de bateaux obtenues à partir des données AIS. Dans [77], les auteurs ont utilisé l'historique des données AIS (voir figure (4.23)) afin de construire les chenaux d'accès des navires au rade de Brest en délimitant les vrais couloirs de navigation. Ces couloirs ont été obtenus en calculant des densités sur les trajectoires de navires (voir figure (4.24)). La densité du Kernel est l'une des méthodes utilisées dans cet article pour l'estimation des densités de trajectoires. Une densité de kernel est une estimation de la densité de probabilité d'une variable aléatoire. Si on considère un ensemble de variables aléatoires indépendantes et identiquement distribuées $(x_1 \dots x_N)$, l'estimateur par la méthode du noyau de kernel de cette densité de probabilité s'écrit :

$$\hat{f}_h(x) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{x - x_i}{h}\right)$$

où K est un noyau et h est un paramètre nommé fenêtre, qui régit le degré de lissage de l'estimation. K est souvent choisi comme étant la densité d'une fonction gaussienne standard (espérance nulle et variance unitaire) :

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

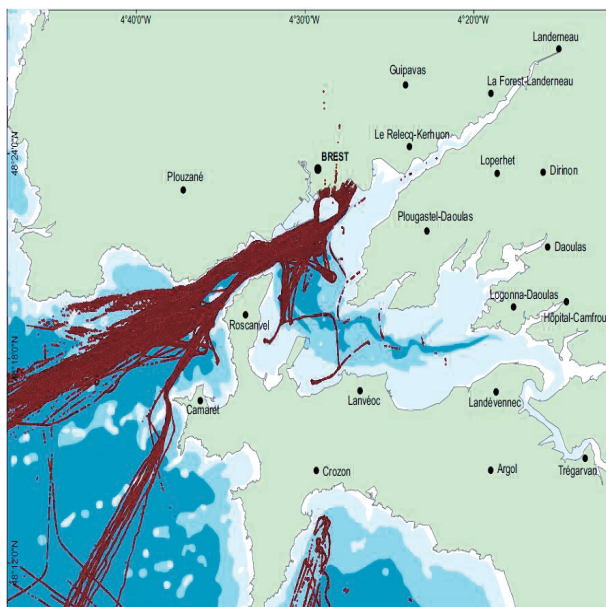
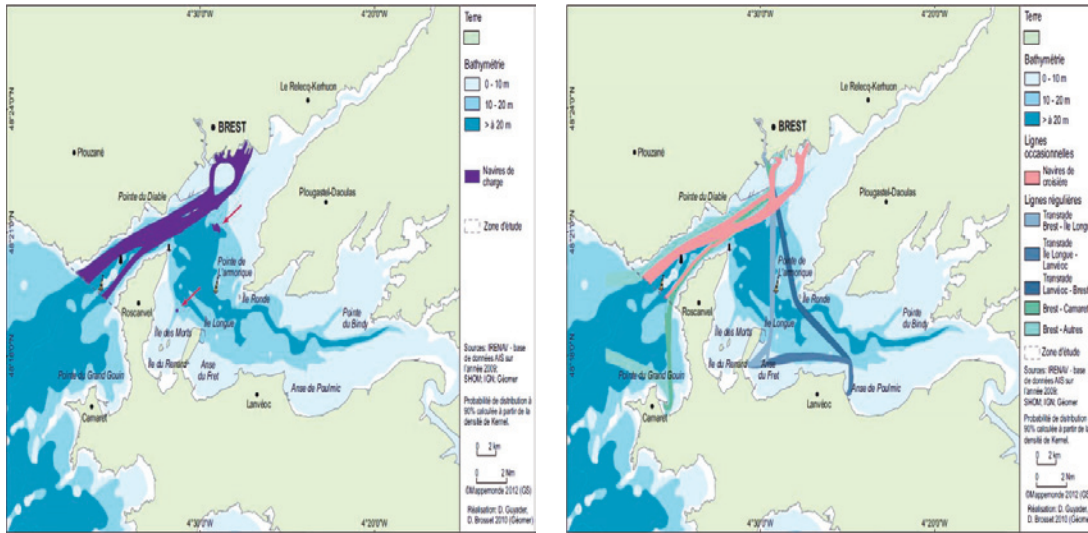


FIGURE 4.23 – Trajectoires de bateaux reconstruites à partir de données AIS pour l’année 2009 [77].

Par cette méthode, les auteurs ont pu calculer les couloirs de navigation pour les navires de charges et les navires de passagers (voir figure (4.24)).

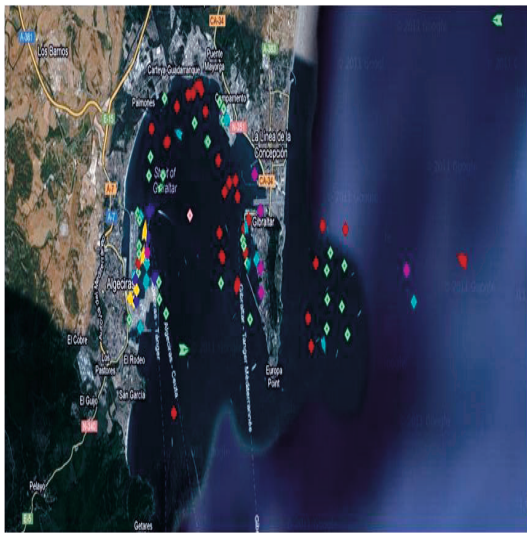
L’approche probabiliste MMC ainsi que l’approche réactive synchrone ESTEREL peuvent également être utilisées pour sécuriser le domaine aéroportuaire (voir figure (4.25b)). Étant donné que la structure du domaine portuaire et du domaine aéroportuaire ainsi que le comportement d’un avion et d’un bateau (voir figure (4.25a)) sont comparables, il est possible de définir un comportement normal ou anormal d’un avion de la même manière que pour les bateaux . En effet, le comportement nominal d’un avion est celui qui suit correctement sa route (son plan de vol) et le comportement anormal peut se caractériser par un changement brutal de cap, un abordage d’un autre avion ou encore un déplacement vers des zones interdites. De cette manière, notre BAM serait peut-être utile pour les contrôleurs aériens.



