



HAL
open science

Reconnaissance des formes dans un environnement dynamique appliquée au diagnostic et au suivi des systèmes évolutifs

Laurent Hartert

► **To cite this version:**

Laurent Hartert. Reconnaissance des formes dans un environnement dynamique appliquée au diagnostic et au suivi des systèmes évolutifs. Automatique / Robotique. Université de Reims - Champagne Ardenne, 2010. Français. NNT : . tel-00549782

HAL Id: tel-00549782

<https://theses.hal.science/tel-00549782>

Submitted on 22 Dec 2010

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.



UNIVERSITE DE REIMS CHAMPAGNE-ARDENNE

Ecole Doctorale Sciences, Technologies, Santé

THÈSE

Présentée à

L'UFR SCIENCES EXACTES ET NATURELLES

Pour l'obtention du grade de

**DOCTEUR DE L'UNIVERSITE DE
REIMS CHAMPAGNE-ARDENNE**

Spécialité

Génie Informatique, Automatique et Traitement du Signal

par

Laurent HARTERT

**Reconnaissance des formes dans un environnement
dynamique
appliquée au diagnostic et au suivi des systèmes évolutifs**

Soutenue le 24 Novembre 2010 devant le jury composé de :

Frédéric VANDERHAEGEN	Professeur, Université de Valenciennes et du Hainaut-Cambrésis (<i>président du jury</i>)
Denis HAMAD	Professeur, Université du Littoral Côte d'Opale (<i>rapporteur</i>)
Sylvie CHARBONNIER	Maître de conférences HDR, Université Joseph Fourier de Grenoble (<i>rapporteur</i>)
Nada MATTA	HDR, Université de Technologie de Troyes (<i>examinatrice</i>)
Moamar SAYED MOUCHAWEH	Maître de conférences HDR, Université de Reims Champagne-Ardenne (<i>directeur de thèse</i>)
Patrice BILLAUDEL	Professeur, Université de Reims Champagne-Ardenne (<i>directeur de thèse</i>)

Remerciements

Ce manuscrit de thèse présente les travaux de recherche réalisés durant mon doctorat au sein de l'Université de Reims Champagne Ardenne et plus précisément au sein de l'Institut de Formation Technique Supérieure (IFTS) de Charleville-Mézières.

Durant ces trois années, j'ai eu l'occasion de créer des liens et de travailler avec de nombreuses personnes. Je tiens tout d'abord à remercier M. Janan ZAYTOON, professeur à l'Université de Reims et directeur du Centre de Recherche en Sciences et Technologies de l'Information et de la Communication (CReSTIC), pour m'avoir accueilli dans son laboratoire. Je remercie également M. Yvon GARDAN, professeur à l'Université de Reims et directeur de l'IFTS durant mes années de doctorat, pour m'avoir permis d'intégrer l'IFTS.

Parmi les personnes ayant joué un rôle majeur dans les travaux de cette thèse, je tiens à remercier vivement M. Moamar SAYED MOUCHAWEH, maître de conférences HDR à l'Université de Reims et directeur de ma thèse, pour m'avoir transmis sa passion de la recherche, m'avoir aidé et guidé tout au long de mes travaux de recherche et pour son amitié au quotidien. Je tiens également à témoigner toute ma gratitude à M. Patrice BILLAUDEL, professeur à l'Université de Reims, directeur de l'IFTS et de ma thèse, pour sa disponibilité, ses conseils et sa gentillesse.

Je remercie chaleureusement, M. Denis HAMAD, professeur à l'Université du Littoral Côte d'Opale et M^{me} Sylvie CHARBONNIER, maître de conférences HDR à l'Université Joseph Fourier de Grenoble pour avoir accepté d'être les rapporteurs de ma thèse, et pour l'ensemble des remarques et des conseils qu'ils m'ont donné. Tous mes remerciements vont également à M. Frédéric VANDERHAEGEN, professeur à l'Université de Valenciennes et du Hainaut-Cambrésis et à M^{me} Nada MATTA, HDR à l'Université de Technologie de Troyes, pour avoir accepté d'être les examinateurs de mes travaux de thèse et pour tous leurs précieux conseils.

J'adresse également mes sincères remerciements à M^{me} Estelle PERRIN pour m'avoir donné l'opportunité d'enseigner, ce que j'ai tout particulièrement apprécié, durant mes années de doctorat.

Je pense à tout le personnel de l'IFTS avec qui un contact très chaleureux a pu être établi, notamment avec Sébastien, Benoît, Tahar, Cosmin, Walid, Oana, Alexandre, Nicolas, Frédéric, Géraldine, Nicole, ... ainsi qu'aux membres du CReSTIC situés à Reims, Bernard, Noureddine, Kevin, Alexandre, François, Pascale, Benoît, Omar, Noureddine, Sofiane, Rachid, ... pour l'amitié qui nous unit, pour tous les bons moments passés ensemble ainsi que pour l'aide mutuelle que l'on a su s'apporter.

Je tiens à remercier le Conseil Général des Ardennes pour leur aide financière sans laquelle ces travaux de thèse n'auraient pas pu être réalisés.

Je ne peux terminer ces remerciements sans parler de toutes les personnes qui ont eu une place majeure dans ma vie personnelle, et également dans le cadre des travaux de cette thèse. Je pense particulièrement à mes parents, à ma grand-mère, ainsi qu'à mon frère et à sa famille pour leur affection et pour m'avoir donné les moyens et la volonté de faire ces études. Je pense également à ma belle famille qui a toujours été présente durant ces années importantes. Enfin, je tiens à dédier ce mémoire de thèse à ma merveilleuse épouse Amélie qui a su me soutenir durant ces trois années, ainsi qu'à ma magnifique fille Méline qui fait de moi un papa comblé.

Table des matières

Introduction générale.....	11
Chapitre 1. Introduction à la reconnaissance des formes.....	20
1.1. Principe général des méthodes de reconnaissance des formes.....	20
1.1.1. Introduction	20
<i>1.1.1.a. Principe</i>	<i>21</i>
<i>1.1.1.b. Prétraitement des données</i>	<i>23</i>
<i>1.1.1.c. Apprentissage et classification.....</i>	<i>24</i>
<i>1.1.1.d. Environnement statique et environnement dynamique.....</i>	<i>25</i>
1.1.2. Reconnaissance statique des formes statiques	25
1.1.3. Reconnaissance dynamique des formes statiques	27
1.1.4. Reconnaissance statique des formes dynamiques	28
1.1.5. Reconnaissance dynamique des formes dynamiques.....	31
1.2. Reconnaissance des formes dans un environnement dynamique.....	32
1.2.1. Définitions et objectifs	32
1.2.2. Principe d'adaptation des paramètres du classifieur.....	33
1.2.3. Méthodes de reconnaissance dynamique des formes.....	35
<i>1.2.3.a. Réseaux de neurones</i>	<i>36</i>
<i>1.2.3.b. Méthodes à noyau.....</i>	<i>41</i>
<i>1.2.3.c. Arbres de décision</i>	<i>46</i>
<i>1.2.3.d. Autres méthodes de reconnaissance dynamique des formes.....</i>	<i>47</i>
1.2.4. Méthodes de reconnaissance des formes dynamiques	50
<i>1.2.4.a. Méthodes de segmentation</i>	<i>51</i>
<i>1.2.4.b. Mesures de similarité entre les formes dynamiques.....</i>	<i>54</i>
<i>1.2.4.c. Classification des formes dynamiques.....</i>	<i>56</i>
1.2.5. Applications des méthodes de classification dynamique	59
1.3. Conclusion.....	60

Chapitre 2. Contributions à la reconnaissance dynamique des formes	62
2.1. Fuzzy Pattern Matching Dynamique (FPMD)	62
2.1.1. Principe de la méthode Fuzzy Pattern Matching Incrémentale (FPMI).....	63
2.1.1.a. Phase d'apprentissage	63
2.1.1.b. Phase de classification.....	64
2.1.1.c. Phase de mise à jour incrémentale.....	65
2.1.2. Principe de la méthode Fuzzy Pattern Matching Dynamique (FPMD)	66
2.1.2.a. Détection des évolutions.....	67
2.1.2.b. Adaptation des classes évoluées.....	69
2.1.2.c. Validation des classes.....	70
2.1.3. Définition des paramètres de la méthode FPMD supervisée	74
2.2. K-Plus Proches Voisins Flous Dynamique (KPPVFD)	76
2.2.1. Versions antérieures des K-Plus Proches Voisins.....	76
2.2.2. KPPVFD en version Supervisée (KPPVFD-S).....	78
2.2.2.a. Détection des évolutions de classes	78
2.2.2.b. Adaptation des classes évoluées.....	80
2.2.2.c. Validation des classes.....	80
2.2.3. KPPVFD en version Semi-Supervisée (KPPVFD-SS)	83
2.2.3.a. Apprentissage et classification.....	84
2.2.3.b. Détection de l'évolution d'une classe	85
2.2.3.c. Adaptation d'une classe après validation de son évolution et création d'une nouvelle classe.....	85
2.2.3.d. Validation des classes	86
2.2.4. Définition des paramètres des méthodes KPPVFD-S et KPPVFD-SS	89
2.3. Protocole expérimental pour la validation des méthodes proposées.....	90
2.3.1. Limites des méthodes incrémentales pour les classes dynamiques.....	93
2.3.2. Application de la méthode FPMD sur des classes dynamiques	102
2.3.3. Application de la méthode KPPVFD-S sur des classes dynamiques	115
2.3.4. Application de la méthode KPPVFD-SS sur des classes dynamiques.....	121
2.3.5. Comparatif des résultats obtenus.....	127
2.4. Conclusion.....	128

Chapitre 3. Contributions aux méthodes de reconnaissance des formes dynamiques .. 131

3.1. Introduction	131
3.2. Approches de reconnaissance structurelle des formes dynamiques.....	134
3.3. Approche proposée de reconnaissance structurelle des formes dynamiques.....	137
3.3.1. Adaptation du nombre de primitives	139
3.3.2. Choix des primitives.....	141
3.3.3. Méthode de segmentation.....	141
3.3.4. Classification des formes dynamiques par mesure de similarité.....	144
3.3.5. Exemples illustratifs	146
3.4. Combinaison de classifieurs basée sur les caractéristiques statistiques et structurelles des formes dynamiques	152
3.4.1. Méthodes existantes	153
3.4.1.a. Principe de fonctionnement des multi-classifieurs.....	153
3.4.1.b. Exemples de multi-classifieur.....	155
3.4.2. Approche mixte proposée de reconnaissance des formes dynamiques.....	156
3.4.2.a. Méthode de classification statistique	159
3.4.2.b. Fusion des résultats par le multi-classifieur	162
3.4.2.c. Signature mixte interprétable et mesure de similarité	163
3.5. Conclusion.....	165

Chapitre 4. Application et validation des approches proposées 169

4.1. Application et validation des méthodes proposées de reconnaissance dynamique des formes.....	169
4.1.1. Benchmark d'un système hydraulique à deux réservoirs	170
4.1.1.a. Présentation du benchmark.....	170
4.1.1.b. Protocole expérimental	172
4.1.1.c. Résultats de classification	174
4.1.2. Système de commutation	177
4.1.2.a. Le système	177
4.1.2.b. Recherche des paramètres discriminatifs	178
4.1.2.c. Résultats de classification	180
4.1.3. Suivi de qualité des soudures réalisées par un robot industriel.....	184

4.1.3.a. <i>Système industriel de soudure</i>	184
4.1.3.b. <i>Protocole expérimental</i>	185
4.1.3.c. <i>Phase d'analyse des signaux acquis</i>	188
4.1.3.d. <i>Résultats de classification</i>	191
4.1.4. Suivi du fonctionnement d'un générateur de vapeur du Réacteur à Neutrons Rapides (RNR) Prototype Fast Reactor (PFR).....	195
4.1.4.a. <i>Présentation du système</i>	195
4.1.4.b. <i>Analyse des données</i>	196
4.1.4.c. <i>Résultats de classification</i>	197
4.1.4.d. <i>Définition d'un automate pour améliorer le taux de classification</i>	199
4.2. Application de la méthode mixte proposée pour la reconnaissance des formes dynamiques.....	201
4.2.1. Caractérisation et interprétation de la coordination inter-segmentaire de la marche des patients hémiparétiques.....	201
4.2.1.a. <i>Présentation de l'application</i>	201
4.2.1.b. <i>Etablissement de la base de données clinique</i>	202
4.2.1.c. <i>Approches existantes</i>	205
4.2.1.d. <i>Résultats de classification</i>	206
4.2.1.e. <i>Interprétation des résultats</i>	211
4.2.1.f. <i>Utilisation de la mesure de similarité mixte</i>	213
4.3. Conclusion.....	215
Conclusion générale et Perspectives	218
Bibliographie.....	225
Résumé	243
Abstract.....	245

Introduction générale

Contexte

Les fonctions de diagnostic et de suivi s'appuient généralement sur un modèle de fonctionnement normal et/ou défaillant. Ce modèle est construit en utilisant une connaissance *a priori* sur le fonctionnement du système. Lorsque cette connaissance est suffisante pour construire un modèle mathématique ou analytique du système, les méthodes à base de modèle qualitatif et/ou quantitatif sont privilégiées. Lorsque cette connaissance est minimale et insuffisante pour construire un modèle analytique, les méthodes de Reconnaissance des Formes (RdF) constituent une solution efficace pour construire un modèle du système. Ce modèle est construit par apprentissage en utilisant un historique basé sur des observations du fonctionnement du système.

La RdF regroupe l'ensemble des méthodes permettant la classification des formes (vecteurs de caractéristiques de l'état de fonctionnement du système) dans des classes. Chaque classe est associée à un mode de fonctionnement du système et elle est représentée par un modèle. Les modèles de classes sont utilisés pour la classification d'une nouvelle forme à une des classes existantes.

Dans le contexte du diagnostic, une forme est l'observation simplifiée de l'état du système. La classification d'une forme est réalisée par un module de classification, appelé classifieur, en utilisant une règle de décision. Le problème de la RdF est donc de caractériser le modèle et l'étiquette de chaque classe associée à une forme. Cela requiert le recours à des techniques de classification permettant de regrouper, par apprentissage, les formes similaires. Le diagnostic par RdF devient alors un problème de classification ; le mode de fonctionnement actuel du système est déterminé en connaissant la classe de l'observation actuelle de l'état de fonctionnement du système. Le modèle de chaque classe est représenté par une fonction d'appartenance qui détermine la valeur d'appartenance d'une forme à chaque classe.

Il existe trois grands types d'apprentissage : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage semi-supervisé. L'apprentissage supervisé nécessite que les observations soient étiquetées ; on connaît *a priori* le mode de fonctionnement associé à chacune de ces observations. L'apprentissage non supervisé cherche à former des classes dans un ensemble d'observations non étiquetées pour lesquelles aucune connaissance n'est disponible sur leur appartenance aux différentes classes. L'apprentissage semi-supervisé

combine les deux modes d'apprentissage supervisé et non supervisé. Il utilise l'ensemble d'apprentissage étiqueté pour estimer les fonctions d'appartenance de chaque classe connue et l'apprentissage non supervisé pour améliorer la précision de cette estimation, détecter les nouvelles classes et apprendre leurs fonctions d'appartenance. Cette combinaison est particulièrement efficace quand l'ensemble d'apprentissage étiqueté contient peu de points de chaque classe. Cela est souvent le cas pour les classes représentant les modes de fonctionnement défectueux. De plus, les modes de fonctionnement, surtout défectueux, sont rarement tous représentés dans l'ensemble d'apprentissage. En effet pour des raisons de coût ou de sécurité, certains modes de fonctionnement défectueux ou dangereux ne peuvent pas être provoqués et sont donc absents de l'ensemble d'apprentissage. Un apprentissage non supervisé permet donc de détecter et d'apprendre les nouveaux modes de fonctionnement au fur et à mesure qu'ils apparaissent afin d'enrichir la connaissance initiale.

Problématique

Les classes peuvent être statiques ou dynamiques. Les classes statiques sont basées sur des données stationnaires, ce qui signifie que les paramètres du modèle de classification demeurent inchangés au cours du temps. Pour ce type de données, l'enrichissement de la base de connaissance se traduit par une déformation locale du contour ou de la forme des classes sans mettre en cause les informations acquises précédemment. Toutefois, comme l'environnement d'un système est en perpétuelle évolution, la majeure partie des données issues du monde réel est non stationnaire. Ce type de données entraîne une variation au cours du temps des paramètres du modèle de classification. Dans ce cas, les classes et/ou les formes sont dynamiques. Cette dynamique se traduit par le déplacement, l'élimination, la fusion, la scission, ..., des classes. Parmi les applications réelles nécessitant une classification dynamique, on peut citer le diagnostic industriel (évolution des modes de fonctionnement), le diagnostic médical (expansion de maladie tel qu'un cancer), la surveillance vidéo (mouvement des cibles), etc.

Les travaux de cette thèse prennent place dans le contexte du diagnostic et du suivi des systèmes évolutifs à l'aide des méthodes de classification dynamique. Un système évolutif évolue d'un mode à un autre de façon brutale (saut) ou progressive (dérive) au cours du temps. Cette évolution est la conséquence des changements de caractéristiques d'un système à cause d'une fuite, de l'usure d'un outil, d'un réglage, etc. Lorsqu'une méthode statique de reconnaissance des formes est utilisée pour construire les modèles de classes d'un système

évolutif, elle permet de classifier les nouvelles observations en fonction des observations historiques connues mais elle ne tient pas compte des nouvelles informations caractéristiques pour mettre à jour les modèles des classes (fonctions d'appartenance par exemple). Avec une méthode de classification statique quand un changement survient dans un système, les modèles des classes ne deviennent plus aptes à représenter les caractéristiques actuelles du système.

Le suivi et le diagnostic des systèmes évolutifs nécessitent donc un apprentissage continu au cours du temps afin de tenir compte des évolutions et des changements dans leur environnement. Les méthodes d'apprentissage incrémental constituent une solution efficace pour réaliser l'apprentissage continu. Ces méthodes permettent d'intégrer en ligne l'information apportée par une nouvelle forme pour améliorer l'estimation des modèles de classes. Cependant, ces méthodes adaptent les modèles de classes sans remettre en cause les connaissances acquises précédemment. Elles considèrent toutes les formes comme représentatives des classes. Or, certaines de ces formes deviennent obsolètes et ne doivent pas être utilisées pour la mise à jour des modèles de classes. Cela nécessite donc l'utilisation d'un mécanisme permettant d'oublier les formes devenues obsolètes et de ne garder que les formes représentatives des caractéristiques actuelles des classes après leur évolution. Ces méthodes sont appelées méthodes de reconnaissance dynamique des formes (ou méthodes de classification dynamique). Le défi de ces méthodes est de sélectionner les formes représentatives des changements des classes et d'éviter un oubli catastrophique.

Les données issues du fonctionnement d'un système évolutif sont représentées par un certain nombre de paramètres, statistiques, structurels ou mixte (statistiques et structurels) dont les caractéristiques évoluent avec le temps. En fonction des paramètres utilisés par la méthode, cette dernière fera partie des méthodes statistique, structurelle ou mixte de classification. Ces méthodes peuvent être supervisées, non supervisées, ou semi-supervisées, selon le type d'informations disponibles sur le système, respectivement étiquetées *a priori*, non étiquetées, ou partiellement étiquetées (Figure I.1).

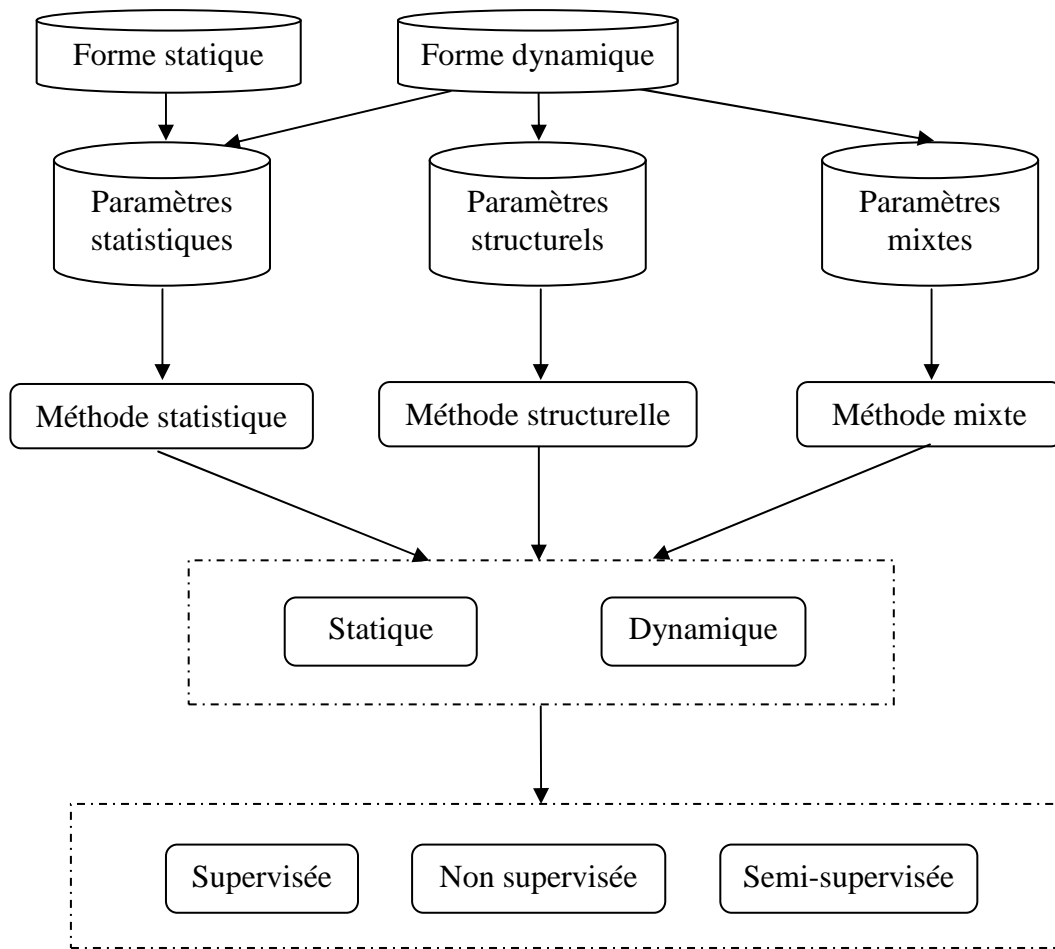


Figure I.1. Représentation des types de méthodes de classification utilisables en fonction des formes et des paramètres d'un système.