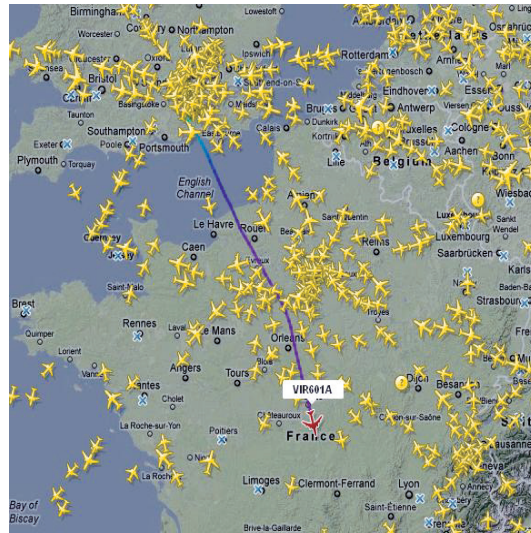
(a) Couloirs de navigation pour les navires de charges (2009)

(b) Couloirs de navigation pour les navires de passagers(2009)

FIGURE 4.24 – Les couloirs de navigations des navires de charges et des navires de passagers obtenus par le calcul de la densité du Kernel [77].



(a) Domaine portuaire



(b) Domaine aéroportuaire

FIGURE 4.25 – Les domaines portuaires et aéroportuaires

Annexe A

Espace de probabilité (Ω, \mathcal{B}, P) La modélisation de calcul des probabilités été inventée par Kolmogorov dans les années 30. Cette modélisation a été faite à partir de trois objets :

- Ω est l'espace des observables. En effectuant une expérience, on sélectionne par un procédé quelconque un élément ω dans un ensemble Ω : jeter un dé revient à sélectionner un élément de $\Omega = \{1, 2, 3, 4, 5, 6\}$.
- \mathcal{B} est la tribu des événements : on choisit un sous-ensemble A de l'espace des observables et on cherche à savoir si le résultat d'une expérience ω va tomber dans Ω .
- P est une application de \mathcal{B} dans $[0, 1]$ qui associe à tout événement B un nombre $P(B)$ compris entre 0 et 1 appelé probabilité de B .

Processus stochastique Un processus stochastique est une famille de variables aléatoires $X = (X_1 \dots X_t)$ définie sur un espace de probabilité (Ω, \mathcal{B}, P) et qui évoluent dans le temps.

L'hypothèse de Markov d'ordre 1 Elle considère qu'au cours de son évolution chaque variable aléatoire, à un instant t , dépend uniquement de sa variable aléatoire à l'instant $t-1$.

Autrement dit, pour un ensemble de variables aléatoires $(X_1 \dots X_t)$ qui évoluent dans le temps avec $t \in [1, T]$, l'hypothèse de Markov d'ordre 1, nous permet d'écrire : $P(X_t / X_{t-1} \dots X_1) = P(X_t / X_{t-1})$.

Processus stationnaire Un processus stationnaire est un processus qui a des propriétés statistiques (caractérisées par son espérance mathématiques) qui sont indépendantes du temps.

Loi conjointe La loi conjointe est une loi à plusieurs variables qui s'écrit $P(X_1 \dots X_t)$ et permet de calculer la probabilité que cet ensemble de variable aléatoire se produise en même temps. Si on considère le cas de deux variables aléatoires (X_1, X_2) , la loi conjointe sur ces variables aléatoires s'écrit $P(X_1, X_2) = P(X_1 = s_1, X_2 = s_2)$ avec $(s_1, s_2) \in \Omega$.

Loi marginale La donnée d'une probabilité conjointe d'un ensemble de variables aléatoires $P(X_1 \dots X_t)$ permet de retrouver la probabilité jointe de chacun de ses sous-ensembles : $P(X_1)$, $P(X_1 X_2)$, il s'agit de la probabilité marginale qui s'écrit :

$$\forall x \in X, P(X) = \sum_{y \in Y} P(x, y)$$

Règle de Bayes Etant donné deux événements A et B , le théorème de Bayes s'écrit :

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}$$

avec $P(A)$ dite la probabilité à priori de A , $P(A/B)$ est appelée la probabilité à posteriori de A sachant B , $P(B/A)$ est la fonction de vraisemblance et $P(B)$ est la probabilité marginale à priori de B .

Annexe B

La fonction de vraisemblance Soit $X = (x_1, \dots, x_n)$ un ensemble de variables aléatoires indépendantes et identiquement distribuées. On veut estimer le paramètre θ associé à cet ensemble de variables aléatoires. On appelle la *vraisemblance de θ* au vu des observations (x_1, x_2, \dots, x_n) la fonction :

$$L(\theta) = L(x_1, \dots, x_n/\theta) = \prod_{i=1}^n p(x_i/\theta) = P(X/\theta) \quad (4.9)$$

Le maximum de vraisemblance (MV) est obtenu pour un θ_{opt} :

$$\theta_{opt} = \arg \max_{\theta} L(\theta) \quad (4.10)$$

Généralement, on maximise le $\log L(\theta)$ qui est plus facile à calculer :

$$\theta_{opt} = \arg \max_{\theta} (\log L(\theta)) \quad (4.11)$$

Espérance-Maximisation Dans le cas où les variables aléatoires dépendent d'autres types de variables non-observables, le calcul analytique de la formule (4.11) serait très compliquée. L'algorithme Espérance-Maximisation (EM) est alors une méthode statistique itérative qui permet d'estimer le MV des paramètres d'une distribution qui dépend de variables cachées non-observables. Nous nous intéresserons par la suite à cet algorithme.

On a :

$$L(\theta) = \log(P(X/\theta)) \quad (4.12)$$

On considère qu'après n itérations, l'estimation courante pour θ est donnée par θ_n . Sachant que le \log est une fonction croissante et qu'on veut trouver θ pour lequel L est maximale, on veut calculer une mise à jour de θ tel que :

$$L(\theta) \succ L(\theta_n) \quad (4.13)$$

est vérifiée pour chaque itération.

Ceci revient à maximiser la différence :

$$L(\theta) - L(\theta_n) = \log P(X/\theta) - \log P(X/\theta_n) \quad (4.14)$$

Sachant que les variables aléatoires (x_1, \dots, x_n) sont observables et qu'elles dépendent d'un ensemble caché de variables aléatoires $Z = (z_1, \dots, z_n)$, $P(Z/\theta)$ s'exprime en fonction de la loi marginale :

$$P(X/\theta) = \sum_z P(X/z, \theta)P(z/\theta) \quad (4.15)$$

En remplaçant (4.15) dans (4.14), on obtient :

$$L(\theta) - L(\theta_n) = \log \sum_z P(X/z, \theta)P(z/\theta) - \log P(X/\theta_n) \quad (4.16)$$

D'après l'inégalité de Jensen :

$$\log \sum_{i=1}^n \lambda_i x_i \geq \sum_{i=1}^n \lambda_i \log(x_i) \quad (4.17)$$

pour des constantes $\lambda_i \geq 0$ et $\sum_{i=1}^n \lambda_i = 1$, l'équation (4.16) s'écrit :

$$L(\theta) - L(\theta_n) = \log \left(\sum_z P(X/z, \theta)P(z/\theta) \right) - \log P(X/\theta_n) \quad (4.18)$$

$$= \log \left(\sum_z P(X/z, \theta)P(z/\theta) \frac{P(z/X, \theta_n)}{P(z/X, \theta_n)} \right) - \log P(X/\theta_n) \quad (4.19)$$

$$= \log \left(\sum_z P(z/X, \theta_n) \frac{P(X/z, \theta)P(z/\theta)}{P(z/X, \theta_n)} \right) - \log P(X/\theta_n) \quad (4.20)$$

$$\geq \sum_z P(z/X, \theta_n) \log \left(\frac{P(X/z, \theta)P(z/\theta)}{P(z/X, \theta_n)} \right) - \log P(X/\theta_n) \quad (4.21)$$

$$= \sum_z P(z/X, \theta_n) \log \left(\frac{P(X/z, \theta)P(z/\theta)}{P(z/X, \theta_n)P(X/\theta_n)} \right) \quad (4.22)$$

$$= \Delta(\theta, \theta_n) \quad (4.23)$$

Pour passer de (4.22) à (4.23), on a utilisé $\sum_z P(z/X, \theta_n) = 1$.

Ainsi $\sum_z P(z/X, \theta_n) \log P(X/\theta_n) = \log P(X/\theta_n)$.

De ce qui précède, on écrit :

$$L(\theta) \geq L(\theta_n) + \Delta(\theta, \theta_n) \quad (4.24)$$

et on définit :

$$l(\theta/\theta_n) = L(\theta_n) + \Delta(\theta, \theta_n) \quad (4.25)$$

L'équation (4.24) s'écrit :

$$L(\theta) \geq l(\theta/\theta_n) \quad (4.26)$$

Notre objectif est de choisir θ tel que $L(\theta)$ est maximisée. D'après (4.26), $l(\theta/\theta_n)$ est borné par la fonction de vraisemblance et $L(\theta) = l(\theta/\theta_n)$ pour $\theta = \theta_n$. Ainsi toute valeur de θ qui va augmenter $L(\theta)$ va augmenter $l(\theta/\theta_n)$.

Dans le but de trouver la meilleure θ faisant croître $L(\theta)$, on cherchera celle qui maximise $l(\theta/\theta_n)$.

Soit θ_{n+1} la valeur de la mise à jour.

$$\theta_{n+1} = \arg \max_{\theta} \{l(\theta/\theta_n)\} \quad (4.27)$$

$$= \arg \max_{\theta} \{L(\theta_n) + \Delta(\theta, \theta_n)\} \quad (4.28)$$

$$= \arg \max_{\theta} \left\{ L(\theta_n) + \sum_z P(z/X, \theta_n) \log \left(\frac{P(X/z, \theta)P(z/\theta)}{P(z/X, \theta_n)P(X/\theta_n)} \right) \right\} \quad (4.29)$$

or $L(\theta_n)$ est une constante.

$$= \arg \max_{\theta} \left\{ \sum_z P(z/X, \theta_n) \log \left(\frac{P(X/z, \theta)P(z/\theta)}{P(z/X, \theta_n)P(X/\theta_n)} \right) \right\} \quad (4.30)$$

$$= \arg \max_{\theta} \left\{ \sum_z P(z/X, \theta_n) \log (P(X/z, \theta)P(z/\theta)) \right\} \quad (4.31)$$

$$= \arg \max_{\theta} \left\{ \sum_z P(z/X, \theta_n) \log \frac{P(X, z, \theta)}{P(z, \theta)} \frac{P(z, \theta)}{P(\theta)} \right\}$$

$$= \arg \max_{\theta} \left\{ \sum_z P(z/X, \theta_n) \log P(X, z/\theta) \right\} \quad (4.32)$$

$$= \arg \max_{\theta} \{E_{Z/X, \theta_n} \{\log P(X, z/\theta)\}\} \quad (4.33)$$

$$= \arg \max_{\theta} \{Q(\theta, \theta_n)\} \quad (4.34)$$

L'algorithme EM se résume en ces deux étapes suivantes :

1. *Etape – E* : Evaluer l'espérance : $Q(\theta, \theta_n) = E_{Z/X, \theta_n} \{\log P(X, z/\theta)\}$
2. *Etape – M* : Maximiser Q par rapport à θ : $\theta_{n+1} = \arg \max_{\theta} \{Q(\theta, \theta_n)\}$

Pour récapituler sur l'utilisation de l'EM, il s'agit d'une méthode statistique itérative qui permet de calculer le maximum de vraisemblance d'une fonction à variables observables $X = (x_1, \dots, x_n)$ dépendant de variables cachées $Z = (z_1, \dots, z_n)$.

Application de l'algorithme EM au MMC Rappelons qu'un modèle λ est défini par trois paramètres π , A et B . Le modèle λ est celui qu'on cherche à optimiser dans la vraisemblance $P(O/\lambda)$.

Notations : On appellera $O = (O_1, \dots, O_T)$ la série de variables d'observations et $X = (X_1, \dots, X_T)$ la série de variables non observables (variables d'état) qui évolue dans le temps.

Cas discret des MMC

Dans le cas discret d'un modèle de Markov caché, l'état et l'observation sont discrets.