Contribution

Les travaux effectués dans le cadre de cette thèse sont orientés vers la recherche et le développement d'algorithmes de reconnaissance dynamique des formes adaptés à la classification des données issues des systèmes évolutifs.

Ces algorithmes sont basés sur diverses méthodes de la littérature dont les capacités pour la classification des données statistiques ont été démontrées et dont l'utilisation est fréquente. Ainsi, les développements réalisés pourront être pleinement intégrés aux systèmes utilisant déjà ces méthodes de classification.

Les méthodes proposées utilisent les informations de manière supervisée ou semi-supervisée et elles sont adaptées aux formes statiques ou dynamiques. Ces méthodes se veulent complémentaires aux approches de classification dynamique déjà développées. Les méthodes statistiques proposées peuvent fonctionner en ligne ou hors ligne. Seules les

méthodes structurelle et mixte proposées sont utilisables uniquement hors ligne. Nous nous sommes intéressés aux seules méthodes supervisées et semi-supervisées, tenant compte des informations connues *a priori* sur un système. En effet, puisque l'on s'intéresse dans cette thèse au suivi des modes de fonctionnement des systèmes évolutifs, on part du point de vue qu'un certain nombre d'informations sont déjà connues sur un système.

Les méthodes proposées dans cette thèse pour réaliser la reconnaissance des formes dans un environnement dynamique sont :

- Fuzzy Pattern Matching Dynamique (FPMD) : nouvelle version de la méthode Fuzzy Pattern Matching Incrémentale (FPMI) intégrant plusieurs mécanismes adaptés au suivi des classes dynamiques. Ces méthodes utilisent la théorie des ensembles flous et la théorie des possibilités. La classification des formes est réalisée à l'aide des densités linéaires de possibilité obtenues pour chaque histogramme d'une classe par rapport à chaque paramètre de l'espace de représentation. FPMD tient compte du changement graduel des caractéristiques des classes à l'aide d'une phase de détection des évolutions des classes, d'une phase d'adaptation des classes évoluées, et d'une phase de validation des classes existantes. La phase de détection des évolutions permet de calculer les valeurs de représentativité des formes en fonction des changements progressifs de la distribution des données. La mesure de représentativité des formes représente l'importance qu'ont les données classées dans la mise à jour d'un modèle de classe. Basé sur cette mesure, un résidu est calculé afin de suivre l'évolution des changements de distribution des données et donc l'évolution des modèles de classes. A l'aide de ces deux mesures, la méthode FPMD peut valider l'évolution détectée pour une classe et adapter incrémentalement cette dernière. L'adaptation d'une classe ou de son modèle consiste en sa mise à jour en utilisant les formes représentatives des caractéristiques actuelles de cette classe. Lorsque l'adaptation d'une classe a lieu, seules les formes caractéristiques de son évolution sont conservées dans le modèle de la classe. Des procédures de fusion, de scission et de suppression des classes ont également été développées pour mettre à jour les modèles des classes en fonction de certaines évolutions pouvant avoir lieu sur les systèmes évolutifs. D'autres évolutions des modèles de classes telles que des déplacements et rotations des classes sont aussi gérées par la méthode FPMD,
- K-Plus Proches Voisins Flous Dynamique (KPPVFD) : développée en version supervisée (KPPVFD-S) et semi-supervisée (KPPVFD-SS). KPPVFD-S et KPPVFD-

SS sont des méthodes de classification dynamique basées sur la méthode des K-Plus Proches Voisins Flous (KPPVF). Tout comme KPPVF, elles utilisent donc la notion de distance entre les formes pour classer les nouvelles données. Deux indicateurs d'évolutions sont calculés par rapport aux paramètres des classes. Ces indicateurs sont utilisés dans la phase de détection et de confirmation des évolutions de classe. Durant la phase d'adaptation des classes, les formes les plus représentatives de l'évolution confirmée, sont sélectionnées et le classifieur est mis à jour. Ces méthodes permettent de répondre aux problèmes de mise à jour des classifieurs dynamiques pour les cas de déplacement, fusion, scission, rotation des classes, etc. La méthode semi-supervisée a été développée dans le but de pouvoir utiliser la méthode KPPVFD sur les systèmes dont les données disponibles sont très limitées. En effet, lorsque des données étiquetées sont disponibles, leur utilisation permet d'améliorer l'estimation des modèles des classes. Les deux méthodes KPPVFD développées sont complémentaires. La méthode supervisée est plus adaptée au suivi des systèmes dont les classes principales sont connues tandis que la méthode semi-supervisée est plus adaptée à la détection des nouvelles classes et à leur adaptation en fonction des évolutions des systèmes,

- une méthode de classification structurelle pour les formes dynamiques. Cette méthode comporte une phase de segmentation des formes et d'adaptation du nombre de primitives utilisées ainsi qu'une phase de classification des formes dynamiques. Une primitive est une représentation symbolique donnée à chaque segment (appelé épisode) caractérisant une forme dynamique après segmentation. La phase de segmentation est basée sur les approches descendantes de segmentation. Elle cherche à minimiser l'erreur d'approximation de segmentation en utilisant un nombre optimal de primitives trouvé récursivement. Ce nombre de primitives est identique pour toutes les formes dynamiques issues d'un système afin de mesurer leur similarité. Cette première phase permet d'obtenir une représentation simplifiée et caractéristique de chaque forme dynamique d'une même classe, i.e. une signature, sans utiliser de seuil d'erreur d'approximation. Dans la deuxième phase de la méthode, la classification des signatures est réalisée à l'aide d'une mesure de similarité. Cette mesure permet de calculer la similarité d'une forme dynamique à une classe. La méthode se veut générale et applicable à plusieurs systèmes réels,

- une méthode mixte de classification pour les données dynamiques. Cette méthode utilise d'une part la méthode structurale définie dans cette thèse et d'autre part une méthode de classification statistique. L'utilisation de ces deux méthodes a pour but d'obtenir un meilleur taux de classification en tenant compte des deux types d'information (statistique et structurel) présents dans certaines données. L'originalité de la méthode repose particulièrement sur la définition des paramètres statistiques à partir des épisodes trouvés par la méthode de segmentation pour obtenir la signature d'une forme dynamique. De plus, une technique de caractérisation des classes à l'aide de signatures mixtes, i.e. basées sur les caractéristiques structurales et statistiques obtenues pour une forme, est proposée pour obtenir une signature interprétable par les opérateurs humains. Dans le cadre des évolutions des formes dynamiques, une mesure de similarité mixte est définie afin de quantifier l'évolution réalisée par les formes. Cette mesure de similarité est calculée à partir des paramètres mixtes qui caractérisent chaque forme. La mesure de similarité ainsi calculée permet également d'obtenir le degré d'appartenance d'une forme à une classe.

Organisation du manuscrit

Le manuscrit est séparé en quatre chapitres organisés comme suit.

Dans le premier chapitre, les différents types de reconnaissance des formes, statiques et dynamiques, sont présentés. Le principe de fonctionnement de plusieurs méthodes dynamiques est détaillé et des exemples d'applications de ces méthodes dans un milieu dynamique sont présentés.

Le chapitre deux correspond au développement réalisé sur les méthodes de reconnaissance dynamique des formes. Ces méthodes adaptées au suivi des classes dynamiques sont basées sur les informations statistiques obtenues sur les données. Chaque méthode est illustrée par des exemples et est comparée à plusieurs méthodes de classification.

Dans le chapitre trois, la contribution aux méthodes structurale et mixte de reconnaissance des formes dynamiques est détaillée. Des exemples de fonctionnement caractérisant les méthodes proposées sont présentés et plusieurs méthodes structurales et mixtes sont décrites. Le fonctionnement de ces méthodes permet d'établir une comparaison entre les méthodes proposées adaptées à certaines applications et certaines méthodes existantes plus adaptées à d'autres types d'applications.

Le chapitre quatre contient l'ensemble des applications simulées et réelles utilisées dans le cadre de cette thèse. En effet, en plus de l'utilisation d'exemples académiques de bases de données dynamiques, les approches proposées ont été validées sur des applications dynamiques. Les applications utilisées dans le chapitre quatre sont : un benchmark de deux réservoirs dont le suivi des fuites est réalisé, un système de commutation des modes de fonctionnement dont l'écart est faible, un système industriel dont le suivi de qualité des soudures doit permettre de détecter l'évolution de qualité des pièces réalisées, un générateur de vapeur de réacteur nucléaire pour lequel on souhaite détecter et caractériser l'ampleur des fuites d'eau, et une application clinique concernant la coordination inter-segmentaire de la marche des patients hémiparétiques.

Chapitre 1. Introduction à la reconnaissance des formes

Dans ce chapitre, une synthèse des méthodes de reconnaissance des formes de la littérature est présentée. Ces méthodes sont divisées en quatre catégories selon les données disponibles sur le fonctionnement du système et sur son environnement (statique, dynamique). Ces catégories sont : la reconnaissance statique des formes statiques, la reconnaissance dynamique des formes statiques, la reconnaissance statique des formes dynamiques et la reconnaissance dynamique des formes dynamiques. Dans le cadre de cette thèse, nous détaillerons plus particulièrement les méthodes de classification dynamique adaptées pour le diagnostic et le suivi des systèmes évolutifs dont les caractéristiques évoluent avec le temps. Ces méthodes permettent l'adaptation de la structure et des paramètres du classifieur en utilisant les formes représentatives du fonctionnement du système après son évolution.

1.1. Principe général des méthodes de reconnaissance des formes

1.1.1. Introduction

Le suivi de fonctionnement continu des systèmes industriels, tels que les robots, les systèmes hydrauliques, les machines de tri, etc. permet d'améliorer leur productivité et de faire décroître leur coût de production. Quand une faute est détectée, le système de suivi exécute une procédure de correction pour ramener le procédé dans des conditions normales d'opération. Plusieurs méthodes peuvent être utilisées pour réaliser le suivi de ces systèmes. La sélection d'une de ces méthodes dépend de plusieurs facteurs comme :

- la dynamique du système (discrète, continue ou hybride),
- les contraintes d'environnement (en ligne pour une réaction souhaitée en temps réel, ou hors ligne si la réaction peut être prise après un certain temps),
- la représentation des informations (quantitative et/ou qualitative),
- la complexité du système (grande ou simple),
- l'information disponible sur le système (structurelle, analytique, connaissances heuristiques, etc.),
- ...

Quand les connaissances *a priori* du comportement d'un procédé ne sont pas suffisantes pour construire un modèle analytique et quand seules des mesures sur son état de

fonctionnement sont disponibles, les méthodes de Reconnaissance des Formes (RdF) (Jain, 2000) (Brown, 1989) (Dubuisson, 2001) (Duda, 2001) (Webb, 2002) sont particulièrement intéressantes. En effet, ces méthodes sont adaptées à la surveillance des modes de fonctionnement des systèmes, au diagnostic des fautes, ou à l'accomplissement du pronostic des défaillances. En RdF, le système est considéré comme une boîte noire, c'est-à-dire qu'aucune équation mathématique n'est nécessaire pour modéliser son fonctionnement. Les méthodes de RdF utilisent exclusivement un ensemble de mesures et/ou de connaissances heuristiques du fonctionnement d'un procédé pour passer de l'espace d'observation à l'espace de décision, appelé espace de représentation. Les résultats de classification dépendent alors seulement, de la méthode elle-même, des connaissances *a priori*, des paramètres caractérisant le système, et de la qualité des données.

1.1.1.a. Principe

Le principe de la RdF est de classer les nouvelles formes en utilisant un classifieur qui génère une fonction d'appartenance pour chaque classe. Ainsi la classification d'un nouveau point peut se faire en fonction de la valeur d'appartenance qu'il obtient par rapport à chaque classe. Les méthodes de RdF peuvent généralement être divisées en deux catégories : les méthodes paramétriques et les non paramétriques. Les méthodes paramétriques considèrent l'ensemble d'apprentissage comme des données indépendantes, toutes distribuées selon la même loi de probabilité, comme le fait le classifieur Bayésien (Dubuisson, 1990) (Yager, 2006). Cette hypothèse n'est bien souvent pas valide dans le cas d'applications réelles, c'est pourquoi nous nous intéresserons seulement aux méthodes non paramétriques dans l'ensemble de ce manuscrit. Les méthodes non paramétriques génèrent les fonctions d'appartenance des classifieurs (Medasani, 1998) soit :

- en estimant la fonction de densité de probabilité conditionnelle pour chaque classe, comme le fait la méthode des fenêtres de Parzen (Parzen, 1962) (Dubuisson, 2001) et la méthode des K-Plus Proches Voisins (Cover, 1967) (Denoeux, 2001),
- en construisant par apprentissage les régions de décision comme le font, les fonctions potentielles (Dubuisson, 2001) (Batchelor, 1973), les réseaux de neurones (Ripley, 1996) ou les Séparateurs à Vaste Marge (Vapnik, 1995),
- en se basant sur des heuristiques (Dombi, 1990) (Dimitru, 1986).

Les performances des méthodes de RdF dépendent des connaissances *a priori* obtenues sur les modes de fonctionnement du système. Souvent, ce savoir est imparfait et incomplet.

L'imperfection est due à l'utilisation des capteurs, au bruit des données, et aux évaluations de l'expert du système. Le savoir est incomplet puisqu'il ne contient pas toutes les informations concernant l'ensemble des modes de fonctionnement possibles pour un procédé. C'est surtout le cas concernant les modes dangereux et les modes pour lesquels un temps de fonctionnement très important est nécessaire. Le problème de l'imperfection du savoir et des données peut être solutionné en utilisant des techniques de la théorie des ensembles flous. Le problème d'incomplétude est résolu par un apprentissage continu des données permettant de mettre à jour le système de surveillance. En utilisant une méthode de RdF, chaque nouvelle information, i.e. chaque nouvelle forme, pourra être prise en compte dans la mise à jour du classifieur et intégrée dans la base des données connues. Cependant, il peut être très difficile d'obtenir des données exhaustives pour certains procédés.

La RdF est réalisée en deux phases : l'apprentissage à partir des données connues et la classification des nouvelles données. En amont de ces deux phases, une étape de prétraitement est utilisée pour trouver l'ensemble minimal de paramètres informatifs nécessaire à l'établissement de l'espace de représentation. L'ensemble des étapes majeures d'une approche de reconnaissance des formes est présenté sur la figure 1.1.

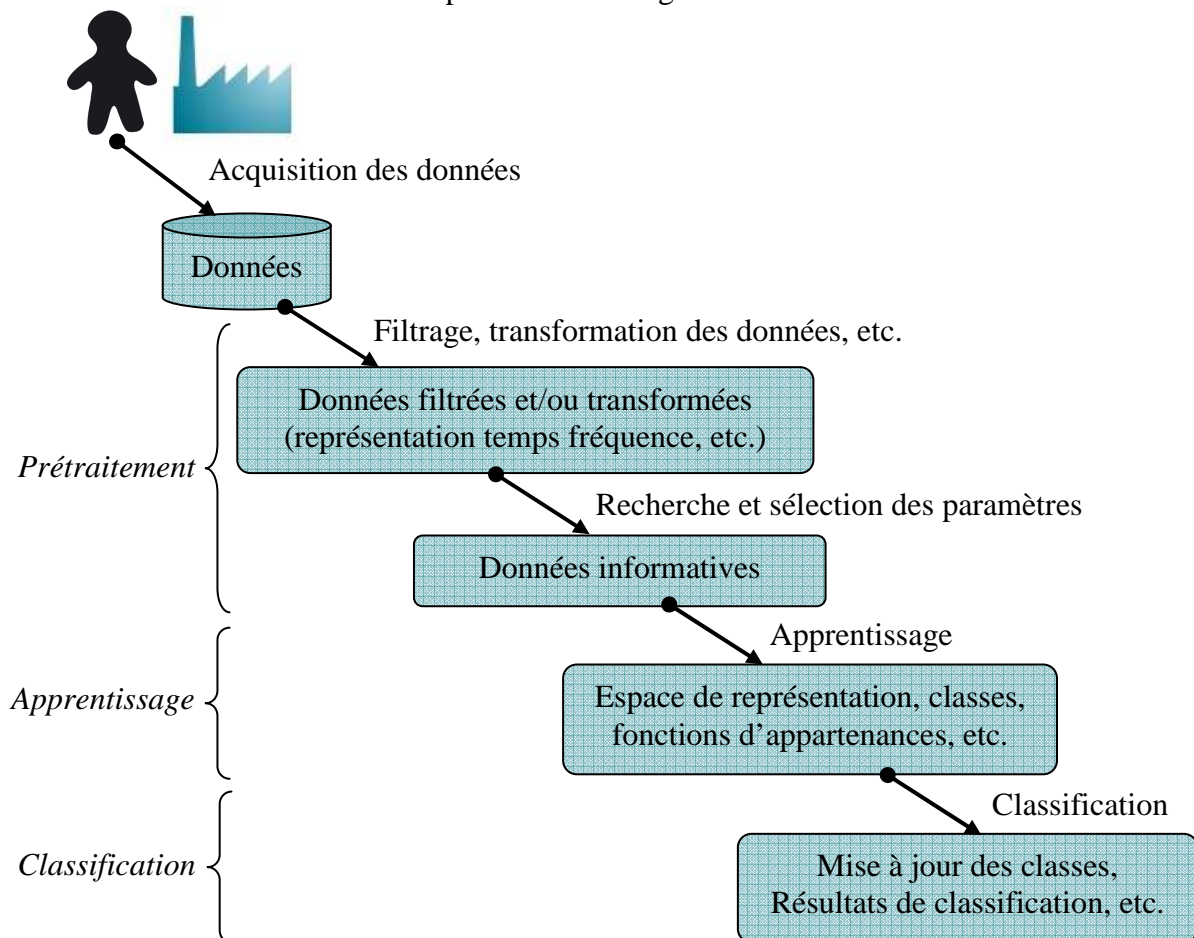


Figure 1.1. Les étapes principales d'une approche de reconnaissance des formes.

1.1.1.b. Prétraitement des données

La première phase du prétraitement concerne le filtrage et la transformation des données qui permettent d'obtenir des données plus adaptées à la recherche de caractéristiques informatives. Cette phase peut être réalisée par de nombreux algorithmes de traitement (Trouilhet, 1994) tels que : l'analyse standard de Fourier, les méthodes Temps-Fréquence (Wigner-Ville, Gabor, etc.), les ondelettes (Chimentin, 2008), les filtres Auto-Régressif (AR), moyenne mobile (Moving Average ou MA) ou auto-régressif et moyenne mobile (ARMA) (Box, 1970), etc. L'analyse des données obtenues sur un système permet d'obtenir des caractéristiques statistiques, c'est-à-dire des valeurs scalaires, ou structurelles. Les caractéristiques statistiques fournissent des informations spatiales telles que le nombre de pics présents dans un signal, l'écart type des données, la moyenne quadratique (RMS), la valeur maximale ou minimale, le coefficient d'aplatissement (Kurtosis), etc. Lorsque les données sont obtenues sous forme continue dans le temps (signaux), des caractéristiques structurelles peuvent être obtenues en utilisant une technique de segmentation. Les techniques de segmentation (Triki, 2005) peuvent utiliser différents types de primitives (Colomer, 2002) qui sont connectées en maintenant la structure de chaque signal. En conservant la connexion entre les différentes parties d'un signal, une technique de segmentation permet d'obtenir une signature, qui est une représentation simplifiée et caractéristique de chaque signal.

Lors de la deuxième phase de prétraitement, on effectue la sélection des paramètres les plus informatifs. Généralement un grand nombre de paramètres est calculé à partir des données recueillies par les capteurs d'un système. C'est pourquoi, il est nécessaire d'extraire et/ou de sélectionner les paramètres les plus informatifs (Benzécri, 1973), c'est-à-dire ceux qui permettront de discriminer au maximum les classes. Les méthodes de sélection des paramètres choisissent le sous-ensemble de paramètres les plus informatifs. Les méthodes d'extraction des paramètres créent un sous-ensemble de nouveaux paramètres par combinaison des paramètres existants. Pour réaliser ces phases de sélection et/ou d'extraction, on utilise des techniques d'analyse des données (Benzécri, 1973) telles que l'Analyse en Composantes Principales (ACP) (Jolliffe, 1988), la mesure discriminante de Fisher (Fisher, 1936), Multidimensional scaling (Sammon, 1969), les techniques de wrapper (Uncu, 2007), etc. Plus les paramètres du système permettent de bien discriminer les formes, plus les résultats de classification sont bons. L'ensemble des paramètres trouvés par ces méthodes représente les attributs qui permettent de caractériser chaque forme. Lorsque les données issues de l'observation du fonctionnement d'un système sont représentées par des paramètres

statistiques, elles sont transformées en formes, c'est-à-dire en points, dans l'espace de représentation. Les groupes de formes similaires sont appelés classes. Si les paramètres sont bien déterminés, les classes sont bien discriminées et elles sont situées dans différentes régions de l'espace de représentation. Chaque classe est associée à un mode de fonctionnement (normal ou défaillant). Ces formes, avec leurs assignements à une classe, constituent l'ensemble d'apprentissage. Elles sont représentées par d caractéristiques, ou attributs, ce qui permet de les voir en tant que vecteurs de d dimensions, c'est à dire des points, dans l'espace de représentation.