Pour un modèle λ donné et à partir d'une série d'observation $O = (O_1, \dots, O_T)$, on veut trouver le modèle λ optimal ayant donné cette série d'observation (problème 3 des MMC).

$$\lambda_{opt} = \arg \max_{\lambda} P(O/\lambda) \quad (4.35)$$

$$= \arg \max_{\lambda} \sum_q P(O, q/\lambda) \quad (4.36)$$

En utilisant l'équation Q définie par Baum et al.[11], on a :

$$Q(\lambda, \lambda_{opt}) = \sum_{q \in S} \log P(O, q/\lambda) P(q/O, \lambda_{opt}) \quad (4.37)$$

On a :

$$P(O, X/\lambda_{opt}) = P(O_0 \dots O_T X_0 \dots X_T/\lambda_{opt}) \quad (4.38)$$

$$= \pi_{X_1} \prod_{t=1}^{T-1} a_{X_t X_{t+1}} \prod_{t=1}^T b_{X_t}(O_t) \quad (4.39)$$

La fonction Q s'écrit :

$$Q(\lambda, \lambda_{opt}) = \sum_{q \in S} \log \pi_{X_0} P(X/O, \lambda_{opt}) + \sum_{q \in S} \left(\sum_{t=1}^{T-1} \log a_{X_t X_{t+1}} \right) P(X/O, \lambda_{opt}) \quad (4.40)$$

$$+ \sum_{X \in S} \left(\sum_{t=1}^T \log b_{X_t}(O_t) \right) P(X/O, \lambda_{opt}) \quad (4.41)$$

$$= Q(\lambda, \pi_{opt}) + Q(\lambda, A_{opt}) + Q(\lambda, B_{opt}) \quad (4.42)$$

La fonction Q a été décomposée en 3 termes qu'on va utiliser séparément pour estimer les 3 paramètres d'une MMC. Le multiplicateur de Lagrange¹ permettra

1. La méthode des multiplicateurs de Lagrange permet de trouver des points stationnaires (maximum, minimum,...) d'une fonction dérivable d'une ou de plusieurs variables sous contraintes.

de calculer les trois paramètres optimaux π_{opt} , A_{opt} et B_{opt} en définissant les trois fonctions L_1 , L_2 et L_3 suivantes :

$$L_1(\pi_{opt}, \gamma) = Q(\lambda, \pi_{opt}) + \gamma \left(\sum_{i=1}^N \pi_i - 1 \right) \quad (4.43)$$

$$L_2(A_{opt}, \gamma) = Q(\lambda, A_{opt}) + \gamma \left(\sum_{j=1}^N a_{ij} - 1 \right) \quad (4.44)$$

$$L_3(B_{opt}, \gamma) = Q(\lambda, B_{opt}) + \gamma \left(\sum_{k=1}^M b_{ik} - 1 \right) \quad (4.45)$$

qu'on dérive respectivement par rapport à π_i , A_i et B_i puis par rapport au multiplicateur de Lagrange γ pour trouver ces paramètres optimaux sous les contraintes (4.46), (4.47) et (4.48) suivantes :

$$\sum_{i=1}^N \pi_i = 1 \quad (4.46)$$

$$\sum_{j=1}^N a_{ij} = 1 \quad (4.47)$$

$$\sum_{k=1}^M b_{ik} = 1 \quad (4.48)$$

1er terme : Pour

$$Q(\lambda, \pi_{opt}) = \sum_{X \in S} \log \pi_{X_1} P(X/O, \lambda_{opt}) \quad (4.49)$$

$$= \sum_{i=1}^N \log \pi_i P(X_1 = S_i/O, \lambda_{opt}) \quad (4.50)$$

on trouve les paramètres optimaux du terme d'initialisation :

$$\Rightarrow \bar{\pi}_i = P(X_1 = S_i/O, \lambda_{opt}) \quad (4.51)$$

$$= \gamma_1(i) \quad (4.52)$$

2eme terme : Pour :

$$Q(\lambda, A_{opt}) = \sum_{q \in S} \left(\sum_{t=1}^{T-1} \log a_{X_t X_{t+1}} \right) P(X/O, \lambda_{opt}) \quad (4.53)$$

$$= \sum_{i=1}^N \sum_{j=1}^N \sum_{t=1}^{T-1} \log a_{ij} P(X_t = i, X_{t+1} = j/O, \lambda_{opt}) \quad (4.54)$$

on trouve les paramètres optimaux du terme de transition.

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} P(X_t = i, X_{t+1} = j/O, \lambda_{opt})}{\sum_{t=1}^{T-1} P(X_t = i/O, \lambda_{opt})}, \quad 1 \leq i \leq N, 1 \leq j \leq N \quad (4.55)$$

$$= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (4.56)$$

3eme terme : Pour

$$Q(\lambda, B_{opt}) = \sum_{q \in S} \left(\sum_{t=1}^T \log(b_{X_t}(o_t)) \right) P(X/O, \lambda_{opt}) \quad (4.57)$$

$$= \sum_{i=1}^N \sum_{t=1}^T \log(b_i(o_t)) P(X_t = i/O, \lambda_{opt}) \quad (4.58)$$

on trouve les paramètres optimaux du terme d'émission.

$$\log(b_i(o_t)) = \sum_{k=1}^M \log b_{ik} \delta_{k o_t} \quad (4.59)$$

On trouve :

$$\bar{b}_{ik} = \frac{\sum_{t=1}^T P(X_t = i/O, \lambda_{opt}) \delta_{k, o_t}}{\sum_{t=1}^T P(X_t = i/O, \lambda_{opt})}, \quad 1 \leq i \leq N, 1 \leq k \leq M \quad (4.60)$$

$$= \frac{\sum_{t=1, o_t=v_k}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad (4.61)$$

Cas continu des MMC Dans le cas continu d'un modèle de Markov caché, l'état est discret et l'observation est continue.

Selon [110], certaines restrictions doivent être imposées sur la densité de probabilité des observations connaissant les états pour que l'estimation des paramètres soit faite de manière efficace. La représentation la plus générale de cette densité est une somme de lois multinormales de la forme :

$$b_i(O) = \sum_{m=1}^M c_{im} N(\mu_{im}, U_{im}; O) \quad (4.62)$$

où

$$N(\mu_{im}, U_{im}; O) = \frac{1}{(2\pi)^{\frac{N}{2}} |U_{im}|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (O - \mu_{im})^T U_{im}^{-1} (O - \mu_{im}) \right\} \quad (4.63)$$

et les coefficients c_{im} vérifient :

$$\sum_{m=1}^M c_{im} = 1, \quad 1 \leq i \leq N \quad (4.64)$$

$$c_{im} \geq 0, \quad 1 \leq i \leq N, 1 \leq m \leq M \quad (4.65)$$

L'estimation des paramètres d'initialisation (4.51) et de transition d'état (4.55) reste la même puisque leurs estimations ne dépendent pas du paramètre m représentant le numéro de la gaussienne dans la somme. Dans ce cas, l'écriture de Q est différente et les paramètres qu'on cherche à estimer sont $B_{opt} = (c_{opt}, \mu_{opt}, v_{opt})$:

$$Q(\lambda, \lambda_{opt}) = \sum_{q \in Q} \sum_{m \in M} \log P(O, q, m/\lambda) P(O, q, m/\lambda) \quad (4.66)$$

et donc

$$Q(\lambda, B_{opt}) = \sum_{i=1}^N \sum_{m=1}^M \sum_{t=1}^T \log b_{qt}(O_t, m_{qt}) P(q_t = i, c_{qt} = m/O, \lambda_{opt}) \quad (4.67)$$