1.1.1.c. Apprentissage et classification

Une fois l'espace de représentation établi, il est possible de travailler avec une méthode de RdF. En fonction des informations *a priori* disponibles sur le système, trois types de méthodes de RdF peuvent être utilisées : les méthodes supervisées, les méthodes non supervisées et les méthodes semi-supervisées. Lorsque les classes des formes sont connues initialement, la RdF est supervisée (Therrien, 1989). Les méthodes supervisées utilisent les formes connues, c'est-à-dire l'ensemble d'apprentissage, pour construire un classifieur qui sépare au mieux les différentes classes connues, dans le but de minimiser l'erreur de classification. Le modèle de chaque classe est alors représenté par une fonction d'appartenance qui détermine la valeur d'appartenance d'une forme à une classe. Au contraire, quand aucune information n'est disponible sur les classes d'un système, la RdF est non supervisée (Bezdek, 1981) (Frigui, 1996) (Frigui, 1997) (Vachkov, 2009). Les méthodes non supervisées, aussi appelées méthodes de coalescence, sont basées sur des fonctions de similarité. Quand des formes aux caractéristiques similaires apparaissent, elles sont classifiées dans la même classe et à l'inverse quand leurs caractéristiques sont différentes une nouvelle classe est créée par le classifieur. Une fois que le classifieur a estimé les fonctions d'appartenance des classes, les nouvelles formes peuvent être assignées à la classe pour laquelle elles ont la valeur d'appartenance maximale. Le troisième type de méthodes de RdF, concerne les méthodes semi-supervisées (Cozman, 2003) (Gabrys, 2000) (Liu, 00) qui utilisent les informations connues, c'est-à-dire les formes et classes connues, pour estimer les caractéristiques des classes et leurs fonctions d'appartenances tout en utilisant également l'apprentissage non supervisé pour détecter les nouvelles classes et apprendre leurs fonctions d'appartenance.

1.1.1.d. Environnement statique et environnement dynamique

Selon le type de système étudié, les formes obtenues sur un système peuvent être statiques, ou dynamiques lorsque leurs caractéristiques évoluent avec le temps. Une forme statique est représentée par un point dans l'espace de représentation tandis qu'une forme dynamique est représentée par une trajectoire multidimensionnelle. Dans ce dernier cas, l'espace de représentation possède une dimension supplémentaire qui est le temps (Angstenberger, 2000). De la même façon, les classes peuvent être statiques ou dynamiques. Les classes statiques sont représentées par des zones restreintes contenant des formes similaires dans l'espace de représentation. Pour ces classes, l'ordre d'arrivée des formes n'est pas lié à leur valeur d'appartenance et les paramètres du classifieur restent inchangés. De nombreuses méthodes de RdF existent pour traiter le cas des données statiques. On peut citer la méthode des K-Plus Proches Voisins (Dasarathy, 1991), les Séparateurs à Vaste Marge (SVM) (Vapnik, 1995), les méthodes Bayésiennes (Freedman, 1989), la méthode d'Analyse en Composantes Principales (ACP) (Jolliffe, 1988) (Elshenawy, 2008), la méthode Fuzzy Pattern Matching (FPM) (Cayrol, 1982) (Dubois, 1988), la méthode Fuzzy C-Means (FCM) (Bezdek, 1981), ainsi que les nombreuses déclinaisons de ces méthodes, etc. Cependant, bien souvent les systèmes sont constamment en évolution entre leurs différents modes de fonctionnement, on parle alors de systèmes évolutifs (Angelov, 2010) (Lughofer, 2009) pour lesquels il est nécessaire d'utiliser des méthodes de classification dynamique (Ben-David, 2004) (Tsybmal, 2004). Plusieurs approches de classification dynamique sont détaillées dans les parties suivantes, de même que les quatre types de RdF existantes. En effet, la combinaison des deux types de formes et des deux types de classes mène à quatre types de RdF.

1.1.2. Reconnaissance statique des formes statiques

Le premier type de reconnaissance des formes concerne les formes et les classes statiques. Il correspond au cas le plus classique de reconnaissance des formes. Les données statiques sont des données qui conservent les mêmes caractéristiques statistiques au cours du temps. Le problème correspond donc ici à la classification de nouvelles formes dont les caractéristiques ne changent pas par rapport à celles connues. Ces formes doivent être classées dans des classes statiques ne se déplaçant pas dans l'espace de représentation. Le classifieur utilise comme ensemble d'apprentissage ces formes et classes statiques, et il reste inchangé avec le temps.

Toutefois, la base de données concernant un système est souvent incomplète, tous les modes de fonctionnement et notamment les modes défectueux ou dangereux ne peuvent pas être provoqués pour obtenir des données informatives. L'apparition des nouvelles données (bruitées ou non) va permettre d'affiner les fonctions d'appartenance estimées par le classifieur. Dans ce cas, les petites adaptations de la classe sont réalisées, incrémentalement ou non, et elles permettent d'améliorer les résultats de classification. Dans certains cas, lorsqu'on peut réaliser un traitement hors ligne, il peut être plus intéressant de recommencer complètement la phase d'apprentissage en réapprenant un nouveau lot de données plus informatives.

Du point de vue statique, une forme représente l'observation du système en fonction de plusieurs paramètres à un instant donné. L'espace de représentation contient l'ensemble des formes connues observées à différents instants. Sur la figure 1.2, on peut voir une représentation de deux classes statiques à un instant $t=10$ (avec 5 formes dans chaque classe) et à un autre instant $t=11$ (où une nouvelle forme est apparue dans la classe C_2). L'espace est composé de deux attributs A_1 et A_2 , correspondant aux paramètres observés sur le système. On peut par exemple, utiliser un critère de similarité basé sur la distance entre les formes pour discriminer les classes. L'ordre et le temps d'arrivée des formes ne sont pas des informations utiles pour ce type de classification.

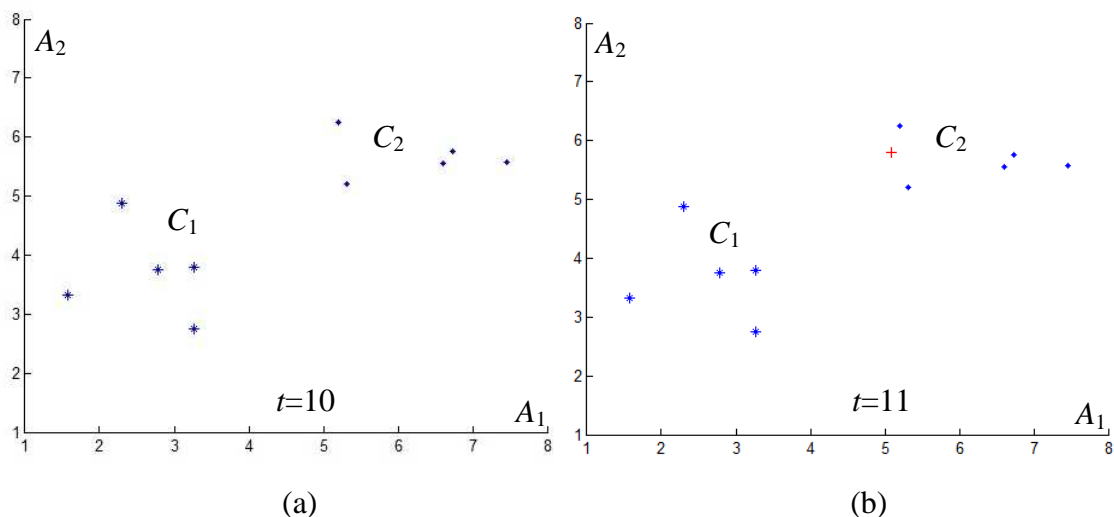


Figure 1.2. Les deux classes possèdent chacune 5 formes à $t=10$ (a). A l'instant suivant $t=11$, une seule nouvelle forme '+' est apparue dans l'une des deux classes (b).

Sur la figure 1.2, les deux attributs sont notés A_1 et A_2 . L'exemple d'espace de représentation montré sur cette figure peut être utilisé par n'importe quelle méthode statistique de reconnaissance des formes.

A l'inverse de ce premier cas possible de reconnaissance des formes, les trois cas suivants concernent la reconnaissance dynamique des formes. En effet dès qu'un système génère des formes dynamiques et/ou des classes dynamiques, il est nécessaire d'utiliser un classifieur dynamique.

1.1.3. Reconnaissance dynamique des formes statiques

Dans ce cas de classification, les classes ne sont plus statiques, mais dynamiques ; elles ont des caractéristiques qui évoluent au cours du temps. Le temps est donc un aspect important à prendre en compte pour empêcher la dégradation des performances du classifieur. Cette évolution des caractéristiques des classes peut se traduire par un déplacement, une scission, une fusion, une rotation, etc.

Ces modifications sont dues à des changements progressifs ou brutaux des modes de fonctionnement d'un système. Une évolution est brutale si le point apparaissant à l'instant $t+1$ n'est pas affecté à la même classe que le point apparaissant à l'instant t (Figure 1.3.a). Dans ce cas, la prédiction de l'évolution est impossible et aucune méthode dynamique ne pourra savoir si la classe évoluée provient d'une ancienne classe ayant réalisé un changement important de ses caractéristiques ou s'il s'agit simplement de l'apparition d'une nouvelle classe. S'il s'agit d'une nouvelle classe, une méthode de classification non supervisée ou semi-supervisée sera capable d'identifier ses caractéristiques. L'évolution progressive (Figure 1.3.b) se caractérise par l'apparition de plusieurs points intermédiaires, i.e. des formes transitoires, entre un point du début de l'évolution et un point de la fin de l'évolution. Ces points intermédiaires pourront ensuite être rejetés puisqu'ils ne représentent qu'une liaison transitoire entre la classe présente avant l'évolution et celle présente en fin d'évolution. Les changements survenant dans les caractéristiques des données peuvent être cycliques (changement de saison, d'heure, etc.) ou irréguliers (inflation, valeurs boursières, etc.). Dans ce dernier cas, les trajectoires d'évolution des classes ne sont pas connues initialement ; il est donc impossible d'inclure *a priori* dans le classifieur toutes les évolutions possibles dans l'espace de représentation. Sur la figure 1.3.b, les formes initiales de la classe sont en bleu. Ensuite, de nouvelles formes apparaissent et dérivent progressivement dans l'espace de représentation.

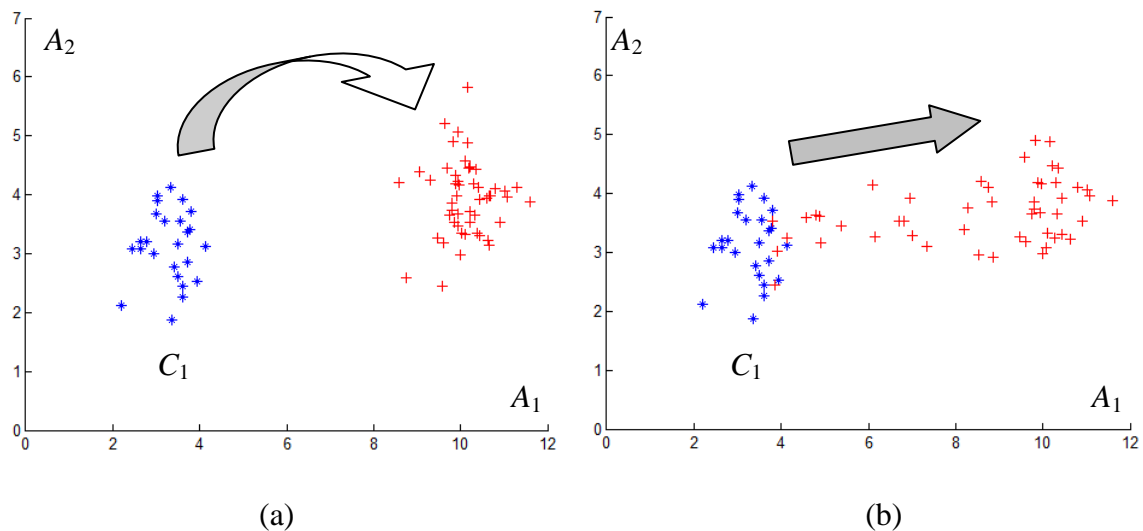


Figure 1.3. Evolution brutale d'une classe dynamique (a). Evolution progressive d'une classe dynamique (b). Les formes de l'évolution sont représentées par des '+'.

Le problème de classification posé ici est d'autant plus important que suite à leurs évolutions, les caractéristiques des classes sont mal estimées si aucun suivi ni aucune adaptation de ces dernières n'est réalisée. Dans certains cas, ce problème devient majeur ; par exemple lorsqu'une classe correspondant à un fonctionnement normal prend progressivement les caractéristiques d'une classe de fonctionnement dangereux. Le problème de classification consiste donc, pour les classes dynamiques, à détecter et à suivre les changements temporels des caractéristiques des classes. Une fois que les évolutions sont détectées et validées, il est nécessaire d'adapter le classifieur pour bien estimer les nouvelles caractéristiques des classes et pour suivre efficacement ces évolutions.

Dans le cas d'une évolution progressive, il peut être intéressant de prédire la classe finale de l'évolution. Cette prédiction permet de prendre plus rapidement une démarche corrective et d'éviter ainsi les conséquences d'un mode de fonctionnement anormal. Prévoir la classe ou le mode de fonctionnement d'arrivée revient à effectuer un pronostic ou un diagnostic prédictif (Schwabacher, 2007) (Ribot, 2009).

Les méthodes que nous proposons dans le chapitre 2 répondent à cette problématique de reconnaissance dynamique des formes statiques.

1.1.4. Reconnaissance statique des formes dynamiques

Les formes dynamiques sont représentées par une séquence temporelle d'observations, i.e. un signal. Cette représentation décrit les formes dynamiques par des trajectoires

multidimensionnelles dans l'espace de représentation. La classification statique des formes dynamiques ne se base généralement pas sur la similarité de leurs indications ponctuelles mais sur la similarité de leurs structures temporelles. Dans bien des cas réels, les informations concernant les données ne sont pas retenues pour leurs valeurs à un instant donné, mais pour leurs valeurs durant une période donnée. L'ordre d'arrivée des informations et leur évolution historique sont donc des informations importantes. Tout comme pour les formes statiques, les formes dynamiques peuvent être décrites par différents paramètres informatifs influençant leur classification.

Sur la figure 1.4, une représentation de plusieurs formes dynamiques différentes est faite en fonction du temps.

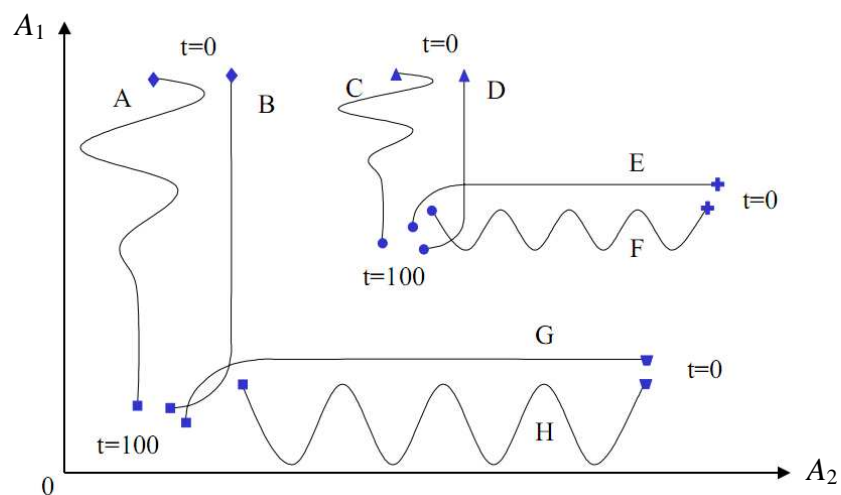


Figure 1.4. Formes dynamiques (A-H) dans l'espace de représentation en 2 dimensions (Angstenberger, 2000). Cet espace est constitué de 2 paramètres autres que le temps. La représentation temporelle est réalisée par la position des 't=0' et 't=100' dans l'espace.

Plusieurs points de vue peuvent être considérés pour distinguer ou regrouper les formes de la figure 1.4. On peut dire que deux formes dynamiques font partie de la même classe en se basant sur plusieurs critères de similarité comme par exemple :

- la même orientation, dans ce cas pour la figure 1.4 deux classes sont obtenues {A, B, C, D} et {E, F, G, H},
- une allure ou structure très proche, dans ce cas trois classes peuvent être obtenues {A, C}, {B, D, E, G} et {F, H},

- un emplacement similaire dans l'espace de représentation, i.e. une proximité forte, les classes sont alors {A, B}, {C, D}, {E, F} et {G, H},
- une combinaison de plusieurs similarités parmi celles déjà énoncées.

Ces similarités considèrent à la fois le temps, l'historique des informations et les positions des formes dans l'espace de représentation. En fonction de la précision souhaitée dans la classification, il est possible de considérer que deux formes sont différentes seulement si elles ont la même structure, localisation ou orientation dans l'espace de représentation.

Pour réaliser la classification de ces formes dynamiques, ce ne sont donc généralement pas des paramètres statistiques qui les décrivent, mais des paramètres structurels connectés les uns aux autres. Ces paramètres structurels sont obtenus en utilisant une technique de segmentation, qui va diviser chaque signal en parties représentatives. La méthode de segmentation devra trouver les points de rupture caractéristiques des parties informatives de chaque signal. Ces parties représentatives sont appelées des épisodes ; elles vont permettre d'obtenir une représentation simplifiée d'un signal, dont la structure, c'est-à-dire les liaisons entre les parties, va être conservée. Chacun de ces épisodes sera ensuite assimilé à une forme approximant au mieux la partie du signal lui correspondant. On appelle primitive la ou les formes représentant un épisode. Chaque signal va finalement être représenté par une signature, i.e. une structure caractéristique. Puis, bien souvent, cette signature va être analysée comme une chaîne de caractères dont la syntaxe va être similaire pour les formes dynamiques de la même classe. La similitude dans la syntaxe est désignée par une grammaire définie par la méthode de classification.

Dans le chapitre 3, une méthode est proposée pour traiter le problème de classification des formes dynamiques. Dans certains cas, nous verrons également qu'il peut être intéressant d'utiliser à la fois des caractéristiques statistiques et structurelles pour améliorer le taux de classification. Ce type de méthode appelé, méthode mixte, sera également présenté dans le chapitre 3.

Dans la partie suivante, le dernier type de reconnaissance des formes est présenté. Il s'agit du cas où les formes et les classes sont dynamiques.

1.1.5. Reconnaissance dynamique des formes dynamiques

Ce dernier cas de classification correspond au cas des formes dynamiques dont les caractéristiques évoluent avec le temps. La structure d'une forme dynamique va donc changer progressivement ou brutalement. Ce changement de caractéristiques va mener ces formes dynamiques à changer de classe au cours du temps. La structure d'un signal pourra être la même durant une certaine période, i.e. elle peut se répéter, puis évoluer (Figure 1.5).

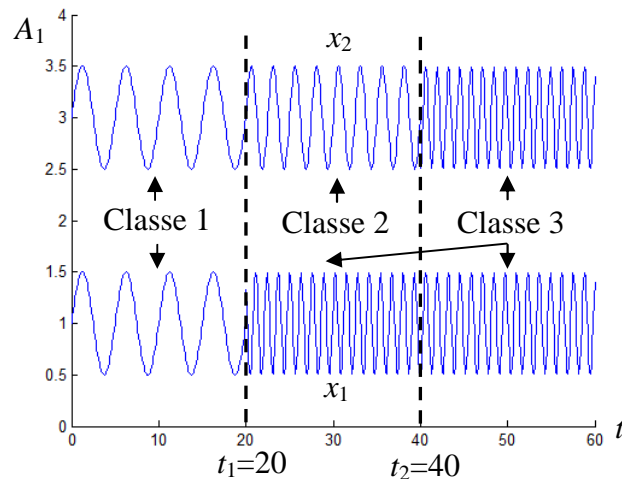


Figure 1.5. Changements de classe progressifs pour la forme dynamique x_2 et changement de classe brutal pour x_1 .

La mesure de similarité utilisée pour les formes de la figure 1.5 considère la structure de chaque forme uniquement. Les deux formes dynamiques ont la même structure avant $t=20$, elles font donc partie de la même classe (Classe 1). Puis les deux formes dynamiques changent de classe suite à un changement de leurs structures. En effet, sur la figure 1.5 on peut voir un changement de la fréquence de chaque signal à partir de $t=20$. La forme x_2 passe par une phase transitoire (Classe 2) entre les instants t_1 et t_2 pour ensuite obtenir une structure correspondant à celle de la classe 3. A l'inverse pour x_1 , le changement de classe est brutal à t_1 (Classe 3). Dans ce cas de classification, la méthode de RdF va devoir détecter un changement dans la structure des formes. Ce changement peut correspondre à une détérioration progressive ou brutale du fonctionnement d'un système. La méthode de classification dynamique doit donc dans ce cas détecter que ce changement ne fait pas partie de la structure normale d'un signal, mais qu'il y a une évolution du système. Ce type de problème est fréquent dans le domaine médical avec le suivi de différentes données concernant les patients, ou dans le domaine industriel avec le suivi de la qualité de production où chaque système est amené à avoir des changements dans ces modes de fonctionnement, qu'ils soient connus

initialement ou non. Généralement, les méthodes dynamiques de RdF traitent ce problème de classification des formes dynamiques qui changent brutalement ou progressivement en analysant l'évolution de leur structure. En effet, lors de l'obtention des signatures décrivant ces formes dynamiques, un changement doit apparaître à partir de l'apparition d'une évolution.

Dans la partie suivante, la problématique instaurée par les systèmes évolutifs est développée et des solutions sont proposées. Puis, la description de plusieurs méthodes existantes de reconnaissance dynamique des formes est réalisée.

1.2. Reconnaissance des formes dans un environnement dynamique

1.2.1. Définitions et objectifs

Les systèmes évolutifs ont des modes de fonctionnement qui peuvent passer au cours du temps d'un mode normal à un autre mode normal ou à un mode défaillant. La prise en compte du temps et de l'ordre d'arrivée des observations sont des informations importantes dans le suivi de ces systèmes. Elles permettent de suivre la progression des évolutions des modes de fonctionnement en fonction du temps. Plusieurs changements dans les conditions du système, tels qu'une fuite, l'usure d'un outil, ou un mauvais réglage, peuvent mener le système vers un mode défaillant. Les classes peuvent évoluer progressivement ou brutalement vers une nouvelle position de l'espace de représentation, en fonction des paramètres du système qui évoluent dans le temps. Quand une classe évolue dans l'espace de représentation, par exemple en cas de déplacement, rotation, fusion ou scission, la méthode de RdF doit considérer les nouvelles informations avec plus d'importance que les anciennes puisqu'elles apportent les observations les plus récentes et donc pertinentes concernant un système. Il est alors important d'oublier les informations qui ne sont plus représentatives. Ainsi, chaque fonction d'appartenance d'une classe doit être adaptée en fonction de ses changements temporels. Cela nécessite un classifieur adaptatif avec un mécanisme d'ajustement de ses paramètres. Parmi les nouvelles formes, certaines peuvent renforcer et confirmer l'information contenue dans les données déjà classifiées tandis que d'autres peuvent apporter des nouvelles informations (création, déplacement, fusion de classes, etc.). Ces nouvelles informations peuvent concerner un changement dans les conditions d'opération d'un système, le développement d'une faute ou simplement des changements significatifs dans la dynamique du système.

Le suivi et l'adaptation des classes suite à leurs évolutions permettent de bien estimer les nouvelles caractéristiques des classes. De ce fait, une meilleure estimation du mode de fonctionnement actuel peut être réalisée, évitant de mal classer les nouvelles formes. En effet,

l'estimation des caractéristiques d'une classe dans l'espace de représentation, correspond à l'estimation d'un mode de fonctionnement. Plus les caractéristiques d'une classe sont donc précises, plus le mode de fonctionnement correspondra à ce qui se passe réellement sur un système. Pour faire cette adaptation plus précise des caractéristiques des classes, leurs fonctions d'appartenance doivent être mises à jour. La partie suivante décrit le principe de mise à jour des fonctions d'appartenance des classes.

1.2.2. Principe d'adaptation des paramètres du classifieur

Le principe général des méthodes de reconnaissance dynamique des formes (Angstenberger, 2000) (Cohen, 2004) (Last, 2002) (Nakhaeizadeh, 1997) est d'observer le changement dans les propriétés statistiques des classes, dans le but de décider dans quel état le système se trouve : stable, en alerte ou en action. Ces états correspondent respectivement à aucun changement, à un changement graduel ou à un changement abrupt. Ainsi, les fonctions d'appartenance du classifieur devront rester respectivement inchangées, légèrement adaptées ou complètement changées en se basant sur les données de l'évolution. Le taux d'erreur de classification est une des propriétés les plus observées pour percevoir ces changements. Lorsqu'il est utilisé, les données sont divisées en différents groupes et leur vraie classe est connue à l'avance, de telle façon que le taux d'erreur de classification soit facile à calculer. Si ce taux augmente significativement après la classification de certaines formes, alors le système se situe dans le mode action et les paramètres du classifieur doivent être complètement recalculés.

Dans la littérature, les fonctions d'appartenance sont adaptées en agissant directement sur les paramètres du classifieur, en substituant ou en ajoutant certaines formes récentes et représentatives à l'ensemble d'apprentissage en fonction de l'état (stable, alerte, action) dans lequel le système se trouve (Angelov, 2004a) (Angstenberger, 2000) (Nakhaeizadeh, 1997) (Tong, 2001). Cette adaptation est basée sur le lot de données le plus récent sélectionné par l'utilisation :

- d'une fenêtre glissante, de taille fixe ou variable, qui permet de réduire ou limiter la taille grandissante de la base de données en acceptant seulement les n formes les plus récentes (Nakhaeizadeh, 1997) (Widmer, 1996). La taille de la fenêtre glissante doit être bien choisie pour obtenir un compromis entre une adaptation rapide et un nombre suffisant de formes représentatives,
- d'un nombre fixe de formes sélectionnées en fonction de leur âge et de leur utilité (Gibb, 1994). L'âge d'une forme ne peut pas être le seul critère de sélection d'une

forme puisque certaines formes récentes peuvent être aberrantes, peuvent correspondre à du bruit, etc. L'utilité d'une forme peut être définie par le changement auquel elle correspond sur le système. Par exemple, si un paramètre du système évolue de façon importante suite à un changement de caractéristique du système, le déplacement réalisé par les formes pour ce paramètre correspond à l'ampleur de l'évolution. Néanmoins, il est difficile d'estimer l'utilité des formes,

- d'un facteur d'oubli sur les formes les moins caractéristiques du fonctionnement d'un système. Ce facteur d'oubli peut être à taux constant ou variable. Chaque forme classée dont l'utilité est faible, voit son importance, ou sa valeur d'appartenance à une classe multipliée par un coefficient compris entre 0 (forme pouvant être oubliée) et 1.

Certaines méthodes adaptent les paramètres de leur classifieur mais également la structure de leur classifieur, ce sont les réseaux de neurones évolutifs (Amadou Boubacar, 2006) (Angelov, 2004a) (Cohen, 2005) (Lecoeuche, 2003). Dans (Angelov 2004a), la méthode evolving Takagi-Sugeno (eTS) est développée pour tenir compte des évolutions des données. Dans eTS, une fonction potentielle basée sur la distance entre les formes est définie pour les nouvelles formes. Le potentiel du premier point est considéré comme égal à un. Il établit le premier neurone (ou la première règle) qui est considéré comme le prototype (ou centre) de la première classe. Puis, les nouvelles données peuvent avoir un potentiel proche de ce neurone ou supérieur à celui-ci. Elles peuvent alors renforcer ou confirmer les informations contenues dans les neurones précédemment appris. Si la forme est plus informative que celle qui a servi à établir le prototype, alors un nouveau neurone (nouvelle règle) est ajouté. Dans (Amadou Boubacar, 2006) (Lecoeuche, 2003), le réseau de neurone est basé sur une modélisation Gaussienne multiprototype de classes non convexes. La fonction d'activation de chaque neurone caché détermine le degré d'appartenance d'une observation à un prototype de classe. Lors de l'initialisation de la méthode, la première forme permet de créer le premier prototype qui constitue la première classe. Le prototype est caractérisé par son centre et sa matrice de covariance initiale. Puis, en fonction des valeurs d'appartenance des nouvelles acquisitions, le prototype (le neurone caché) peut être adapté, éliminé ou un nouveau prototype peut être créé.

De nombreuses approches différentes sont encore développées pour suivre au mieux les changements des systèmes évolutifs. On peut citer entre autres les travaux de (Geetha, 2010) (Gil-Garcia, 2010) (Ilarri, 2008) (Nezhad, 2010) (Yang, 2010) (Bifet, 2009) (Sahel, 2007) et (Gama, 2006). Plusieurs approches parmi celles là seront décrites dans les parties suivantes.

La figure 1.6 présente la structure typique que peut avoir une méthode de classification dynamique.

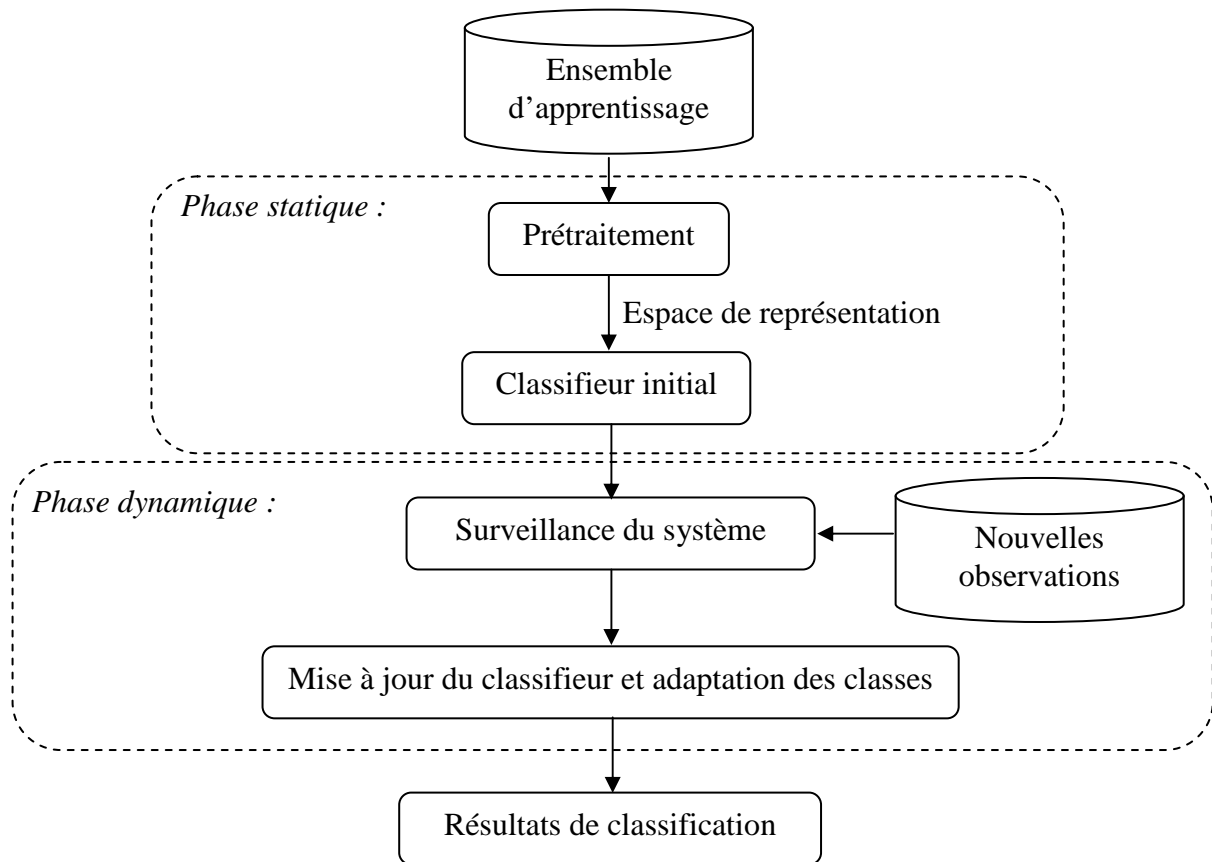


Figure 1.6. Principe d'une méthode de classification dynamique.

Dans la partie suivante, plusieurs méthodes de reconnaissance dynamique des formes sont présentées.

1.2.3. Méthodes de reconnaissance dynamique des formes

Plusieurs méthodes de reconnaissance des formes ont été développées pour intégrer au fur et à mesure les informations des nouvelles formes dans leur classifieur. Plusieurs types de mise à jour peuvent être réalisés, soit :

- par lot de données, dans ce cas la classification est arrêtée, le classifieur est mis à jour hors ligne en utilisant ces données, puis la classification peut recommencer,
- de façon incrémentale pour ajouter en ligne, chaque nouvelle information au classifieur.

Les méthodes dont le fonctionnement se fait hors ligne, n'ont pas pour vocation d'être utilisées sur des systèmes évolutifs évoluant en continu. Dans ce cas on utilise plutôt les

méthodes de classification dont le classifieur est actualisé incrémentalement. Ces méthodes permettent de mettre à jour en ligne, les fonctions d'appartenance des classes après la classification d'une ou de plusieurs formes. En effet, chaque nouvelle forme apporte son lot d'information à propos des caractéristiques des classes et donc des modes de fonctionnement d'un système. Le problème de ces méthodes est qu'elles ne permettent pas de suivre les évolutions des formes et des classes dynamiques. En effet, ces méthodes dites incrémentales ne réalisent pas l'oubli des formes non-représentatives et les nouvelles formes ne sont pas considérées avec plus de représentativité que les anciennes. Elles ne peuvent donc pas prendre en considération la notion de dérive des formes et des classes. Parmi les méthodes incrémentales on peut citer les méthodes Learn++ (Byorick, 2003), SVM Incrémentale (ISVM) (Cauwenberghs, 2000), Clustering and Classification Algorithm-Supervised (CCA-S) (Ye, 2002), K-PPV Incrémentale (IKNN) (Zhou, 2001), FPM Incrémentale (IFPM) (Sayed Mouchaweh, 2002), arbre de Hoeffding incrémental à décision très rapide (VFDT) (Domingos, 2000) etc. Ces méthodes ont été appliquées et évaluées sur différentes applications.

Pour parfaire ces méthodes et les rendre compatibles avec les systèmes évolutifs, plusieurs méthodes de classification dynamiques ont été développées. Comme il a été dit précédemment, ces méthodes tiennent compte du temps. Leur utilisation en ligne permet d'intégrer les nouvelles données (apprentissage adaptatif) après les avoir sélectionnées de différentes manières (fenêtre glissante, indice de représentativité, lot de données, etc.). Puis, l'adaptation du classifieur est réalisée afin d'estimer au mieux les nouvelles caractéristiques des classes. Les méthodes de classification dynamique utilisent différents paramètres (noyau, seuils, mesures, etc.) et différentes données (binaire, numérique, etc.), c'est pourquoi il est intéressant d'utiliser et donc de développer différentes méthodes afin d'obtenir celle donnant les meilleurs résultats pour un problème particulier.

Dans les parties suivantes nous nous intéressons à plusieurs types de méthodes de classification dynamique des formes statiques et nous détaillons leur principe de fonctionnement ainsi que leurs avantages et inconvénients.

1.2.3.a. Réseaux de neurones

Les réseaux de neurones sont des méthodes connexionnistes dont la structure est organisée en différentes couches reliées les unes aux autres. Ils sont basés sur une représentation mathématique (statistique) simplifiée des neurones biologiques ; on parle alors de l'utilisation de neurones formels. Les réseaux de neurones possèdent une couche d'entrées, une ou

plusieurs couches cachées et une couche de sortie. La couche d'entrées correspond aux vecteurs formes de l'espace de représentation. La (ou les) couche cachée comprend les prototypes des classes. La couche de sortie définit le nombre de classes connues et permet de connaître le degré d'appartenance d'une observation à une classe. Pour chaque nouvelle observation présentée à l'entrée d'un réseau de neurones, les degrés d'appartenance de cette observation sont évalués par rapport aux différents prototypes des classes. A la suite de la phase d'apprentissage, le réseau dispose d'un ensemble d'observations classées, et d'observations écartées par ambiguïté. Plusieurs versions incrémentales des réseaux de neurones ont été développées : evolving Neural Network (eNN) (Angelov 2004b), SIMilarity MAPping (SIMAP) (Medina, 2009), General Fuzzy Min-Max Neural Network (GFMMNN) (Gabrys, 2000), fuzzy Adaptive Resonance Theory MAPs (ARTMAP) (Grossberg, 1988), etc. Ces réseaux ont une structure évolutive, c'est-à-dire que leur nombre de neurones et de connexions change en utilisant les nouvelles informations. Avant de détailler plusieurs réseaux de neurones dynamiques, nous allons commencer par détailler la structure incrémentale utilisée dans SIMAP.

Dans l'architecture de SIMAP, le but est de trouver des liaisons entre les entrées (les formes x_i) et les sorties (les classes z_i) lors de la phase d'apprentissage et également lors de la phase de classification. Ces liaisons sont réalisées à l'aide de l'estimation des poids reliant les formes aux prototypes des classes. Chaque prototype appartient à une seule classe. Les poids des connexions sont estimés à l'aide de mesures de similarité et de plusieurs cycles d'optimisation des paramètres. Tous les éléments d'une même classe doivent avoir un niveau significatif de similarité. Pour ce faire, une forme est assignée à une classe en fonction de sa similarité par rapport à l'ensemble des prototypes des classes. Cette mesure de similarité peut engendrer la création d'une nouvelle classe et l'agrandissement du réseau avec l'ajout d'un prototype de classe si ceux existants ne sont pas suffisamment adaptés à la nouvelle forme. Dans la phase d'apprentissage, on vérifie la liaison entre le prototype correspondant à une donnée et chaque classe. Si la connexion existe le prototype et la classe sont actualisés en utilisant la nouvelle donnée ; sinon on ajuste le niveau de la mesure de similarité et on génère un nouveau prototype. Ce prototype peut alors engendrer une nouvelle classe. Dans la phase d'optimisation des paramètres, chaque forme est classée en utilisant les valeurs de similarité des prototypes, les poids et les connexions des couches cachées. Les résultats de classification dépendent également du type de fonction d'appartenance utilisé (rectangulaire, gaussien, trapézoïdal, etc.). Dans SIMAP, les nouvelles informations sont intégrées en ligne, de même que l'évolution des poids. Ces mises à jour permettent d'améliorer les fonctions

d'appartenance des classes mais aucune adaptation des classes n'est réalisée pour traiter les cas de déplacement, fusion, scission, etc. C'est pourquoi les réseaux de neurones eTS amélioré, OLIN et AUDyC qui vont être détaillés apportent une amélioration aux réseaux de neurones concernant la classification dynamique.

evolving Takagi-Sugeno (eTS) amélioré :

(Lughofer, 2009) propose une amélioration de la méthode eTS initialement développée par (Angelov, 2004a). Pour surveiller la détection des évolutions, les auteurs utilisent deux indicateurs, l'âge des classes et des règles (les poids des connexions), et leur utilité au cours du temps. Le premier indicateur suit l'arrivée des formes dans les classes existantes. Si une nouvelle forme est classée dans une classe, cette dernière est mise à jour. Les autres classes n'ayant pas reçu la nouvelle forme ont leur âge qui augmente progressivement. L'âge d'une classe correspond au temps accumulé depuis la dernière donnée reçue par cette classe. Pour suivre l'évolution de cet indicateur, son gradient et sa dérivée seconde sont observés en ligne. Un changement de pente égal à un seuil défini doit alors indiquer un changement important de l'âge des classes. Le second indicateur est défini comme l'utilité accumulée des règles floues depuis leur création. Elle est calculée par comparaison de chaque règle envers l'ensemble des règles floues. En fonction de la valeur de la dérivée première de cet indicateur, (Lughofer, 2009) détecte qu'une évolution de classe est en cours ou que cette classe (ou l'un de ses prototypes) est devenue obsolète.

Suite au suivi de ces détections, (Lughofer, 2009) réalise l'adaptation des classes par une approche d'oubli progressif. Cette adaptation est réalisée en fonction de la valeur des deux indicateurs d'évolution utilisés par la méthode. Elle permet de réaliser le suivi d'une classe au cours du temps et de mieux estimer les caractéristiques actuelles de cette dernière. Le problème de la méthode vient de la définition des paramètres du facteur d'oubli et des seuils de validation des évolutions.

On-Line Information Network (OLIN) :

La méthode OLIN (Last, 2002) se base sur l'utilisation d'un réseau de neurones comportant autant de couches qu'il y a d'attributs dans l'espace de représentation. Elle permet d'optimiser le nombre d'attributs à utiliser ainsi que la structure générale du réseau et des données conservées. Le système répète initialement son optimisation en utilisant les données de l'ensemble d'apprentissage. Il adapte dynamiquement la taille de la fenêtre d'apprentissage et la fréquence de reconstruction du réseau en fonction de l'importance des évolutions. Pour ce faire le système est composé de trois modules :

- le module d'apprentissage qui permet de construire initialement le réseau de neurones et de le mettre à jour,
- le module de classification qui utilise le réseau de neurones actuel pour classer les nouvelles données,
- le module de méta-apprentissage, i.e. module de validation, qui contrôle les opérations réalisées par le module d'apprentissage.

C'est ce dernier module de validation qui permet également la sélection des données informatives sur l'évolution. Il étudie les données d'apprentissage, les attributs sélectionnés, l'entropie des données, etc., pour mettre à jour la taille de la fenêtre d'apprentissage et le nombre de données à sélectionner pour valider une nouvelle classe (ou un nouveau prototype). La taille de la fenêtre est définie initialement par une taille minimale. Le modèle actuel d'une classe est conservé tant qu'il n'y a pas d'évolution. Si aucune évolution n'est détectée, la taille de la fenêtre de sélection des données d'évolution est augmentée. À l'inverse si une évolution est détectée la fenêtre est réduite. La détection d'une évolution se fait en fonction de la différence maximale entre l'erreur de classification obtenue lors de la phase d'apprentissage et l'erreur de validation calculée en ligne à partir de l'approximation de la distribution binomiale des données. La méthode permet ainsi l'adaptation de la classe actuelle des observations en tenant compte des évolutions. Le problème de la méthode OLIN vient de la taille de la fenêtre qui peut vite devenir importante. (Last, 2002) préconise de limiter cette taille de fenêtre puisqu'elle influence également beaucoup le temps de calcul pour la classification d'une forme.

Auto-Adaptive & Dynamical Clustering (AUDyC) :

Cette méthode, développée par (Lurette, 2003) et (Amadou Boubacar, 2005b), est basée sur un réseau de neurones structuré en trois couches. La structure d'AUDyC correspond à une organisation multiprototype permettant de caractériser les classes de formes complexes à l'aide d'un ensemble de sous-classes gaussiennes. La couche d'entrée d'AUDyC représente les observations de l'espace de représentation en fonction de l'ensemble des attributs utilisés. La couche cachée représente les prototypes des classes et la couche de sortie correspond aux classes connues. Ces deux couches sont évolutives au niveau de leurs connexions et de leur nombre de neurones. La couche cachée est entièrement connectée à la couche d'entrée à l'aide d'arcs pondérés. L'utilisation des poids en fonction de chacune des dimensions de l'espace de représentation permet d'obtenir un prototype pour chaque neurone j de la couche cachée.

Dans la couche cachée, on appelle fonction d'activation d'un neurone j la fonction d'appartenance μ_j correspondant au prototype j . Le choix de cette fonction d'appartenance est libre. AUDyC intègre une fonction d'appartenance gaussienne utilisant la distance de Mahalanobis. La couche de sortie est reliée à la couche cachée afin d'associer les prototypes existants aux classes connues. Ces liaisons sont réalisées à l'aide de valeurs binaires déterminant ou non l'appartenance d'un prototype à une classe.

Initialement la méthode commence par apprendre une première observation pour créer le premier prototype et la première classe. Il est alors nécessaire de définir la valeur initiale de l'écart type de la classe par rapport à chaque dimension. Puis, les autres formes sont apprises et les valeurs des poids sont affinées.

Dans la phase de classification, le degré d'appartenance d'une forme x aux classes et aux prototypes est comparé à deux seuils μ_{\min} et μ_{\max} . μ_{\min} représente le seuil minimum d'appartenance d'une forme à une classe et μ_{\max} est le seuil minimal d'appartenance à un prototype. Chaque nouvelle forme est classée selon ses valeurs d'appartenance. Soit :

- la forme est proche d'un prototype existant ($\mu(x) \geq \mu_{\max}$), une mise à jour du prototype est réalisée (Figure 1.7.a),
- la forme est proche d'une classe connue mais sa valeur d'appartenance est faible ($\mu_{\min} \leq \mu(x) < \mu_{\max}$). Dans ce cas, un nouveau prototype est créé (Figure 1.7.b),
- la forme est loin des classes connues ($\mu(x) < \mu_{\min}$), ce qui engendre la création d'une classe et de son premier prototype (Figure 1.7.c).