Pour estimer le paramètre du terme d'émission B_{opt} dans le cas continu, on utilise toujours le multiplicateur de Lagrange γ sous la contrainte $\sum_{m=1}^M c_{im} = 1$:

$$L(B_{opt}, \gamma) = Q(\lambda, B_{opt}) + \gamma \left(\sum_{j=1}^N c_{im} - 1 \right) \quad (4.68)$$

puis pour i fixé, on dérive L par rapport à c_{im} :

$$\bar{c}_{im} = \frac{\sum_{t=1}^T P(q_t = i, c_{qt} = m/O, \lambda_{opt})}{\sum_{t=1}^T \sum_{m=1}^M P(q_t = i, c_{qt} = m/O, \lambda_{opt})} \quad (4.69)$$

$$= \frac{\sum_{t=1}^T \gamma_t(i, m)}{\sum_{t=1}^T \sum_{k=1}^M \gamma_t(i, m)} \quad (4.70)$$

– On dérive Q par rapport à μ_{im} puis par rapport à U_{im} et on trouve :

$$\bar{\mu}_{im} = \frac{\sum_{t=1}^T P(q_t = i, c_{qt} = m/O, \lambda_{opt}) O_t}{\sum_{t=1}^T P(q_t = i, c_{qt} = m/O, \lambda_{opt})} \quad (4.71)$$

$$= \frac{\sum_{t=1}^T \gamma_t(i, m) O_t}{\sum_{t=1}^T \gamma_t(i, m)} \quad (4.72)$$

$$\bar{U}_{im} = \frac{\sum_{t=1}^T P(q_t = i, c_{qt} = m/O, \lambda_{opt}) (O_t - \mu_{im}) (O_t - \mu_{im})^T}{\sum_{t=1}^T P(q_t = i, c_{qt} = m/O, \lambda_{opt})} \quad (4.73)$$

$$= \frac{\sum_{t=1}^T \gamma_t(i, m) (O_t - \mu_{im}) (O_t - \mu_{im})^T}{\sum_{t=1}^T \gamma_t(i, m)} \quad (4.74)$$

Bibliographie

- [1] J.K. Aggarwal and Q. Cai. "human motion analysis : A review". pages 90–102, 1997.
- [2] Jeffrey D. Ullman Alfred V. Aho, Ravi Sethi. *Compilers : Principles, Techniques, and Tools*. Addison Wesley Publishing Company, 1986.
- [3] J.R. Anderson. *The architecture of cognition*. Lawrence Erlbaum Assoc Inc, 1996.
- [4] C. André and M.A. Peraldi. Synchronous programming : Introduction and application to industrial process control. pages 461–470.
- [5] C. Bahlmann and H. Burkhardt. Measuring HMM similarity with the Bayes probability of error and its application to online handwriting recognition. pages 406–411, 2001.
- [6] A.V. Balakrishnan. *Kalman filtering theory*. Optimization Software, Inc., 1987.
- [7] Y. Bar-Shalom and X.R. Li. *Multitarget-multisensor tracking : principles and techniques*. 1995.
- [8] L.E. Baum. An equality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3 :1–8, 1972.
- [9] L.E. Baum and JA Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bull. Amer. Math. Soc*, 73(3) :360–363, 1967.
- [10] L.E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37(6) :1554–1563, 1966.
- [11] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1) :164–171, 1970.

- [12] L.E. Baum and G.R. Sell. Growth transformations for functions on manifolds. *Pacific Journal of Mathematics*, 27(2) :211–227, 1968.
- [13] Damien Belleville. Comportement maritime en haute mer : détection, intervention. Master’s thesis, 2008.
- [14] D. Bellot. Fusion de données avec des réseaux bayésiens pour la modélisation des systèmes dynamiques et son application en télémédecine. 2002.
- [15] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G.J. Pappas. Symbolic planning and control of robot motion [grand challenges of robotics]. *Robotics & Automation Magazine, IEEE*, 14(1) :61–70, 2007.
- [16] A. Benveniste and G. Berry. The synchronous approach to reactive and real-time systems. *Proceedings of the IEEE*, 79(9) :1270–1282, 1991.
- [17] A. Benveniste, P. Le Guernic, and C. Jacquemot. Synchronous programming with events and relations : the signal language and its semantics. volume 16, pages 103–149. Elsevier, 1991.
- [18] Albert Benveniste, Paul Caspi, Stefan A. Edwards, Nicolas Halbwachs, Paul Le Guernic, and Robert De Simone. The synchronous languages 12 years later. *Proceedings of the IEEE*, 91(1), January 2003.
- [19] Albert Benveniste, Michel Le Borgne, and Paul Le Guernic. Signal as a model for real-time and hybrid systems. Rapport de recherche RR-1608, INRIA, 1992.
- [20] G. Berry. The constructive semantics of pure esterel, 1996.
- [21] G. Berry. *The Esterel v5 Language Primer-Version 5.10, release 2.0*. Citeseer, 1998.
- [22] G. Berry, P. Couronne, and G. Gonthier. Synchronous programming of reactive systems : an introduction to ESTEREL. 1987.
- [23] G. Berry, P. Couronné, G. Gonthier, and J.P. BANATRE. Programmation synchrone des systèmes réactifs : le langage ESTEREL= Synchronous programming of reactive systems : the language ESTEREL. *TSI. Technique et science informatiques*, 6(4) :305–316, 1987.
- [24] G. Berry and G. Gonthier. The esterel synchronous programming language ; design, semantics, implementation. *Science of computer programming*, 19(2) :87–152, 1992.
- [25] Gérard Berry. The foundations of esterel. In *Proof, Language, and Interaction*, pages 425–454, 2000.
- [26] Gérard Berry. Synchronous programming techniques for embedded systems : Present and future. In *EMSOFT*, page 50, 2001.