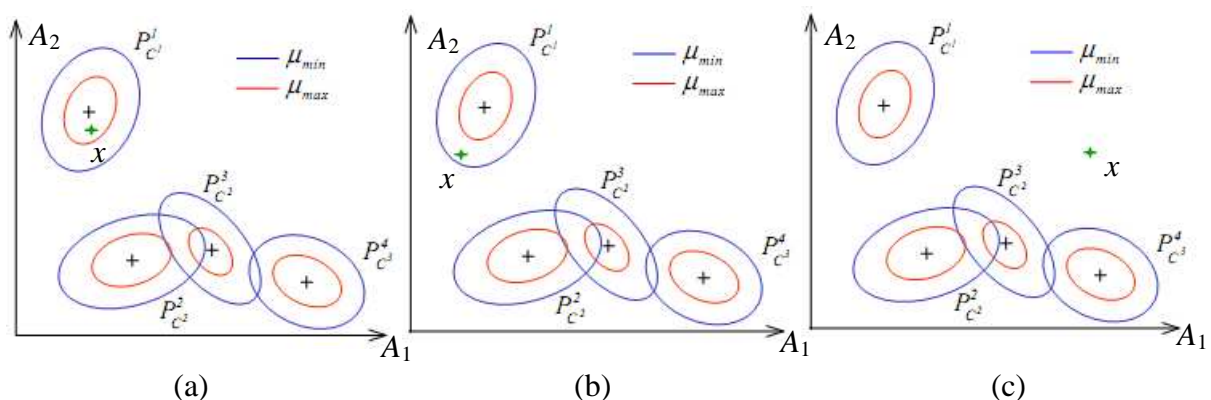


Figure 1.7. Classification d'une forme par AUDyC (Lurette, 2003).

Si des prototypes d'une même classe ont des fonctions d'appartenance qui se chevauchent et si un certain nombre de points apparaissent dans cette zone d'ambiguïté, les prototypes des classes sont fusionnés. Si une nouvelle forme ne peut être classée en fonction d'une valeur d'appartenance ambiguë entre plusieurs classes, elle est placée dans un ensemble d'ambiguïté qui sera analysé par la suite pour vérifier l'éventuelle fusion entre deux classes. Cette vérification se fait en fonction du nombre de formes ambiguës pouvant apparaître entre deux classes. AUDyC permet également l'élimination des classes et des prototypes peu représentatifs à l'aide d'un critère de cardinalité. Une étape de scission des classes a été ajoutée à la méthode par (Amadou Boubacar, 2006) afin de diviser une classe en deux lorsque c'est nécessaire. Cette étape utilise la mesure de Fisher pour mesurer la proximité des prototypes. La scission a lieu si deux prototypes ont une mesure les séparant supérieure à un seuil donné.

Cette méthode permet l'adaptation des classes au cours du temps. Elle intègre plusieurs mécanismes afin de tenir compte de l'évolution des caractéristiques des classes évolutives. Cependant, de nombreux paramètres doivent être définis pour utiliser la méthode et plusieurs d'entre eux ont une importance majeure dans ses résultats. Ces paramètres sont entre autres les seuils d'appartenance minimum et maximum aux classes et aux prototypes, la valeur initiale de la matrice de covariance des prototypes, les seuils de fusion et de scission et les paramètres de robustesse (seuil d'ambiguïté, seuil de cardinalité).

Parmi les autres réseaux de neurones ayant des capacités d'adaptation des classes évolutives, on peut citer les méthodes Cluster Detection and Labeling (CDL) (Eltoft, 1998), Fuzzy Min-Max Clustering (FMFC) (Simpson, 1993), Evolving Self-Organizing Map (ESOM) (Deng, 2003), etc. Cependant, la plupart de ces méthodes conservent l'ensemble des données dans l'espace de représentation.

1.2.3.b. Méthodes à noyau

Les méthodes d'apprentissage à noyau sont des méthodes non paramétriques qui permettent de modéliser des formes complexes en se basant sur la théorie d'apprentissage statistique sans *a priori* sur la distribution de données. Elles se basent sur la minimisation des erreurs de Vapnik (risque empirique) mais aussi sur le choix de modèles simples et adaptés à différentes types de classes. Les méthodes à noyaux aboutissent à un problème de minimisation convexe, sous forme duale ou le plus souvent quadratique dont la résolution est

rapide. Parmi les noyaux généralement utilisés on peut citer le noyau gaussien, le noyau d'ondelette, le noyau polynomial, etc.

Pour commencer, nous allons expliquer le principe des méthodes à noyau les plus utilisées actuellement : les Séparateurs à Vaste Marge (SVM). SVM est une méthode fortement développée dans la littérature (Vapnik, 1995) (Guigue, 2005) (Suykens, 1999) (Rüping, 2001) (Dufrenois, 2009) qui est adaptée à la classification en ligne. De nombreuses versions de SVM existent afin de s'adapter à toutes sortes de problèmes en utilisant différents paramètres et optimisations. On peut citer les SVM linéaires, dont les vecteurs de supports permettent de discriminer les classes en définissant une frontière de décision linéaire (Figure 1.8) et les SVM non linéaires adaptés aux classes dont la séparation ne peut pas être linéaire sans transformation de l'espace de représentation (Boser, 1992).

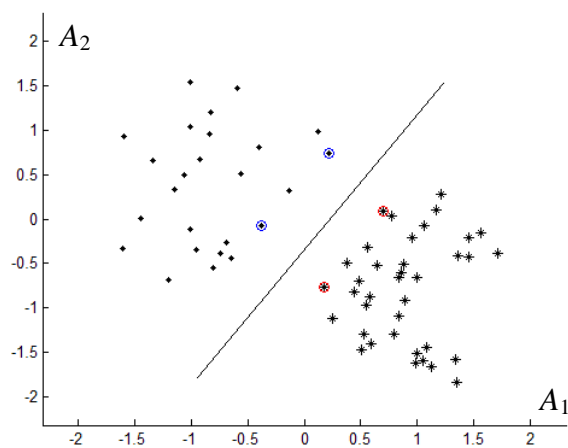


Figure 1.8. Frontière de décision linéaire obtenue par la méthode SVM. Les vecteurs de support, utilisés pour la détermination des frontières de décision, sont représentés entourés d'un cercle.

Les SVM sont des méthodes supervisées qui permettaient initialement de prendre une décision binaire, c'est-à-dire de réaliser la classification d'une forme dans un espace de décision composé de deux classes au maximum. Cependant, des versions multi-classes ont été développées afin de considérer les cas plus complexes comprenant plus de deux classes. Lorsque plusieurs classes sont présentes, le problème de classification est sous-traité par plusieurs SVM. Il existe la stratégie « un contre tous » (Guermeur, 2002) (Borer, 2003) où l'on construit autant de SVM qu'il y a de classes ; on cherche alors à distinguer les données d'une classe de celles de toutes les autres classes. Une autre stratégie appelée « un contre un » (Salomon, 2001) consiste à construire autant de SVM qu'il y a de paires de classes.

Pour classifier les formes, les méthodes SVM cherchent l'hyperplan, i.e. la frontière de décision, qui maximise la distance entre les points de chaque classe. Dans la phase d'apprentissage, la méthode SVM calcule une fonction f qui lie chaque point de l'ensemble d'apprentissage à une classe. Cette fonction f définit l'hyperplan séparateur. L'hyperplan est basé sur un certain nombre de points caractéristiques de la séparation des classes, les vecteurs de supports. Le nombre et le choix des vecteurs de supports sont estimés en minimisant le critère d'optimisation des paramètres. L'optimisation des paramètres passe par une formulation duale du problème, généralement plus simple à traiter. A partir de cette formulation, les coefficients d'optimisation (multiplicateurs de Lagrange) sont obtenus par minimisation d'une fonction objective quadratique. Cette phase d'optimisation est réalisée initialement dans la phase d'apprentissage puis elle peut être poursuivie en ligne. Dans le cas non linéaire, les SVM réalisent une transformation de l'espace de représentation initial en un espace de Hilbert comprenant plus de dimensions. Il est alors nécessaire d'utiliser d'autres types de noyau (gaussien, polynomial, etc.) pour obtenir une frontière de décision non linéaire par rapport à l'espace initial mais linéaire par rapport au nouvel espace.

Cette mise en œuvre permet aux SVM d'être plus résistants que certaines méthodes de classification aux problèmes de dimensionnalité et de bruit. Il peut être reproché à la méthode SVM de reposer uniquement sur des vecteurs de support (données existantes) pour chercher à séparer les classes. De même, la complexité des méthodes SVM est importante et le choix de leurs paramètres initiaux (dont le noyau) est difficile et change complètement les résultats.

Après avoir défini le principe général des SVM, nous allons maintenant détailler une version améliorée de SVM et la méthode SAKM qui sont des méthodes à noyau permettant de suivre les évolutions des classes.

Version dynamique des Séparateurs à Vaste Marge (SVM) :

Dans cette partie, nous nous intéressons à la méthode SVM proposée par (Boukharouba, 2008), mettant à jour les paramètres du classifieur et les caractéristiques des classes. Cette version de SVM est utilisée afin d'améliorer la définition de la séparation des classes en intégrant les nouvelles informations et en supprimant les plus anciennes. Les vecteurs de supports initiaux de cette méthode sont obtenus de la même manière que pour la méthode SVM statique. Cette méthode satisfait les conditions de Karush Kuhn et Tucker (KKT) qui permettent d'obtenir trois types de données : les vecteurs de supports, les formes situées dans la classe à l'intérieur de la zone délimitée par les vecteurs de supports, et les formes situées en dehors de cette zone. Quand une nouvelle donnée est disponible, elle correspond à un de ces

trois types. Si une nouvelle forme est classée dans la zone délimitée par les vecteurs de supports, alors la forme est bien classée mais aucun changement dans les vecteurs de supports ou dans la frontière de décision n'est nécessaire. A l'inverse, si la nouvelle donnée est un vecteur de support ou une forme située en dehors de la classe, i.e. hors de la zone délimitée par les vecteurs de support, elle est utilisée pour mettre à jour le modèle actuel du classifieur. Durant cette mise à jour, les anciennes données voient leurs poids diminués. Si le poids d'un vecteur de support atteint la valeur zéro, celui-ci n'est plus considéré comme un vecteur de support. Lors de cette mise à jour, il est nécessaire de supprimer des anciennes données. Le nombre de données à supprimer est déterminé expérimentalement à l'initiation de la méthode en fonction de la dynamique des données. Plus les données évoluent rapidement, plus le nombre de données à supprimer est important pour changer les caractéristiques d'une classe.

(Boukharouba, 2008) a proposé des mécanismes de fusion et d'élimination afin d'améliorer l'estimation des classes. La fusion est détectée quand un nombre de données ambiguës est présent entre plusieurs classes. Dans ce cas :

- la classe possédant le plus de données parmi les classes à fusionner est entièrement conservée,
- les formes des autres classes sont supprimées à l'exception de leurs vecteurs de supports qui sont conservés,
- les vecteurs de supports conservés sont classés un par un comme des nouvelles formes de la classe possédant le plus de données.

Ainsi, la classe la plus représentative de celles à fusionner est conservée, mais elle est adaptée en tenant compte des formes les plus caractéristiques des autres classes à fusionner, i.e. leurs vecteurs de supports. Un exemple de fusion réalisée par cette méthode est présenté sur la figure 1.9.

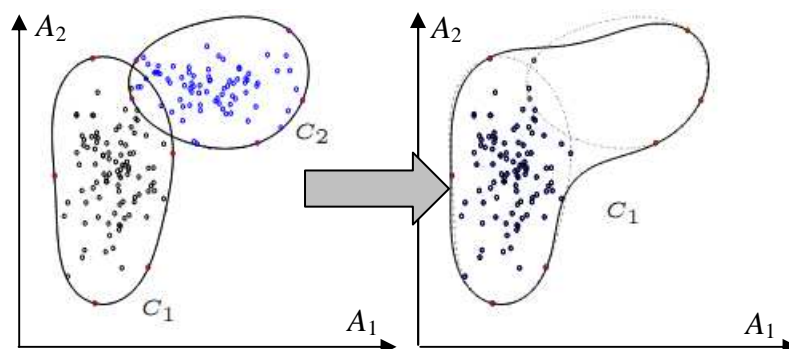


Figure 1.9. Deux classes possédant des données ambiguës sont fusionnées.

La procédure d'élimination des classes obsolètes consiste en l'élimination des classes qui contiennent peu de données après un temps t . Ainsi, le suivi des classes est réalisé, la mise à jour des vecteurs de supports permet de continuer à bien séparer les classes et les anciennes formes sont supprimées. Parmi les contraintes propres à cette méthode, on peut citer la définition du nombre de données anciennes à supprimer, la définition des seuils pour les données ambiguës et les données obsolètes et l'absence de mécanisme concernant la scission d'une classe.

Self-Adaptive Kernel Machine (SAKM) :

La méthode SAKM (Amadou Boubacar, 2005a) est basée sur la méthode SVM. Elle intègre une structure reposant sur une architecture neuronale évolutive où chaque neurone caché correspond à un vecteur de support. La méthode se décompose en trois phases :

- pour chaque nouvelle donnée, une mesure de similarité permet de choisir la phase d'apprentissage adaptée qui suit,
- la phase d'apprentissage peut consister en la création d'une nouvelle classe, en la mise à jour d'une classe connue (adaptation), ou en la fusion de plusieurs classes,
- puis la phase d'évaluation permet de vérifier la représentation des classes créées, de séparer ces dernières si nécessaire (scission) et d'éliminer les classes obsolètes.

Elle modélise les classes de la partition dynamique en s'appuyant sur les fonctions d'apprentissage à noyau représentées par des hyperplans dans l'espace de Hilbert. Le critère de similarité de SAKM exploite la propriété du noyau choisi. Il consiste à mesurer la distance entre chaque nouvelle donnée et le vecteur support le plus proche pour chaque classe dans l'espace d'Hilbert. En fonction de sa position par rapport à chaque classe et à leur contour, la forme peut être rejetée ou classée comme nouvelle forme de la classe ou comme nouveau vecteur de support. Une marge de tolérance est utilisée lors de la classification, elle permet de tenir compte des erreurs de classification réalisées par la méthode lors de la phase d'apprentissage. C'est durant cette phase que les formes et les classes sont progressivement apprises. Le premier vecteur de support est initialement le premier point de chaque classe puis la procédure d'adaptation permet de modifier ces vecteurs de supports ainsi que la représentation locale des classes. En effet, un poids est défini pour chaque forme et durant la procédure d'adaptation de SAKM, ces poids sont mis à jour en utilisant la technique du gradient stochastique. Seul un nombre maximal de vecteurs de supports est contenu dans chaque classe. Par conséquent, lors de la procédure d'adaptation, si un nouveau vecteur de

support est défini, le plus ancien (poids le plus faible) sera supprimé. La procédure de fusion de SAKM est basée sur les données ambiguës se trouvant entre les classes. Cette procédure consiste à sélectionner les classes qui doivent fusionner puis à réinitialiser ces classes en utilisant l'ensemble des nouvelles données sélectionnées pour la fusion. La procédure de scission de SAKM consiste à étudier de façon heuristique la proximité des groupes de formes sur l'ensemble de la partition des données. Cependant, cette procédure est jugée trop coûteuse en temps par les auteurs et elle n'est donc pas utilisable pour les applications en ligne. La procédure d'élimination des classes obsolètes repose sur un critère de cardinalité.

La méthode réalise ainsi le suivi des classes et la mise à jour des vecteurs de supports tout en réalisant certains mécanismes de mise à jour de l'ensemble des données (élimination, fusion). Les limites de la méthode viennent de la difficulté à définir certains de ses nombreux paramètres (noyau, seuil de similarité, ratio d'apprentissage, etc.).

1.2.3.c. Arbres de décision

Ces méthodes sont des algorithmes de classification qui prennent une décision en fonction du chemin parcouru par une nouvelle forme dans un arbre. Un arbre est composé d'une racine, de nœuds, de branches et enfin de feuilles. Les branches sont les connexions reliant la racine, les nœuds et les feuilles. Chaque nœud d'un arbre de décision représente un test réalisé par rapport à un attribut. Chaque branche parcourue à partir d'un nœud correspond à une sortie possible du test réalisé. Enfin, chaque feuille, i.e. chaque terminaison d'une branche d'un arbre, contient l'estimation d'une classe. Pour classifier une nouvelle forme, la forme doit parcourir un chemin passant par la racine de l'arbre de décision pour aller jusqu'à une feuille. En fonction de l'arbre de décision développé, les tests contenus dans les nœuds et les feuilles peuvent être des règles de classification plus ou moins complexes. Lors de la phase de création d'un arbre, les feuilles sont progressivement remplacées par des nœuds. L'attribut qui est testé à un nœud est choisi par comparaison de tous les attributs disponibles en fonction de certaines mesures heuristiques. Parmi les différentes sortes d'arbres existantes (Bifet, 2009) (Hulten, 2001), nous allons développer ici une approche dynamique basée sur l'arbre de Hoeffding : *Concept-adapting Very Fast Decision Tree (CVFDT)* (Hulten, 2001).

Concept-adapting Very Fast Decision Tree (CVFDT) :

Dans (Hulten, 2001), les auteurs choisissent de développer un arbre de Hoeffding pour suivre les évolutions de la distribution des données. Les arbres de Hoeffding sont des arbres de décision reposant sur le support mathématique de la limite de Hoeffding. Cette limite

quantifie le nombre de formes que le classifieur doit apprendre afin de bien estimer certaines statistiques et ainsi bien classifier les nouvelles formes. CVFDT (Hulten, 2001) est un arbre de décision basé sur la méthode VFDT (Domingos, 2000) avec un concept d'adaptation de sa structure pour suivre au mieux l'évolution des données. Il utilise une fenêtre glissante de taille fixe pour mettre à jour le classifieur. En effet, c'est en fonction des données contenues dans cette fenêtre qu'une évolution peut être détectée. Chaque nouvelle forme à classifier est incluse dans la fenêtre glissante et la donnée la plus ancienne est supprimée. Quand une nouvelle donnée est classée à l'aide des tests réalisés pour chaque nœud, ces derniers sont mis à jour et le nombre de données liées à ces nœuds est incrémenté. De la même façon lorsque des anciennes données sont supprimées, les nœuds liés à ces données sont décrémentés. Si durant cette mise à jour, les données évoluent, la structure de l'arbre doit alors changer. En effet, l'importance qui était donnée à une branche lors de sa construction, a désormais changé. Dans ce cas, un nouveau sous arbre est créé avec le nouvel attribut le plus représentatif à sa racine. Une fois ce sous arbre devenu plus précis pour la classification des données, l'ancien arbre est complètement remplacé par le nouveau. La vérification de la structure de l'arbre et de l'information fournie par chaque nœud est analysée périodiquement après la classification d'un nombre de données défini. CVFDT est capable de suivre les changements de distribution des données en renouvelant sa structure en continu. Le problème de la méthode est qu'elle ne peut suivre qu'une seule évolution des données à la fois. De plus, certains paramètres tels que la période de vérification de la structure de l'arbre, ou la taille de la fenêtre doivent être définis initialement et ne sont pas adaptés en ligne.

1.2.3.d. Autres méthodes de reconnaissance dynamique des formes

Parmi les autres méthodes de classification réalisant la mise à jour des classes et certaines adaptations (fusion, scission, etc.), on peut citer FCM Adaptative (Marsili-Libelli, 1998), la méthode de coalescence floue adaptative développée par (Angstenberger, 2000), la méthode AddC (Guedalia, 1999), les méthodes basées sur ISODATA (Ball, 1965), etc. Quelques unes de ces méthodes ont leur principe expliqué dans les parties qui suivent.

Fuzzy C-Means (FCM) Adaptative :

Cette méthode est basée sur la méthode FCM (Bezdek, 1981) déjà bien expliquée dans la littérature. Un mécanisme a été développé afin d'intégrer en ligne les nouvelles données et d'adapter les classes. Suite à la phase d'apprentissage (des classes et de leurs prototypes), chaque nouvelle forme peut être classée en calculant sa valeur d'appartenance par rapport à

chaque classe. Le mécanisme proposé par (Marsili-Libelli, 1998) permet de mettre à jour de façon incrémentale les classes seulement quand se sont uniquement des informations représentatives qui sont classifiées. Par exemple, des données bruitées ne devraient pas modifier les caractéristiques d'une classe. Pour cela, (Marsili-Libelli, 1998) calcule l'entropie d'une classe et en fonction de son évolution suite à la classification d'une donnée, il choisit ou non de mettre à jour les caractéristiques de la classe. Dans le cas où l'évolution de l'entropie d'une classe ne dépasse pas une augmentation autorisée, la classe est adaptée. De la même façon, une entropie évoluant négativement, signifie que le partitionnement en classe devient plus discriminatif et dans ce cas la mise à jour de la classe est également réalisée. Suite à cette mise à jour, un rafraichissement des données conservées par le classifieur est réalisé. C'est une fenêtre glissante de taille fixe qui lui permet de limiter le nombre de données, et de remplacer la plus ancienne par la nouvelle. En conservant seulement les faibles évolutions d'entropie d'une classe, l'auteur considère seulement les déplacements des classes qui sont très progressifs. De la même façon aucune procédure n'est mise en œuvre pour la scission ou la fusion de classe. Cependant, étant une méthode non supervisée, la méthode FCM Adaptative permet de prendre en compte la création de nouvelles classes.

AddC :

Dans (Guedalia, 1999), l'auteur propose une approche non supervisée pour la classification des données non statiques. L'approche proposée consiste à :

- considérer le premier point à classer comme le premier prototype, et à classer les nouvelles données,
- si une nouvelle donnée est trop éloignée des prototypes connus un nouveau prototype de classe est créé,
- le prototype le plus proche d'une nouvelle donnée est considéré comme son représentant. Chaque prototype est désigné comme le représentant d'un certain nombre de points. Ces points sont donc considérés comme appartenant à une même classe,
- après la classification d'un point, le prototype qui le représente est mis à jour incrémentalement,
- seul un nombre maximum de prototypes est autorisé. Quand deux prototypes sont trop près on les fusionne (position dans l'espace et nombre de formes qu'ils représentent).

Cette approche ressemble fortement à la méthode des C-Moyennes (MacQueen, 1967) consistant à minimiser l'inertie intraclasse et à maximiser l'inertie interclasse, seulement cette méthode permet une approche dynamique puisque chaque nouvelle donnée qui peut potentiellement représenter une nouvelle classe est directement traitée comme telle. Ainsi l'approche proposée permet de détecter les évolutions des classes (déplacements progressifs et brutaux) et de faire leur mise à jour. Il est cependant important de bien définir les paramètres de la méthode (seuil de fusion, nombre de prototypes autorisés, etc.) pour obtenir les meilleurs résultats.

Méthode de coalescence floue adaptative :

(Angstenberger, 2000) a développé une méthode non supervisée floue permettant de traiter le cas des données non statiques. Elle est basée sur la méthode Unsupervised Optimal Fuzzy Clustering (UOFC) (Gath, 1989). La méthode proposée utilise une représentation des classes par prototype à l'aide de trois modules réalisant : la surveillance d'un système, l'adaptation du classifieur et l'adaptation des données d'apprentissage. La procédure de surveillance consiste en l'application de plusieurs algorithmes heuristiques permettant la reconnaissance des changements abrupts, ainsi que la fusion ou la scission de classes. La phase d'adaptation est contrôlée par une mesure de validité qui est utilisée comme un indicateur de la qualité de l'estimation des classes. L'adaptation des classes est réalisée si la mesure de validité montre que leur estimation peut s'améliorer après adaptation. Seulement il est nécessaire d'adapter cette mesure de validité sur le partitionnement actuel des classes et sur le partitionnement des classes après adaptation. Seul le meilleur partitionnement est alors conservé. La détection des nouvelles classes est réalisée à l'aide d'une mesure de distance. En fonction de la valeur d'appartenance d'une forme à une classe, la forme sera considérée comme potentiellement représentative d'une nouvelle classe ou elle sera intégrée à la classe connue. Dans le premier cas, il sera nécessaire de valider les deux critères suivants pour créer une nouvelle classe :

- un nombre suffisant de points non classés dans les classes connues doit être disponible. Ce nombre de points minimum est défini comme le nombre de points moyen de chaque classe ou comme le nombre minimum de points représentant les classes actuelles,
- les points potentiellement représentatifs de la nouvelle classe doivent être suffisamment similaires. Pour ce faire, les points doivent valider un critère de compacité.

C'est en fonction de la compacité des points que le prototype d'une classe est sélectionné. Un seuil de compacité minimum doit ainsi être défini initialement. La fusion des classes est réalisée en fonction du chevauchement de ces dernières calculé à partir des valeurs d'appartenance des formes qui sont ambiguës entre deux classes. La scission des classes est réalisée lorsque la distribution des données n'est plus homogène pour une classe. Les évolutions progressives des données sont considérées en fonction de l'évolution des valeurs d'appartenance des données déjà classées. En effet, si une classe évolue, ses caractéristiques vont changer et par conséquent certaines données vont devenir plus représentatives (déplacement dans leur sens) et d'autres deviendront moins informatives (éloignement de ces données suite à un déplacement). Quand une évolution est détectée, seules les formes conservées dans une fenêtre glissante sont utilisées pour estimer les nouveaux centres des classes. Une forme est supprimée de la fenêtre glissante en fonction de sa valeur de représentativité. Cette valeur de représentativité est définie comme la valeur d'appartenance maximale d'une forme par rapport à toutes les classes. La mise à jour des formes de la fenêtre glissante est réalisée si la taille maximale autorisée de la fenêtre est atteinte ou si une évolution a été détectée.

La méthode répond à la problématique de la classification dynamique tout en apportant une solution non supervisée. Cependant, de nombreux paramètres doivent être définis et leur importance est primordiale dans les résultats de classification de la méthode. De même, dans le cas d'une évolution abrupte, la méthode souffre d'un problème de réinitialisation totale des classes pour considérer les nouveaux prototypes des données.

1.2.4. Méthodes de reconnaissance des formes dynamiques

Ces méthodes traitent le problème de classification des formes dynamiques, c'est à dire des formes décrites en fonction de leur évolution dans le temps. Pour cela, on commence généralement par étudier la structure des formes dynamiques obtenues sur un système. Cette structure peut être obtenue par des méthodes de segmentation qui vont transformer une forme dynamique en une représentation simplifiée appelée tendance. La tendance est composée d'épisodes, c'est à dire d'intervalles de temps, auxquels on associe des formes particulières, appelées primitives. Quand plusieurs primitives similaires se succèdent, ces dernières sont regroupées dans le même épisode. La tendance caractéristique d'un signal est définie comme sa signature. Les primitives d'une signature sont les principales formes caractéristiques identifiées pour une forme dynamique. Elles sont connectées par des points de rupture, correspondant aux instants de début et de fin des épisodes, trouvés par la méthode de

segmentation. Une fois la signature de chaque signal obtenue, il est possible d'utiliser différentes approches pour la classification des données. On peut utiliser entre autres :

- des grammaires qui vont analyser chaque signature comme une chaîne de caractères dont la syntaxe va être similaire pour les formes dynamiques de la même classe,
- des automates qui vont faire passer les primitives par une suite de différents états afin d'identifier le mode de fonctionnement d'un système,
- des mesures de similarité afin de quantifier la ressemblance entre plusieurs signatures,
- des règles floues ou non de classification.

Les méthodes de classification des formes dynamiques sont particulièrement utilisées pour tout ce qui concerne la reconnaissance vocale, les données médicales des patients, etc. Il est à noter qu'elles peuvent être particulièrement efficaces pour détecter des périodes transitoires dans un signal. Par contre, elles doivent bien souvent être appliquées au cas par cas puisqu'elles sont dépendantes du domaine d'application. Nous allons commencer par décrire plusieurs méthodes de segmentation puis nous verrons quelques techniques permettant de mesurer la similarité entre formes dynamiques. Enfin suite à ces présentations, il sera possible de détailler différentes méthodes de classification utilisant les informations recueillies par ces techniques sur les formes dynamiques.

1.2.4.a. Méthodes de segmentation

Parmi les méthodes de segmentation existantes, on peut citer celles de (Höppner, 2000), (Dash, 2001), (Hugueney, 2003), (Keogh, 2001), (Charbonnier, 2004) et (Calvelo, 1999). Ces méthodes recherchent les points de rupture caractéristiques des changements d'un signal en fonction de sa tendance, en utilisant pour la plupart des seuils d'erreur de modélisation. Ces points de rupture peuvent être définis par interpolation, c'est à dire qu'ils sont situés sur le signal, ou par régression. Les points de rupture trouvés par régression sont définis sur ou en dehors du signal.

Dans le cas d'une régression linéaire (Calvelo, 1999), on peut utiliser la méthode des moindres carrés qui consiste à minimiser la somme des carrés des écarts entre les mesures et la droite de régression. Ces méthodes peuvent être définies pour une application hors ligne ou en ligne. Dans le cas d'une méthode en ligne, une fenêtre glissante est généralement utilisée pour sélectionner les données à segmenter. La méthode de (Dijkstra, 1959) présente l'ensemble des points du signal dans un graphe dont l'ensemble des sommets représente les

points de mesure. Chaque arc reliant les sommets, c'est-à-dire chaque lien entre les données a un sens et un coût. La solution optimale de (Dijkstra, 1959) consiste à trouver le chemin le plus court et le moins coûteux permettant ainsi de segmenter le signal. Pour ce faire le graphe est parcouru de multiples fois et le problème est résolu en traitant progressivement les sous-problèmes. (Duda, 1973) propose une méthode hiérarchique descendante qui effectue des divisions successives jusqu'à obtenir une erreur d'approximation inférieure à celle autorisée. Une fois ce seuil atteint la méthode possède un certain nombre d'épisodes représentant le signal complet. Dans (Koski, 1994), l'auteur ajoute les nouvelles données à une fenêtre glissante tant que l'erreur d'approximation est inférieure à une limite définie. Une fois cette limite atteinte, l'épisode est déterminé et une nouvelle fenêtre peut commencer. Dans (Keogh, 2001) les auteurs proposent de combiner l'approche hiérarchique ascendante et la méthode de la fenêtre glissante en une seule méthode appelée Sliding Window And Bottom-up (SWAB). Ils considèrent que la taille initiale de la fenêtre glissante doit être définie de telle façon que cette dernière puisse contenir au moins cinq ou six segments. La méthode est initialisée en utilisant la première fenêtre de données. Dans cette fenêtre la méthode ascendante est appliquée et seul le premier segment est conservé. Puis la fenêtre est décalée de la longueur du premier segment trouvé et elle incorpore de nouvelles données. Ainsi, la méthode fonctionne en deux temps permettant un affinement de la segmentation. La difficulté de la méthode réside dans la définition d'une bonne taille de la fenêtre. Pour cela, les auteurs définissent des limites supérieures et inférieures de taille de fenêtre à ne pas dépasser. Dans (Charbonnier, 2002), les auteurs réalisent la segmentation des signaux en une succession de fonctions linéaires affines. La méthode consiste en la détection des moments où l'approximation linéaire n'est plus acceptable, i.e. quand elle est supérieure à un seuil. Pour cela, les auteurs utilisent la méthode de la somme cumulée (CUSUM) (Brown, 1975) pour détecter les changements d'épisodes. CUSUM compare la valeur des données actuelles avec celle du modèle actuel des écarts d'approximation. Si cette somme cumulée est inférieure à un premier seuil les points sont ajoutés à l'épisode courant, par contre si l'on dépasse ce seuil on stocke les mesures dans un bloc de valeurs anormales. Si la valeur atteint un deuxième seuil une nouvelle fonction linéaire est calculée en utilisant l'estimation des moindres carrés sur les valeurs contenues dans le bloc de valeurs anormales. La valeur du CUSUM est alors remise à zéro. Ainsi on peut déterminer l'intervalle de temps de chaque épisode. De plus, la gestion des données aberrantes est prise en compte par l'approche proposée par (Charbonnier, 2002). En effet, en fonction d'un nombre de données défini dépassant brusquement deux seuils, la méthode supprimera ces données ou les prendra en considération dans la méthode de

segmentation. (Höppner, 2000) propose une méthode Hard c-Connected Lines (HcCL) d'optimisation des fonctions linéaires par morceau d'un signal. Cette méthode part d'un nombre de points de ruptures défini initialement dont la position sur le signal est aléatoire. Puis, ces points de ruptures sont reliés pour former les différents segments qui vont être optimisés par la méthode. Des procédures de fusion, de scission et de suppression des segments sont alors réalisées pour diminuer l'erreur d'approximation.

A partir de la segmentation réalisée par ces méthodes, différents types de primitives peuvent être obtenus pour un signal. Ces dernières sont obtenues soit à partir des dérivées premières d'un signal soit à partir de ses dérivées premières et secondes. Lorsque la valeur qualitative (0 pour une stabilisation, '+' pour une croissance et '-' pour une décroissance) d'une dérivée change, un nouvel épisode débute. Cette valeur change pour une dérivée première lorsqu'un extremum est atteint et pour une dérivée seconde lorsqu'un point d'inflexion est passé. Dans les paragraphes suivants, la dérivée première sera notée dx et la dérivée seconde ddx . Les primitives peuvent considérer généralement six types de structure (Olszewsky, 2001) : constante, linéaire, exponentielle, sinusoïdale, triangulaire et trapézoïdale (ou rectangulaire). Parmi les primitives les plus utilisées pour la segmentation des signaux, on peut citer :

- les primitives du 1^{er} ordre obtenues à partir de dx . En fonction de la valeur qualitative de cette dérivée ('+', 0, '-'), la primitive sera respectivement linéairement croissante, stable, ou décroissante,
- celles de (Cheung, 1990) qui sont au nombre de 7, de formes triangulaires (Figure 1.10.a), et qui reposent sur la valeur qualitative des dérivées premières et secondes d'un signal,
- celles de (Janusz, 1991) qui en plus des 7 primitives de (Cheung, 1990), ajoute deux primitives (Figure 1.10.b) pour la détection du maximum ($dx=0$, $ddx='-'$) et du minimum ($dx=0$, $ddx='+'$),
- celles de (Colomer, 2002) qui en plus des 9 primitives précédemment définies ajoute 4 nouvelles primitives (Figure 1.10.c) concernant les discontinuités (ou cassures) dans le signal lorsque ddx devient très importante.

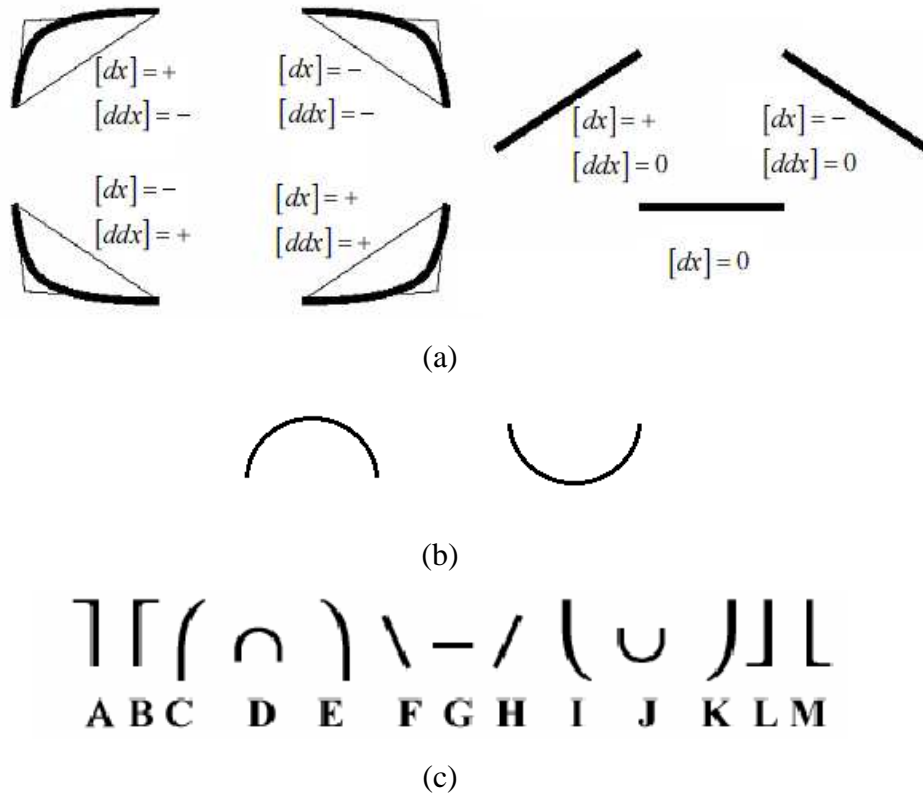


Figure 1.10. Les formes des 7 primitives de Cheung (a), les primitives de Janusz (b), et l'ensemble des 13 primitives de Colomer avec leurs notations (c).

Le choix d'une primitive se fait par minimisation de l'écart d'approximation entre chaque primitive testée et chaque épisode du signal. Quand un signal est bruité ou que sa tendance n'est pas clairement apparente, il est généralement plus efficace d'utiliser une représentation simple à base de la dérivée première seulement. Dans le cas des systèmes médicaux on se limite souvent à une description des primitives du 1^{er} ordre correspondant plus au vocabulaire utilisé par les médecins.

Dans la partie suivante, les signatures obtenues par les méthodes de segmentation sont utilisées pour caractériser la similarité entre formes dynamiques.

1.2.4.b. Mesures de similarité entre les formes dynamiques

Afin de classifier les formes dynamiques, il est nécessaire de leur trouver des similarités. Ces mesures sont généralement basées soit sur les primitives des signatures obtenues pour les formes dynamiques soit sur la distance, calculée point par point sur les signaux, et éventuellement revue globalement à l'aide d'un opérateur d'agrégation (minimum, maximum, moyenne, etc.). (Das, 1997) considère que deux formes dynamiques peuvent être similaires,

ou au moins en partie, en fonction des sous-séquences identiques de ces formes. (Yazdani, 1996) part du même principe mais il considère que seule la plus longue sous séquence identique chez plusieurs formes permet de quantifier leur similarité. Si ces sous-séquences sont décalées et que leur échelle n'est pas la même, elles pourront quand même être considérées comme identiques. On fait alors appel à des méthodes dites de Dynamic Time Warping (DTW) (Koehn, 2001). Dans (Joentgen, 1999), les auteurs mesurent la similarité des formes dynamiques à partir des primitives détectées pour chaque signal. Le même nombre de primitives doit être considéré pour chaque signal. Les auteurs commencent par quantifier la similarité entre chaque couple de primitives possible. Puis, les signaux sont analysés et l'ensemble des mesures de similarité est obtenu. Enfin, l'agrégation de ces valeurs est réalisée pour obtenir une valeur globale de similarité entre les formes dynamiques. Dans (Dash, 2003) la similarité entre deux primitives est basée sur le signe de leur première dérivée. Si leur signe est opposé ('-'/'+') alors leur similarité est nulle sinon une valeur de similarité leur correspond. Les primitives sélectionnées pour cette mesure sont celles de (Cheung, 1990). Il est ensuite nécessaire de faire cette mesure de similarité pour l'ensemble d'un signal ou pour certaines parties spécifiquement définies comme informatives.

Lorsque des caractéristiques structurelles ne peuvent pas être obtenues directement sur un signal on utilise les techniques du traitement de signal pour extraire des singularités et ainsi permettre la classification des formes dynamiques. (Struzik, 1995) propose une méthode basée sur la décomposition des signaux à l'aide des ondelettes afin d'observer des similarités entre les signaux temporels. Une fois la décomposition effectuée, l'auteur récupère des mesures statistiques globale et locale pour quantifier la similarité. (Angstenberger, 2000) considère que deux formes dynamiques peuvent être équivalentes simplement si leur pente ou si leur courbure est équivalente durant une période suffisamment importante. Le nombre d'oscillations, c'est-à-dire de changements de signe de la dérivée, leurs valeurs, ou leurs positions peuvent également être suffisants pour mesurer la similarité entre formes dynamiques (Pinquier, 2004). Si une mesure statistique est réalisée en ligne sur les signaux, il est important que cette mesure soit capable de les discriminer à n'importe quel moment, avec bien sûr un retard de détection possible. A partir de ces informations, obtenues de façon structurelle ou statistique, de nombreuses méthodes de reconnaissance des formes permettent en ligne ou hors ligne, de classifier les formes dynamiques.

1.2.4.c. Classification des formes dynamiques

L'utilisation des caractéristiques recueillies sur les formes dynamiques peut se faire par différents types de méthodes.

Par un ensemble d'automates :

Dans la littérature, (Koski, 1995) s'intéresse à la classification de signaux d'Electrocardiogrammes (ECG) à partir des primitives et à l'aide d'un ensemble d'automates finis. Les auteurs commencent par débruiter les signaux en supprimant les hautes fréquences. Puis, la segmentation des signaux ECG est réalisée pour obtenir une suite de segments. Pour cela ils définissent une taille de fenêtre initiale et ils ajoutent des données à cette fenêtre tant que le segment approximant l'épisode de la fenêtre est inférieur à une erreur d'approximation fixée. A partir des épisodes trouvés, une primitive de premier ordre (une droite) est déterminée pour chaque épisode, en fonction de la valeur de sa pente. Les auteurs ont choisi d'utiliser cinq primitives en fonction des valeurs de pente. Pour réaliser la classification, les auteurs utilisent un ensemble d'automates finis. Il s'agit en fait d'un automate fini faisant appel à divers automates en fonction de la suite de primitives trouvée en ligne sur un signal. Un automate est défini par un état initial et un ensemble de liaisons entre différents états définis comme possibles. Dix automates sont ainsi définis pour identifier chaque type de forme caractéristique d'un ECG. Par exemple, les auteurs utilisent un automate pour trouver les formes de type QRS (forme complexe) et un autre pour trouver la forme T. Chacune de ces formes sera identifiée à l'aide d'un chemin composé de la succession de plusieurs primitives. Ces formes peuvent être obtenues de bien des façons différentes et c'est pourquoi les auteurs ont divisé le problème en plusieurs automates. Les auteurs ont ainsi défini une suite de chemins et d'états possibles, pour obtenir la classification des ECG. Si l'ECG passe par certains chemins, un problème cardiaque est détecté. La méthode utilise seulement quelques primitives qui semblent suffire mais elle est complètement dépendante du domaine médical des ECG. Elle donne de bons résultats concernant certains types de formes (QRS) mais elle est moins précise concernant les formes T et surtout P.

Par une grammaire :

(Rengaswamy, 1995) se base sur le formalisme décrit par (Janusz, 1991) mais le développe pour tenir compte du bruit des données. Il utilise un ensemble de formes fondamentales, les primitives, et un ensemble de règles grammaticales spécifiant les différentes liaisons possibles entre les primitives, appelé un langage. (Rengaswamy, 1995) commence par identifier les primitives des signatures à l'aide d'un réseau de neurone. Dans son réseau de neurone, chaque

entrée correspond à la valeur d'un signal à un instant donné de la fenêtre d'observation. Chaque sortie correspond à la classe, i.e. la primitive dans ce cas précis, à laquelle on affecte la valeur reçue en entrée. Seulement, afin d'identifier cette primitive, il est nécessaire d'utiliser un nombre minimum de formes en entrée. Ce nombre est un paramètre difficile à déterminer. Puis la classification est réalisée par une grammaire qui spécifie quelle suite de primitives est autorisée dans le fonctionnement du système. Plus l'enchaînement comporte de primitives non désirées, plus le système s'éloigne d'un comportement normal. Cette méthode a l'avantage de ne pas nécessiter de filtrage des données, ni de définition de seuils mais par contre il faut définir la taille de la fenêtre, les fonctions d'activation, le nombre de couches, etc.

Par Dynamic Time Warping (DTW) :

La méthode de déformation temporelle dynamique ou Dynamic Time Warping (DTW) peut également être utilisée pour discriminer les données et ainsi permettre leur classification. Dans (Colomer, 2002) les auteurs combinent la méthode DTW et la représentation à base de primitives pour réaliser la classification. La méthode DTW (Sakoe, 1978) est utilisée pour aligner deux séries temporelles qui peuvent avoir subi des distorsions, ce qui peut être le cas pour un problème de reconnaissance vocale par exemple. Pour cela la distance entre chaque point de deux séquences est calculée, puis la méthode évalue les distances cumulées des séquences pour minimiser la distorsion. (Colomer, 2002) propose une méthode n'appliquant pas DTW sur les signaux originaux mais sur leurs épisodes. Ainsi le temps de calcul est énormément réduit. Seulement, il est nécessaire de définir des valeurs de similarité entre l'ensemble des primitives qui peuvent être utilisées. Pour cela, l'auteur propose une matrice de distances basée sur les caractéristiques des primitives.