- [27] Gérard Berry. Synchronous design and verification of critical embedded systems using scade and esterel. In *FMICS*, page 2, 2007.
- [28] Gérard Berry, Amar Bouali, Xavier Fornari, Emmanuel Ledinot, Eric Nassor, and Robert De Simone. ESTEREL : A formal method applied to avionic software development. *Science of Computer Programming*, 36(1) :Pages 5–25, December 2000.
- [29] Gérard Berry and Georges Gonthier. The esterel synchronous programming language : Design, semantics, implementation. *Sci. Comput. Program.*, 19(2) :87–152, 1992.
- [30] Gérard Berry, Michael Kishinevsky, and Satnam Singh. System level design and verification using a synchronous language. In *ICCAD*, pages 433–440, 2003.
- [31] P. Bessi ere. *Probability as an Alternative to Logic for Rational Sensory–Motor Reasoning and Decision*. Springer, 2008.
- [32] J.A. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4 :126, 1998.
- [33] S.S. Blackman. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine, IEEE*, 19(1) :5–18, 2004.
- [34] A.F. Bobick. Movement, activity and action : the role of knowledge in the perception of motion. *Philosophical Transactions of the Royal Society of London. Series B : Biological Sciences*, 352(1358) :1257, 1997.
- [35] M. Brand. Coupled hidden Markov models for modeling interacting processes. *Submitted to Neural Computation*, 1996.
- [36] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 994–999. IEEE, 1997.
- [37] Randal E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, 35 :677–691, 1986.
- [38] H.H. Bui. Abstract hidden markov models for online probabilistic plan recognition. 2002.
- [39] H.H. Bui, S. Venkatesh, and G. West. Tracking and surveillance in wide-area spatial environments using the Abstract Hidden Markov Model. pages 177–196, 2001.

- [40] A. Charles. Representation and analysis of reactive behaviors : A synchronous approach. 1996.
- [41] H. Chen, T. Kirubarajan, and Y. Bar-Shalom. Performance limits of track-to-track fusion versus centralized estimation : theory and application [sensor fusion]. *Aerospace and Electronic Systems, IEEE Transactions on*, 39(2) :386–400, 2003.
- [42] C.Y. Chong, K.C. Chang, and S. Mori. Distributed tracking in distributed sensor networks. In *American Control Conference, 1986*, pages 1863–1868. IEEE, 1986.
- [43] E. Clarke. Model checking. In *Foundations of Software Technology and Theoretical Computer Science*, pages 54–56. Springer, 1997.
- [44] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model checking*. MIT Press, 2000.
- [45] G. Cormode, M. Garofalakis, S. Muthukrishnan, and R. Rastogi. Holistic aggregates in a networked world : Distributed tracking of approximate quantiles. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 25–36. ACM, 2005.
- [46] P Couronne. *Le système Esterel v2*. PhD thesis, 1989.
- [47] L. Daniel. Haziness for common sensical inference from uncertain and inconsistent linear knowledge base. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, pages 163–170. IEEE, 2008.
- [48] A.P. Dempster, N.M. Laird, D.B. Rubin, et al. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1) :1–38, 1977.
- [49] J. Diard et al. *La carte bayésienne : un modèle probabiliste hiérarchique pour la navigation en robotique mobile*. PhD thesis, 2003.
- [50] Elèves du MIG. Métiers de l’ingénieur généraliste. Technical report, Ecole Nationale Supérieure des Mines de Paris, 2010.
- [51] C. Ducruet. A metageography of port-city relationships. 2007.
- [52] S.Z. Elloumi, J.P. Marmorat, V. Roy, and N. Maizi. Modeling ship behavior based on hidden markov models. In *Signal Processing, Pattern Recognition, and Applications/722 : Computer Graphics and Imaging*. ACTA Press, 2011.
- [53] S.Z. Elloumi, J.P. Marmorat, V. Roy, and N. Maizi. Securing harbors by modeling and classifying ships behavior. In *Proceedings on the 20th Behavior Representation in Modeling and Simulation (BRIMS)*, 2011.

- [54] M. Enzweiler and D.M. Gavrilă. Monocular pedestrian detection : Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2179–2195, 2008.
- [55] L.C. Evans. An introduction to mathematical optimal control theory version 0.2. *Lecture notes available at <http://math.berkeley.edu/~evans/control.course.pdf>*, 1983.
- [56] X. Feng and P. Perona. Human action recognition by sequence of movelet codewords. 2002.
- [57] A. Frémont and M. Soppé. Transport maritime conteneurisé et mondialisation. (2) :187–200, 2005.
- [58] D.M. Gavrilă. The visual analysis of human movement : A survey* 1. *Computer vision and image understanding*, 73(1) :82–98, 1999.
- [59] Gonthier Georges. *Sémantiques et modèles d'exécution des langages réactifs synchrones ; application à Esterel*. PhD thesis, 1988.
- [60] F. Glover and M. Laguna. Tabu search. 1998.
- [61] D.E. Goldberg. Genetic algorithms in search, optimization, and machine learning. 1989.
- [62] RO Goss. Economic policies and seaports : The economic functions of seaports. *Maritime Policy & Management*, 17(3) :207–219, 1990.
- [63] N. Halbwachs. *Synchronous programming of reactive systems*. 1998.
- [64] Nicolas Halbwachs, Fabienne Lagnier, and Christophe Ratel. Programming and verifying critical systems by means of the synchronous data-flow programming language LUSTRE. *IEEE Transactions on Software Engineering, Special Issue on the Specification and Analysis of Real-Time Systems*, sep 1992.
- [65] Nicolas Halbwachs, Fabienne Lagnier, and Pascal Raymond. Synchronous observers and the verification of reactive systems. In *Third Int. Conf. on Algebraic Methodology and Software Technology, AMAST'93, Twente*. Springer Verlag, 1993.
- [66] D.L. Hall and J. Llinas. *An introduction to multisensor data fusion*, volume 85. IEEE, 1997.
- [67] K. Han and M. Veloso. Automated robot behavior recognition. 9 :249–256, 2000.
- [68] S. Harnad. The symbol grounding problem. *Physica D : Nonlinear Phenomena*, 42(1-3) :335–346, 1990.