Par un arbre de décision :

Dans (Bakshi, 1994) les auteurs commencent par réaliser une décomposition en ondelettes des signaux qui permet de supprimer le bruit des données. Puis, la décomposition en primitives de (Cheung, 1990) est réalisée sur les formes dynamiques et enfin un arbre de décision est utilisé pour réaliser la classification des données en fonction de la mesure d'entropie de Shannon.

Par un système expert :

(Vedam, 1997) propose une méthode de suivi et de diagnostic par analyse de tendance adaptative en utilisant les ondelettes. Les signaux reçus sont en fait projetés sur différents niveaux définis par les ondelettes. Puis, les auteurs utilisent le réseau de neurone de (Rengaswamy, 1995) avec des paramètres différents, pour identifier les primitives des

signaux. L'auteur considère que le problème de la taille de la fenêtre d'observation des données utilisée par la méthode de (Rengaswamy, 1995) est supprimé puisque les ondelettes vont tenir compte du changement de fréquence des données. La taille de la fenêtre devient ainsi adaptative, ce qui permet de traiter les cas où l'évolution d'un système est très lente. Les primitives identifiées sont ensuite utilisées comme des entrées par un système développé à base des connaissances d'un expert, pour obtenir la classification des signaux.

Par des méthodes tenant compte de la tendance moyenne des données :

Plusieurs méthodes sont basées sur l'évolution de la tendance moyenne des données. Parmi celles-ci on peut citer Cumulative SUM control chart (CUSUM) (Charbonnier, 2004), Dynamic CUSUM (DCS) (Khalil, 1999), Shewart (Shewart, 1939), etc. L'approche de DCS (Khalil, 1999) utilise deux fenêtres glissantes de taille fixe. Chaque fenêtre permet de calculer la somme cumulée qui précède un point et celle qui la suit. DCS est basée sur la somme cumulée locale, c'est à dire proche d'un point, des rapports de vraisemblance entre deux segments locaux estimés à chaque instant. Un retard de détection au minimum égal à la taille d'une fenêtre est alors obligatoire. L'approche développée par (Nezhad, 2010) consiste en une nouvelle mesure récursive qui permet de suivre l'évolution du système en se basant sur les données connues *a priori* tout en tenant compte des nouvelles données. De la même façon que pour les autres méthodes de ce type (Shewart, etc.) la méthode utilise des limites supérieure et inférieure pour déterminer la classe des données. La méthode de (Shewart, 1939) consiste à suivre l'évolution d'un très petit groupe de données (généralement moins de cinq) ou leur valeur moyenne. Il est alors considéré que les valeurs statistiques d'un système sont normalement distribuées. Par conséquent, dès que la mesure de Shewart évolue par rapport à une certaine limite, le mode du système est considéré comme changé et il est nécessaire de lancer une alarme. L'un des principaux problèmes de cette méthode est qu'elle réagit seulement de façon brutale quand une donnée passe une limite de contrôle.

A partir des paramètres des modèles AutoRegressive Moving Average (ARMA) ou Exponentially Weighted Moving Average (EWMA) :

On peut également utiliser différents filtres tels que ceux du modèle Exponentially Weighted Moving Average (EWMA) (Roberts, 1959) ou AutoRegressive Moving Average (ARMA) (Box, 1970). Dans ce cas le signal est représenté par un modèle paramétrique. On extrait des indicateurs basés sur les paramètres de ce modèle. Ces indicateurs sont généralement les coefficients du modèle. Pour le cas de EWMA toutes les données connues *a priori* sont utilisées, y compris les plus récentes. Il est alors nécessaire de définir un facteur de poids qui permet d'être sensible aux dérives du système de façon plus ou moins importante.

Ce poids est une constante qui détermine le taux d'importance que prennent les anciennes données dans le calcul du EWMA. Afin de conserver une certaine importance pour les données, le poids est généralement défini entre 0,2 et 0,3 (Hunter, 1986). Pour le calcul du EWMA, une fois ce poids défini et des valeurs historiques connues, il suffit de calculer les limites supérieures et inférieures du modèle.

Dans la section suivante, de nombreuses applications des méthodes de reconnaissance dynamique des formes sont présentées.

1.2.5. Applications des méthodes de classification dynamique

Les méthodes de classification dynamique (Byington, 2002) (Ciarapica, 2006) (Dragomir, 2008) (Muller, 2008) (Rafiq, 2004) (Wang, 2003a) (Zemouri, 2003) sont utilisées pour résoudre de nombreux problèmes réels tels que : l'exploration des données, la bio-informatique, la classification de documents, l'analyse d'images, la reconnaissance biométrique, la reconnaissance vocale, le diagnostic industriel, etc. Dans (Michie, 1994), un ensemble de règles générées à partir de l'ensemble d'apprentissage est utilisé pour réaliser le diagnostic des fautes des transmissions automatiques. Six mois après, ces règles ne sont plus valides suite à des changements survenant dans les paramètres de l'appareil. L'application de (Angstenberger, 2000) concerne l'évaluation des risques clients pour une banque, en fonction des changements dans leur mode de consommation. (Guedalia, 1999) traite le problème de classification de la qualité des fruits en fonction des dégâts résultant d'un mauvais climat ou d'autres événements extérieurs. Dans (Amadou Boubacar, 2006), les auteurs souhaitent détecter et suivre l'évolution progressive des modes de fonctionnement d'un régulateur thermique en fonction de l'âge de ses composants et d'autres facteurs temporels de son environnement. Dans (Last, 2002), l'auteur cherche à estimer le ratio entre le nombre de cartes microélectroniques défectueuses et le nombre de cartes de bonne qualité faisant partie d'un même lot. Cette estimation est basée sur les valeurs journalières évolutives de plusieurs facteurs matériels, des équipements et également des facteurs humains. (Cohen, 2004) traite les flux dynamiques de données dans le but de réduire le temps d'attente des conducteurs au niveau des intersections des routes. (Mustapha, 2008) réalise la détection de perturbations sur le processus chimique Tennessee Eastman Challenge Process (TECP) en utilisant la méthode du CUSUM dynamique. Dans (Rengaswamy, 1995), l'auteur applique son approche sur le suivi et le diagnostic d'un système qui doit rendre les flux d'hydrocarbure plus fluide. En utilisant sa méthode, l'auteur permet de trouver les caractéristiques principales de la tendance du système et les conséquences potentielles, tout en fournissant suffisamment d'informations

aux opérateurs. (Stephanopoulos, 1997) utilise la méthode définie par (Bakshi, 1994) pour réaliser le diagnostic d'un procédé industriel concernant la fermentation.

En regardant les domaines d'application de ces méthodes, on peut s'apercevoir que les méthodes de reconnaissance dynamique des formes peuvent être appliquées pour tout type de données.

1.3. Conclusion

Dans ce chapitre, quatre catégories de reconnaissance des formes de la littérature ont été présentées. Il s'agissait des méthodes de reconnaissance statique des formes statiques, des méthodes de reconnaissance dynamique des formes statiques, des méthodes de reconnaissance statique des formes dynamiques et des méthodes de reconnaissance dynamique des formes dynamiques. Le choix d'une catégorie de méthodes de reconnaissance des formes se fait en fonction de différents éléments tels que les données (formes, classes, nombre de paramètres observés, etc.) et l'environnement du système (statique, dynamique).

Nous avons détaillé le principe de plusieurs méthodes de reconnaissance dynamique des formes ainsi que plusieurs méthodes de reconnaissance des formes dynamiques. Ces méthodes font partie des axes de recherche et de développement présentés dans ce manuscrit. En effet, nous nous sommes intéressés au diagnostic et au suivi des systèmes évolutifs, i.e. les systèmes dont les caractéristiques évoluent dans le temps, pour lesquels il est nécessaire d'utiliser des méthodes de classification dynamique. Ces méthodes sont capables de suivre les changements de caractéristiques des systèmes évolutifs et d'adapter le classifieur et les classes pour conserver les performances du classifieur après l'évolution du système. Plusieurs exemples d'applications de ces méthodes ont été présentés.

Dans le prochain chapitre, trois méthodes de reconnaissance dynamique des formes statiques sont proposées. Il s'agit d'une part de la méthode Fuzzy Pattern Matching Dynamique (FPMD) basée sur une représentation par histogrammes des distributions des données et intégrant des mécanismes de suivi d'évolution et d'adaptation des classes. D'autre part, deux approches Supervisée et Semi-Supervisée de K-Plus Proches Voisins Flous Dynamique (respectivement KPPVFD-S et KPPVFD-SS) sont également proposées pour suivre les évolutions des classes des systèmes évolutifs.

Chapitre 2. Contributions à la reconnaissance dynamique des formes

Ce chapitre présente trois méthodes de classification dynamique que nous proposons pour le suivi et la discrimination des classes dynamiques dont les caractéristiques changent au cours du temps. La première méthode proposée appelée Fuzzy Pattern Matching Dynamique (FPMD) est supervisée. Elle repose sur la méthode FPM Incrémentale (FPMI) en intégrant différents indicateurs qui permettent de suivre l'évolution des données et d'adapter les classes lorsqu'une évolution a été confirmée par la méthode. La deuxième et la troisième méthodes reposent sur la méthode des K Plus Proches Voisins Flous (KPPVF). La deuxième méthode proposée est appelée KPPVF Dynamique Supervisée (KPPVFD-S). Elle utilise deux indicateurs d'évolution des données pour suivre les changements de caractéristiques d'un système. Elle intègre une phase de validation des évolutions ainsi qu'une phase de mise à jour du classifieur et des modèles des classes, qui n'utilise que les formes représentatives. La troisième méthode appelée KPPVF Dynamique Semi-Supervisée (KPPVFD-SS) intègre une phase de détection des évolutions, une phase de validation des évolutions ainsi qu'une phase de mise à jour du classifieur et des modèles des classes. KPPVFD-SS permet ainsi de suivre les évolutions des données tout en détectant les nouvelles classes d'un système. Ensuite, les principes de ces méthodes sont illustrés et leurs performances sont comparées à d'autres méthodes de la littérature en utilisant plusieurs exemples.

2.1. Fuzzy Pattern Matching Dynamique (FPMD)

La méthode Fuzzy Pattern Matching (FPM) a été initialement développée par Cayrol (Cayrol, 1982). Dans (Sayed Mouchaweh, 2002) une solution pour rendre cette méthode incrémentale (FPMI) a été développée. FPMI donne de bons résultats de classification avec des classes statiques tout en tenant compte des changements locaux des classes (élargissement des classes, changement du contour d'une classe). Cependant FPMI ne contient pas de stratégie lui permettant d'oublier les formes obsolètes et d'utiliser les formes utiles et représentatives pour la mise à jour des paramètres du classifieur. En effet, FPMI considère toutes les formes comme représentatives et utiles pour la mise à jour des paramètres du classifieur. FPMI ne peut donc pas s'adapter en cas de déplacement, scission, rotation, disparition, etc. d'une classe. Cette méthode est une approche simple de reconnaissance des formes avec un temps de classification faible et constant par rapport à la taille de la base de

données. C'est pourquoi nous avons choisi de poursuivre le développement de FPMI afin de tenir compte de l'évolution des classes des systèmes évolutifs, c'est-à-dire du changement de caractéristiques survenant dans les classes représentant les modes de fonctionnement surveillés d'un système. Ce développement consiste en une stratégie de détection des évolutions de classe et d'oubli des formes obsolètes permettant d'utiliser uniquement des formes utiles pour la mise à jour des paramètres du classifieur.

Afin de comprendre le fonctionnement de notre approche dynamique (FPMD) il est important de présenter la méthode Fuzzy Pattern Matching Incrementale (FPMI).

2.1.1. Principe de la méthode Fuzzy Pattern Matching Incrementale (FPMI)

Le fonctionnement de FPMI comporte une phase d'apprentissage, une phase de classification et une phase de mise à jour incrémentale.

2.1.1.a. Phase d'apprentissage

Soit X_i l'ensemble des N_i points d'apprentissage appartenant à la classe C_i dans un espace de représentation, \mathfrak{R}^A , de d attributs. X est l'ensemble d'apprentissage, contenant les N points correspondant aux c classes, noté :

$$X = \bigcup_{i=1}^c X_i \quad (2.1)$$

La phase d'apprentissage consiste à construire, à partir de X , une règle de décision caractérisée par un ensemble de fonctions d'appartenance. Ces dernières sont basées sur des histogrammes permettant d'estimer la probabilité conditionnelle selon chaque attribut j de chaque classe C_i . Le nombre h de barres $b_k^j, k \in \{1, 2, \dots, h\}$ d'un histogramme est déterminé expérimentalement. Le choix de h conditionne les performances de FPMI de manière critique. Les bornes inférieure \min_i^j et supérieure \max_i^j de chaque histogramme sont généralement définies par les valeurs minimale et maximale des coordonnées des points d'apprentissage de la classe C_i suivant l'attribut j . La largeur Δ_i^j d'une barre de l'histogramme par rapport à l'attribut j pour une classe i est définie par:

$$\Delta_i^j = \frac{(\max_i^j - \min_i^j)}{h} \quad (2.2)$$

L'histogramme ou la distribution de probabilité $\{p_i^j(b_{ik}^j), i \in \{1, 2, \dots, c\}, j \in \{1, 2, \dots, d\}, k \in \{1, 2, \dots, h\}\}$ d'une classe C_i suivant l'attribut j est déterminé par le calcul de la probabilité $p_i^j(b_{ik}^j)$ de chaque barre b_{ik}^j :

$$p_i^j(b_{ik}^j) = \frac{n_{ik}^j}{N_i} \quad (2.3)$$

où n_{ik}^j est le nombre de points d'apprentissage de la classe C_i qui sont dans la barre b_{ik}^j . La distribution de probabilité obtenue est alors transformée en distribution de possibilité $\{\pi_i^j(b_{ik}^j), i \in \{1, 2, \dots, c\}, j \in \{1, 2, \dots, d\}, k \in \{1, 2, \dots, h\}\}$ en utilisant la transformation de Dubois et Prade (Dubois, 1993) :

$$\pi_i^j(b_{ik}^j) = \sum_{z=1}^h \min(p_i^j(b_{iz}^j), p_i^j(b_{ik}^j)) \quad (2.4)$$

La mesure de possibilité a l'avantage de prendre en considération l'imprécision et l'incertitude contenues dans les données. En effet, l'imprécision des données est retranscrite par le biais des valeurs d'appartenance floue d'un point à chaque classe et l'incertitude est mesurée à l'aide d'un degré de possibilité. Finalement, la densité de possibilité π_i^j de la classe C_i suivant l'attribut j est obtenue par interpolation linéaire des centres des barres des histogrammes de possibilité. Pour que cela soit réalisable, il est nécessaire de lier les centres de la première et de la dernière barre de l'histogramme à des barres fictives de valeur de possibilité nulle. C'est pourquoi deux barres supplémentaires de valeur nulle sont ajoutées au début et à la fin de chaque histogramme.

2.1.1.b. Phase de classification

Une forme x est assignée à une classe C_i en trois étapes :

- détermination de la valeur d'appartenance en possibilité π_i^j de x_j à chaque classe C_i suivant chaque attribut j . Cette valeur de possibilité est obtenue par projection de x_j sur la densité π_i^j connue pour chaque classe C_i suivant chaque attribut j (Figure 2.1),

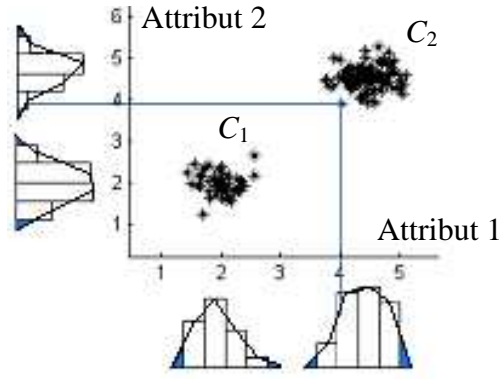


Figure 2.1. Projection du point à classer sur les densités de possibilité des classes.

- fusion, pour chaque classe C_i , des valeurs d'appartenance en possibilité $\pi_i^1, \pi_i^2, \dots, \pi_i^d$ par un opérateur d'agrégation. L'opérateur est généralement l'opérateur "minimum" puisqu'il est simple à calculer et qu'il donne des bons résultats (Devillez, 1998). Le résultat de la fusion représente la possibilité d'appartenance π_i de x à chaque classe C_i ,
- classification de x dans la classe pour laquelle il a la valeur d'appartenance maximale.

2.1.1.c. Phase de mise à jour incrémentale

Chaque nouveau point classé apporte des nouvelles informations sur les caractéristiques d'une classe. Pour prendre en compte ces informations, les fonctions d'appartenance doivent être réappprises après la classification de chaque nouvelle forme. Un algorithme d'apprentissage incrémental est proposé dans (Sayed Mouchaweh, 2002) pour faire cette mise à jour incrémentalement en se basant sur les formes récemment classées et sur les fonctions d'appartenance déjà connues. Quand une nouvelle forme est classée dans C_i , le nombre de points appartenant à cette classe devient N_i+1 , la valeur de probabilité de chaque barre change. Si la forme classée x est située dans une barre b_{ik}^j , la nouvelle probabilité $p_i'^j(b_{ik}^j)$ de cette barre est (Sayed Mouchaweh, 2002) :

$$p_i'^j(b_{ik}^j) = \frac{n_{ik}^j}{N_i} \times \frac{N_i}{N_i+1} + \frac{1}{N_i+1} = p_i^j(b_{ik}^j) \times \frac{N_i}{N_i+1} + \frac{1}{N_i+1} \quad (2.5)$$

Pour les autres barres, la nouvelle probabilité est :

$$p_i'^j(b_{iz}^j) = \frac{n_{iz}^j}{N_i} \times \frac{N_i}{N_i+1} = p_i^j(b_{iz}^j) \times \frac{N_i}{N_i+1}, \quad z=1 \dots h \text{ \& } z \neq k \quad (2.6)$$

2.1.2. Principe de la méthode Fuzzy Pattern Matching Dynamique (FPMD)

La méthode Fuzzy Pattern Matching Dynamique (FPMD) (Hartert, 2009a) (Hartert, 2009b) (Hartert, 2010b) est une version améliorée de la méthode FPMI. FPMD intègre un mécanisme d'ajustement des paramètres du classifieur quand des changements importants des caractéristiques des classes sont détectés pendant une fenêtre glissante. Sa taille dépend de la dynamique de l'application. FPMD comprend une phase de détection des évolutions, une phase d'adaptation des classes évoluées, et une phase de validation des classes (Figure 2.2).

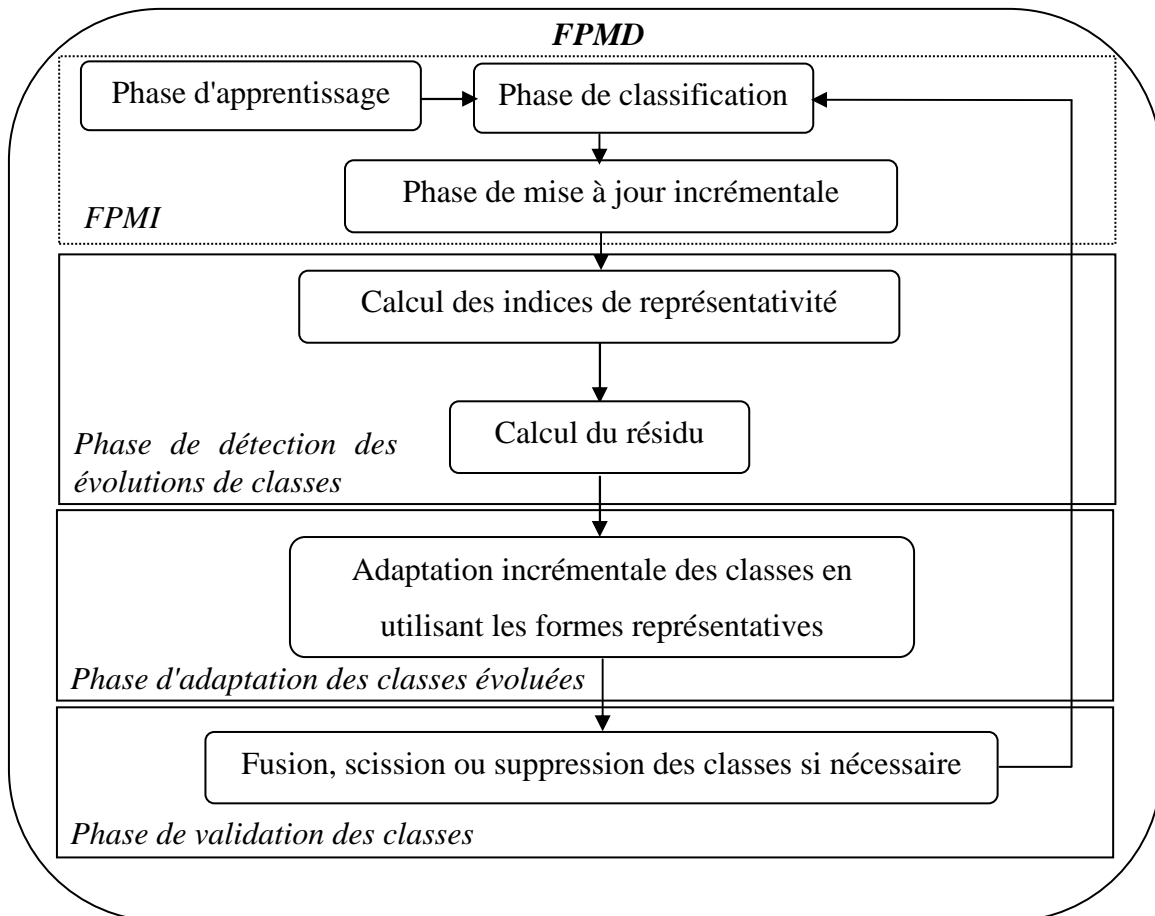


Figure 2.2. Algorithme global de FPMD.