- [69] J.D. Ichbiah. Reference manual for the ada programming language. proposed standard document. Technical report, DTIC Document, 1980.
- [70] H.A. Kautz. *A formal theory of plan recognition*. PhD thesis, Bell Laboratories, 1987.
- [71] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.
- [72] J.H. Kim, S. Hayakawa, T. Suzuki, K. Hayashi, S. Okuma, N. Tsuchida, M. Shimizu, and S. Kido. Modeling of driver’s collision avoidance maneuver based on controller switching model. *Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on*, 35(6) :1131–1143, 2005.
- [73] N. Kuge, T. Yamamura, O. Shimoyama, and A. Liu. *A driver behavior recognition method based on a driver model framework*. SAE International, 2000.
- [74] A. Lakehal and I. Parissis. Structural test coverage criteria for lustre programs. In *Proceedings of the 10th international workshop on Formal methods for industrial critical systems*, pages 35–43. ACM, 2005.
- [75] Cosserat Laurent. *Sémantique opérationnelle du langage synchrone Esterel*. PhD thesis, 1985.
- [76] B.X.H. Le. Reconnaissance des comportements d’une personne âgée vivant seule dans un habitat intelligent pour la santé. 2008.
- [77] D. Le Guyader, D. Brosset, F. Gourmelon, et al. Exploitation de données ais pour la cartographie du transport maritime. *M@ ppemonde*, 2012.
- [78] F. Leblanc. Programmation visuelle pour la description du comportement d’un robot autonome. Master’s thesis, 1999.
- [79] J.J. Lee, J. Kim, J.H. Kim, et al. Data-driven design of hmm topology for on-line handwriting recognition. In *The 7th International Workshop on Frontiers in Handwriting Recognition*. Citeseer, 2000.
- [80] M. E. Lesk and E. Schmidt. *Lex, A Lexical Analyzer Generator*. Bell Laboratories, Murray Hill, New Jersey 07974, 1975.
- [81] S.E. Levinson, L.R. Rabiner, and M.M. Sondhi. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *The Bell System Technical Journal*, 62(4) :1035–1074, 1983.
- [82] Xiaolin Li, M. Parizeau, and R. Plamondon. Training hidden markov models with multiple observations-a combinatorial method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(4) :371 –377, apr. 2000.

- [83] L. Liao, D.J. Patterson, D. Fox, and H. Kautz. Behavior recognition in assisted cognition. In *Proceedings The AAAI-04 Workshop on Supervisory Control of Learning and Adaptive Systems*, 2004.
- [84] L. Liporace. Maximum likelihood estimation for multivariate observations of markov sources. *Information Theory, IEEE Transactions on*, 28(5) :729 – 734, sep 1982.
- [85] A. Liu and A. Pentland. Towards real-time recognition of driver intentions. pages 236–241.
- [86] H.M. Mandalia and D. Salvucci. Pattern recognition techniques to infer driver intentions. Technical report, 2004.
- [87] J. Marcadon. «ports et flux de conteneurs dans le monde, 1983-1992». *Mappe-monde*, 1 :29–34, 1995.
- [88] D. McErlean and S. Narayanan. Distributed detection and tracking in sensor networks. In *Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on*, volume 2, pages 1174–1178. IEEE, 2002.
- [89] X. Meng, K.K. Lee, and Y. Xu. Human driving behavior recognition based on hidden markov models. In *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*, pages 274–279. IEEE, 2006.
- [90] Xiaoning Meng, Ka Keung Lee, and Yangsheng Xu. Human driving behavior recognition based on hidden markov models. pages 274 –279, dec. 2006.
- [91] Frédéric Mignard. *Compilation du langage Esterel en systèmes d'équations booléennes*. PhD thesis, 1994.
- [92] P Mistretta, Nallino M., and Fleuriot F. Rapport final d'exécution de secmar. Technical report, 2012.
- [93] P. Naïm and A. Becker. *Réseaux bayésiens*, volume 5. Eyrolles, 2004.
- [94] J. Neering, M. Bordier, and N. Maizi. Optimal passive source localization. In *Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on*, pages 295–300. IEEE, 2007.
- [95] J. Neering, H. Elloumi, and L. Daniel. On possibilities of driver behavior prediction for driving simulator control. 2006.
- [96] A.V. Nefian and M.H. Hayes III. Hidden Markov models for face recognition. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 5, pages 2721–2724. IEEE, 2002.
- [97] N.Halbwachs. *Conception des systèmes réactifs. Les langages synchrones*. 1991.

- [98] S. Ohta, H. Nakamoto, Y. Shinagawa, and T. Tanikawa. A health monitoring system for elderly people living alone. *Journal of telemedicine and telecare*, 8(3) :151–156, 2002.
- [99] N. Oliver and A.P. Pentland. Driver behavior recognition and prediction in a SmartCar. In *PROC SPIE INT SOC OPT ENG*, volume 4023, pages 280–290. Citeseer, 2000.
- [100] N. Oliver and A.P. Pentland. Graphical models for driver behavior recognition in a smartcar. pages 7 –12, 2000.
- [101] V. Papailiopoulos. Automatic test generation for lustre/scade programs. In *Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering*, pages 517–520. IEEE Computer Society, 2008.
- [102] A. Pentland and A. Liu. Toward augmented control systems. pages 350–355, 1995.
- [103] A. Pentland and A. Liu. Modeling and prediction of human behavior. *Neural Computation*, 11(1) :229–242, 1999.
- [104] Dominique Perrin. Finite automata. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science. Volume B : Formal Models and Semantics*, pages 1–57. Elsevier and the MIT Press, 1990.
- [105] Gordon D. Plotkin. *A structural Approach to operational Semantics*. Journal of Logic and Algebraic Programming, 2004.
- [106] R. Poppe. Vision-based human motion analysis : An overview. *Computer Vision and Image Understanding*, 108(1-2) :4–18, 2007.
- [107] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6) :976–990, 2010.
- [108] F. Porikli. Trajectory distance metric using hidden markov model based representation. 2004.
- [109] L. Qiao, M. Sato, and H. Takeda. Learning algorithm of environmental recognition in driving vehicle. volume 25, pages 917–925. IEEE, 1995.
- [110] Lawrence R. Rabiner. A tutorial on HMM and selected applications in speech recognition. *Proceedings of the IEEE*, 77 :257–286, 1989.
- [111] L.R. Rabiner and B.H. Juang. Fundamentals of speech recognition. PTR Prentice-Hall. Inc., *Englewood Cliffs, NJ*, 1993.
- [112] D. Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6) :843–854, 1979.

- [113] Annie Ressouche, Daniel Gaffé, and Valérie Roy. Modular Compilation of a Synchronous Language. In Roger Lee, editor, *Software Engineering Research, Management and Applications*, volume 150 of *Studies in Computational Intelligence*, pages 151–171. Springer, August 2008.
- [114] I. Robert-Bobée. Projections de population 2005-2050 : vieillissement de la population en france métropolitaine. *Économie et statistique*, 408(1) :95–112, 2007.
- [115] R.A. Rutenbar. Simulated annealing algorithms : An overview. *Circuits and Devices Magazine, IEEE*, 5(1) :19–26, 1989.
- [116] D. Salvucci. “ tracking driver intentions with the act-r driver model. page 1.
- [117] D.D. Salvucci. Inferring driver intent : A case study in lane-change detection. In *Human Factors and Ergonomics Society Annual Meeting Proceedings*, volume 48, pages 2228–2231. Human Factors and Ergonomics Society, 2004.
- [118] D.D. Salvucci and T. Siedlecki. Toward a unified framework for tracking cognitive processes. In *Proceedings of the 25th Annual Conference of the Cognitive Science Society*, pages 1023–1028. Citeseer, 2003.
- [119] S. Sekizawa, S. Inagaki, T. Suzuki, S. Hayakawa, N. Tsuchida, T. Tsuda, and H. Fujinami. Modeling and recognition of driving behavior based on stochastic switched ARX model. *Intelligent Transportation Systems, IEEE Transactions on*, 8(4) :593–606, 2007.
- [120] E. Semerdjiev and L. Mihaylova. Variable-and Fixed-Structure Augmented Interacting Multiple Model Algorithms for Manoeuvring Ship Tracking Based on New Ship Models. *International Journal of Applied Mathematics and Computer Science*, 10(3) :591–604, 2000.
- [121] R. Singer, R. Sea, and K. Housewright. Derivation and evaluation of improved tracking filter for use in dense multitarget environments. *Information Theory, IEEE Transactions on*, 20(4) :423–432, 1974.
- [122] A. Sowmya, D. Tsz-Wang So, and W. Hung Tang. Design of a mobile robot controller using estereel tools. *Electronic Notes in Theoretical Computer Science*, 65(5) :3–10, 2002.
- [123] M. Stamp. A revealing introduction to hidden Markov models. *Department of Computer Science San Jose State University*, 2004.
- [124] F. Steenkeste, H. Bocquet, M. Chan, and E. Campo. La mise en place d’une technologie pour observer le comportement nocturne des personnes âgées en institution. *ITBM-RBM*, 22(1) :25–30, 2001.

- [125] T. Suzuki, S. Sekizawa, S. Inagaki, S. Hayakawa, N. Tsuchida, T. Tsuda, and H. Fujinami. Modeling and recognition of human driving behavior based on stochastic switched arx model. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 5095–5100. IEEE, 2005.
- [126] H. Takahashi and K. Kuroda. A study on automated shifting and shift timing using a driver's mental model. In *Intelligent Vehicles Symposium, 1996., Proceedings of the 1996 IEEE*, pages 300–305. IEEE, 1996.
- [127] R.D. Tennent. *Semantics of Programming Languages*. C.A. R. Hoare Series editor, 1991.
- [128] R. Van Der Merwe, E.A. Wan, and S. Julier. Sigma-point Kalman filters for nonlinear estimation and sensor-fusion : Applications to integrated navigation. pages 2004–5120, 2004.
- [129] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2) :260–269, 1967.
- [130] A.D. Wilson and A.F. Bobick. Hidden Markov models for modeling and recognizing gesture under variation. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1) :123–160, 2001.
- [131] D.H. Wilson. Assistive intelligent environments for automatic health monitoring. 2006.
- [132] Glynn Winskel. *The Normal Semantics of Programming Languages*. The MIT Press, Cambridge, Massachusetts, 1993.
- [133] C.F.J. Wu. On the convergence properties of the em algorithm. *The Annals of Statistics*, pages 95–103, 1983.
- [134] L. Xue, J. Yin, Z. Ji, and L. Jiang. A particle swarm optimization for hidden markov model training. *Signal Processing, 2006 8th International Conference on*, 1, 2006.
- [135] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 379–385. IEEE, 1992.
- [136] F. Yang and C. Zhang. An effective hybrid optimization algorithm for hmm. In *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, volume 4, pages 80–84. IEEE, 2008.

- [137] F. Yang, C. Zhang, and T. Sun. Comparison of particle swarm optimization and genetic algorithm for hmm training. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE.
- [138] J. Yang, Y. Xu, and C.S. Chen. Hidden markov model approach to skill learning and its application to telerobotics. *Robotics and Automation, IEEE Transactions on*, 10(5) :621–631, 1994.
- [139] J. Zeng, J. Duan, and C. Wu. A new distance measure for hidden Markov models. *Expert Systems with Applications*, 37(2) :1550–1555, 2010.

Reconnaissance de comportements Application à la sécurité portuaire

Résumé : Cette thèse s'est déroulée dans le cadre du projet SECMAR qui visait à sécuriser le Grand Port Maritime de Marseille. Notre objectif était d'aider les personnels du port à identifier les comportements menaçant des navires afin de pouvoir agir efficacement en cas de danger réel. A ce titre, nous avons développé un système d'analyse et de reconnaissance de comportements de navires formé de deux sous-modules complémentaires. Le premier est construit à partir de l'approche probabiliste Modèle de Markov Cachée et traite principalement des comportements nominaux des gros bateaux qui se caractérisent par un déplacement régulier et récurrent dans le port. Le second est construit à partir du langage réactif synchrone Esterel et prend en compte les comportements agressifs et transgressifs de tous types de navires, notamment ceux des petits bateaux qui circulent librement et aléatoirement dans le port. Le système global d'aide à la décision a permis une bonne reconnaissance en temps-réel des différents comportements de navires au cours de leurs évolutions dans le port. Au regard des résultats prometteurs que nous avons obtenu à travers ce module, il est envisageable de le généraliser à d'autres ports mondiaux ainsi qu'à d'autres domaines d'application, notamment le domaine aéroportuaire.

Mots clés : sécurité portuaire, reconnaissance de comportement, Modèle de Markov Cachée, langage réactif synchrone Esterel.

Behaviour recognition Application to harbor security

Abstract: The overall aim of this thesis was to create a decision support system that identifies discrepancies in ship behavior. The thesis was a part of the SECMAR project that aimed to improve security at the Marseilles harbor by the creation of decision support system for port staff. For this purpose, we developed a recognition behavior system consisting of two complementary sub-systems. The first system was based on the probabilistic Hidden Markov Model approach and deals with nominal behavior of large to medium size commercial ships showing regular and recurrent behavior. The second system was based on the reactive synchronous language Esterel and concerns aggressive and transgressive behavior of small ships that may navigate freely in the harbor. Real-time evaluations showed that the proposed decision support system efficiently captured and evaluated ship behaviors. The promising results of the system and its diversity in origin makes it suitable for applications in other harbors as well as other environment such as airports.

Keywords: Port security, recognition behavior, Hidden Markov Model, synchronous reactive language