Dans la phase d'apprentissage, FPMD utilise l'ensemble des points étiquetés pour créer les histogrammes de possibilité qui permettent d'estimer les fonctions d'appartenance. Dans les parties suivantes, c est le nombre de classes connues et h_i^j représente le nombre de barres de l'histogramme de la classe i et de l'attribut j . La probabilité d'une barre b_{ik}^j , $k \in \{1, \dots, h\}$, est notée $p_i^j(b_{ik}^j)$. Un nouveau point est classé selon ses valeurs d'appartenance à chaque classe par rapport à chaque attribut.

2.1.2.a. Détection des évolutions

Durant cette phase, deux indicateurs sont utilisés pour surveiller les changements temporels d'un système.

Indicateur de représentativité des formes :

Le premier indicateur évalue les formes non représentatives. Cet indicateur est basé sur les changements temporels accumulés dans les distributions de probabilité pendant une fenêtre glissante de taille variable. Il considère le nombre de nouvelles formes situées dans chaque barre des histogrammes de probabilité et les probabilités initiales avant l'assignement des nouvelles formes. Soit $\Delta T = t_w - t_s$ la durée variable de la fenêtre glissante à l'instant actuel t_w , commencée à l'instant t_s . Soit $\{p_i^j(b_{ik}^j), k \in \{1, 2, \dots, h\}\}$ l'histogramme de probabilité pour la classe C_i en fonction de l'attribut j . Soit ΔN_i le nombre de nouveaux points classés dans la classe C_i et soit Δn_{ik}^j le nombre de nouveaux points situés dans la barre b_{ik}^j . La nouvelle probabilité de cette barre est calculée par :

$$p_i^{\prime j}(b_{ik}^j) = \frac{n_{ik}^j + \Delta n_{ik}^j}{N_i + \Delta N_i} \quad (2.7)$$

où n_{ik}^j est le nombre de formes de la classe C_i situé dans la barre b_{ik}^j avant la classification des nouvelles formes. Le changement temporel accumulé pour chaque barre est évalué comme la différence entre les probabilités des barres à la fin $p_i^{\prime j}(b_{ik}^j)$ et au commencement $p_i^j(b_{ik}^j)$ de la fenêtre glissante :

$$\Delta p_i^j(b_{ik}^j) = p_i^{\prime j}(b_{ik}^j) - p_i^j(b_{ik}^j) = \frac{N_i \cdot \Delta n_{ik}^j - n_{ik}^j \cdot \Delta N_i}{N_i \cdot (N_i + \Delta N_i)} \quad (2.8)$$

Un indicateur graduel $I_i^j(b_{ik}^j) : \{1, 2, \dots, h\} \rightarrow [-1, 1]$ est défini pour chaque barre b_{ik}^j de chaque classe C_i en fonction de chaque attribut j . Cet indicateur est basé sur les changements temporels accumulés, définis par (2.18), dans les distributions de probabilité :

$$I_i^j(b_{ik}^j) = \begin{cases} \frac{p_i^{\prime j}(b_{ik}^j) - p_i^j(b_{ik}^j)}{p_i^j(b_{ik}^j)} \in [-1, 0] & \text{si } \Delta p_i^j(b_{ik}^j) < 0 \\ \frac{p_i^{\prime j}(b_{ik}^j) - p_i^j(b_{ik}^j)}{p_i^{\prime j}(b_{ik}^j)} \in [0, 1] & \text{si } \Delta p_i^j(b_{ik}^j) > 0 \end{cases} \quad (2.9)$$

Cet indicateur est calculé après la classification de chaque nouveau point. Quand sa valeur est négative pour une barre, elle indique une baisse de la représentativité à la fin de la fenêtre temporelle. Cette baisse est due au fait qu'un faible nombre de points est classé dans cette

barre. Au contraire, une valeur positive indique une augmentation de la représentativité d'une barre due à l'assignation de plusieurs nouveaux points dans celle-ci, en fonction de sa valeur initiale. Il est intéressant de prendre en considération les probabilités anciennes ou initiales ; en effet, s'il n'y a pas de changement important dans la distribution de probabilité, il est normal que la barre ayant la plus grande probabilité reçoive plus de nouveaux points que les autres. Si la plupart de ces nouveaux points sont situés dans les barres qui possèdent une faible probabilité initiale, cela indique un changement sérieux dans la distribution de probabilité et par conséquent leurs indicateurs de représentativité doivent avoir une valeur positive importante.

Indicateur d'évolution ou résidu :

Pour détecter une évolution, un second indicateur, qui est un résidu, est calculé en se basant sur la mesure de représentativité de toutes les barres des histogrammes en fonction de tous les attributs. Le but de ce résidu est de fournir un indicateur qui évalue le changement temporel accumulé dans les distributions de probabilité de chaque classe.

Deux seuils $th1$ et $th2$ définis par un expert en fonction de la dynamique du système sont utilisés pour déterminer dynamiquement la taille ΔT de la fenêtre temporelle. Généralement la valeur de $th1$ est fixée aux alentours de 0,1 dans le but de détecter les évolutions des classes sans détecter tous les petits changements de caractéristiques des classes. Au contraire, $th2$ est généralement fixé à une valeur supérieure ou égale à 0,4, ce qui permet d'obtenir une évolution suffisante pour l'adaptation de la classe. Une trop grande ou trop petite valeur $th1$ peut mener à perdre des formes représentatives ou à garder des formes obsolètes, tandis qu'une valeur trop grande ou trop petite de $th2$ mène à détecter trop tard ou trop tôt l'évolution de la classe. Dans le cas des systèmes réels, les valeurs des seuils peuvent être fixées par défaut à ces valeurs (0,1 pour $th1$ et 0,5 pour $th2$) et par la suite elles peuvent être changées par un expert afin de coller au mieux à la dynamique du système étudié. Les classes peuvent avoir différentes dynamiques, c'est pourquoi $th1$ et $th2$ doivent être définis indépendamment pour chaque classe.

La valeur de cet indicateur est mise à jour et comparée à $th1$ et $th2$ après chaque classification de forme. Si le résidu est inférieur à $th1$, alors aucun changement dans les caractéristiques de la classe n'a commencé. Tant que le résidu est inférieur à $th1$, la mesure de représentativité est basée sur la différence entre la probabilité actuelle de la barre et sa probabilité avant la classification de la forme précédente. Si la valeur de résidu est supérieure à $th1$ et inférieure à $th2$, un changement sérieux a commencé à apparaître. La fenêtre

temporelle commence à ce moment, $t=t_s$, et les formes sont stockées dans un bloc, appelé le bloc d'évolution. Durant cette fenêtre, la mesure de représentativité est calculée utilisant (2.9). Les formes sont stockées dans le bloc d'évolution jusqu'à ce que la valeur du résidu atteigne $th2$. A ce moment, la fenêtre glissante temporelle atteint sa taille finale $t_w=t_f$. Le résidu est défini pour chaque classe C_i comme suit :

$$0 \leq R_i = \max \left(\frac{\sum_{k=1}^h |I_i^j(b_{ik}^j)|}{h} \right) \leq 1, j \in \{1, \dots, d\} \quad (2.10)$$

Quand le résidu atteint $th2$, un changement sérieux peut être décidé pour la classe de telle façon que les paramètres du classifieur (densités de possibilité) doivent être adaptés en utilisant seulement les formes stockées dans le bloc d'évolution. Les formes anciennes qui sont devenues obsolètes sont oubliées et une mise à jour en ligne des paramètres du classifieur peut être réalisée.

2.1.2.b. Adaptation des classes évoluées

Dans FPMD, les bornes des histogrammes sont définies initialement pour chaque classe comme pour FPMI. Puis, elles sont mises à jour en ligne avec la classification des nouveaux points et selon la valeur du résidu. Contrairement à FPMI ce ne sont pas seulement les valeurs des bornes des histogrammes qui sont mises à jour, il peut y avoir création de barres pour conserver un pas d'histogramme qui soit représentatif. Si un nouveau point classé implique la création d'une barre pour un histogramme, une barre est ajoutée à l'histogramme en fonction de l'attribut désigné de telle façon que h_i^j est incrémenté et les bornes des histogrammes sont mises à jour :

$$\left[\begin{array}{l} si \ x_i^j < \min_i^j - \frac{\Delta_i^j}{2} \Rightarrow \min_i'^j = \min_i^j - \Delta_i^j \\ si \ x_i^j > \max_i^j + \frac{\Delta_i^j}{2} \Rightarrow \max_i'^j = \max_i^j + \Delta_i^j \end{array} \right. \quad (2.11)$$

$$h_i'^j = h_i^j + 1 \quad (2.12)$$

La mise à jour des histogrammes de probabilité est réalisée en utilisant une approche incrémentale. Quand la valeur de résidu atteint $th2$, les anciennes formes sont supprimées et seules les nouvelles formes de la fenêtre temporelle sont considérées. Ainsi, la valeur de probabilité de chaque barre est calculée incrémentalement par :

$$p_i^{*j}(b_{ik}^j) = \frac{\Delta n_{ik}^j}{\Delta N_i} = \left(\frac{n_{ik}^j + \Delta n_{ik}^j}{N_i + \Delta N_i} - \frac{n_{ik}^j}{N_i} \times \frac{N_i}{N_i + \Delta N_i} \right) \times \frac{N_i + \Delta N_i}{\Delta N_i}$$

$$p_i^{*j}(b_{ik}^j) = \left(p_i^j(b_{ik}^j) - (p_i^j(b_{ik}^j) \times \frac{N_i}{N_i + \Delta N_i}) \right) \times \frac{N_i + \Delta N_i}{\Delta N_i}, k \in \{1, \dots, h\} \quad (2.13)$$

Cette adaptation incrémentale permet de suivre l'évolution des classes en ligne avec un temps de classification faible et constant.

2.1.2.c. Validation des classes

Une fois l'ensemble des adaptations de classes réalisées, FPMD vérifie la validité de chaque classe en utilisant trois mesures. La première est une mesure de similarité permettant de fusionner les classes dont les caractéristiques sont devenues proches. La deuxième mesure permet de détecter si la scission d'une classe est nécessaire. La troisième mesure utilisée par FPMD est une mesure permettant de supprimer les classes n'étant plus représentatives des modes de fonctionnement actuels.

Mesure de similarité pour la fusion des classes :

Puisque les classes dynamiques sont amenées à se déplacer, il est possible qu'à un moment donné les caractéristiques de plusieurs classes soient suffisamment proches pour permettre leur fusion. Pour décider de cette fusion on utilise une mesure de similarité basée sur le chevauchement ou la proximité entre plusieurs classes. Nous avons utilisé la mesure de similarité proposée par (Frigui, 1996) puisqu'elle évalue le rapprochement entre deux classes, en se basant sur les valeurs d'appartenances de leurs points :

$$\delta_{iz} = 1 - \frac{\sum_{x \in C_i \text{ et } x \in C_z} |\pi_i(x) - \pi_z(x)|}{\sum_{x \in C_i} \pi_i(x) + \sum_{x \in C_z} \pi_z(x)} \quad (2.14)$$

où $\pi_i(x)$ et $\pi_z(x)$ sont respectivement les valeurs d'appartenance de x à C_i et C_z . Plus δ_{iz} est proche de 1, plus les deux classes sont similaires. La valeur maximale représente deux classes complètement chevauchées donc il n'est pas nécessaire d'attendre que la valeur de similarité obtenue soit égale à 1 pour fusionner deux classes. La figure 2.3 montre les valeurs de similarité obtenues en fonction de la proximité de deux classes gaussiennes.

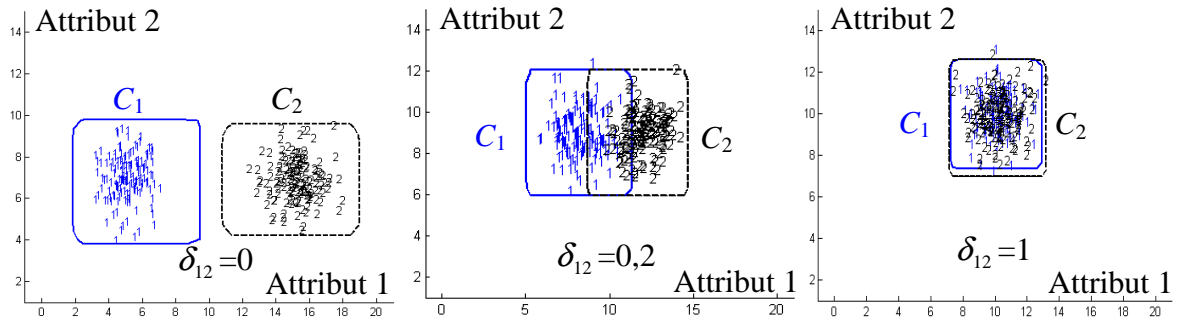


Figure 2.3. Evolution de la mesure de similarité entre deux classes fusionnant.

La fusion de deux classes est réalisée automatiquement si la mesure de similarité entre deux classes C_i et C_j est supérieure à un seuil $th3$ défini par un expert.

Mesure de validité des classes suite à une scission :

Les changements de caractéristiques d'une classe dynamique peuvent également conduire à la scission d'une classe sur un ou plusieurs attribut(s). Si cette scission est suffisamment importante pour une classe, il est nécessaire de la diviser en plusieurs parties pour coller au mieux aux changements du système étudié. Une scission de classe correspond à l'apparition de barres centrales nulles dans un histogramme de probabilité et de possibilité. Quand une scission est détectée par FPMD à l'aide des barres centrales nulles d'un histogramme, la méthode Fuzzy C-Means (FCM) (Bezdek, 1981) est appliquée sur la classe réalisant une scission. FCM permet de trouver un nombre de classes optimal qui est validé par la mesure de validité de Xie et Beni (Xie, 1991) :

$$XB(c) = \frac{J_m(U, V) / n}{\min_{i,j=1,c; j \neq i} \|v_i - v_j\|^2} \quad \text{où } J_m(U, V) = \sum_{k=1}^n \sum_{i=1}^c \pi_i(x_k)^m \|x_k - v_i\|^2 \quad (2.15)$$

où J_m est la fonction objectif à minimiser, $\|\cdot\|$ est la norme euclidienne et $\pi_i(x_k)$ est la valeur d'appartenance du $k^{\text{ème}}$ point de la $i^{\text{ème}}$ classe qui a pour centre v_i . Toutes les valeurs d'appartenances sont contenues dans U et les centres des classes sont contenus dans V . m est un coefficient, généralement fixé à 2, permettant d'obtenir un partitionnement plus ou moins flou. Le nombre de classes trouvé par FCM doit être supérieur à 1 pour appliquer cette mesure de validité. La mesure de validité choisie utilise les valeurs d'appartenance des points pour trouver la compacité et la séparabilité du nombre de classes testé. Plus la mesure est faible, plus le nombre de classes est optimal. Cependant, elle ne peut pas être calculée en ligne. Plusieurs autres index de validité des classes ont été utilisés, comme l'index de Fukuyama-Sugeno (Fukuyama, 1989) et l'index Fuzzy eXclusive (FX) utilisant l'opérateur Fuzzy

eXclusive OR (fXOR) (Frélicot, 2006). Cependant, la mesure de validité des classes de Xie et Beni a fourni les meilleurs résultats.

Suppression des classes non représentatives :

Une mesure de suppression des classes non représentatives est également intégrée à FPMD. Elle considère qu'une classe est devenue inutile si deux critères sont validés :

- un nombre n_1 insuffisant de formes est contenu dans la classe. En effet, si trop peu de formes sont contenues dans une classe alors elle n'est pas représentative, elle correspond plutôt à du bruit,
- aucune forme n'a été classée dans une classe pendant qu'un nombre n_2 suffisant de formes ont été classées dans d'autres classes.

Pour valider la suppression d'une classe, les deux conditions précédentes doivent être remplies.

L'ensemble des phases de fonctionnement de FPMD sont représentées sur le schéma de la figure 2.4.

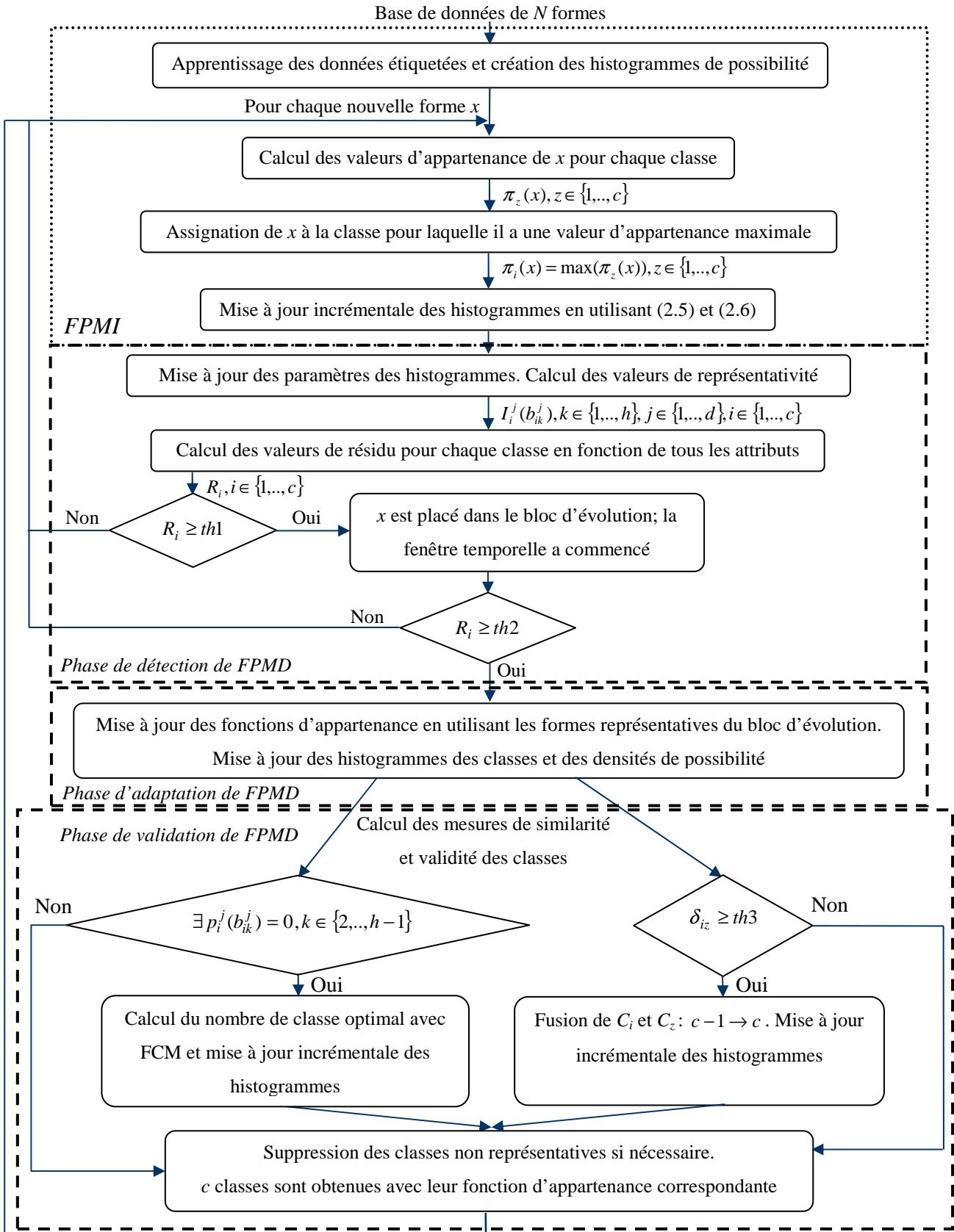


Figure 2.4. Schéma du fonctionnement de Fuzzy Pattern Matching Dynamique.

Nous avons également développé une autre version de FPMD (Hartert, 2008) basée sur un facteur d'oubli. Ce facteur d'oubli est basé sur un poids flou défini pour chaque forme compris entre 0 (forme devenue inutile) à 1 (forme représentative de l'état actuel du système). Dans cette méthode, nous fixions un poids initial de 1 à toutes les formes de l'ensemble d'apprentissage. En effet ces formes sont censées être parfaitement représentatives des classes connues d'un système. Afin de ne pas diminuer l'importance de toutes les formes, il était nécessaire de ne pas seulement se baser sur l'âge des formes. Ainsi, nous avons défini le poids des formes en tenant compte de leur utilité et de leur âge. Par exemple, une classe possède un histogramme contenant 5 barres. On considère que les nouvelles formes ne font qu'apparaître dans les barres 3, 4 et 5. On peut donc vraisemblablement diminuer l'importance des formes correspondant aux barres 1 et 2 de cette classe par rapport à cet histogramme. C'est pourquoi à la classification de chaque nouvelle forme, un poids (fixé initialement) permettait de diminuer la représentativité des barres des histogrammes n'ayant pas reçues de nouvelles formes. Si l'ensemble des barres d'un histogramme, reçoit des nouvelles formes, alors la représentativité des barres ne diminue pas. C'est le fait de ne recevoir successivement plus de formes dans une barre qui fait que son poids d'oubli devient progressivement beaucoup plus important. Cette adaptation des histogrammes permettait ainsi de tenir compte de l'âge et de la représentativité des formes. Toutefois, le poids d'oubli était difficile à définir et celui-ci fut considéré comme trop subjectif. C'est pour ces raisons que la méthode FPMD avec facteur d'oubli ne sera pas présentée sur les exemples illustratifs.

Dans la partie suivante, des techniques d'aide à la définition des paramètres utilisés par FPMD sont données.

2.1.3. Définition des paramètres de la méthode FPMD supervisée

La méthode FPMD utilise quelques paramètres dont certains sont communs à la méthode FPMI. Ces paramètres influencent les résultats de classification, ainsi que la détection des évolutions des classes dynamiques. Nous proposons des valeurs par défaut pouvant être utilisées pour ces paramètres, ainsi que des explications sur leur définition :

- h , le nombre de barres défini pour chaque histogramme par rapport à chaque attribut. Cette valeur détermine la précision de la représentation de la densité de possibilité obtenue pour chaque histogramme. Elle rentre donc parmi les paramètres les plus importants pour déterminer la valeur d'appartenance d'une forme à une classe. Toutefois, dans la méthode FPMD, une valeur initiale est définie pour h , puis cette

valeur est adaptée avec les évolutions de chaque classe. La valeur initiale de h devient donc moins critique. Généralement on définit comme valeur initiale par défaut $h=5$. Cette valeur a été déterminée expérimentalement,

- $th1$ est le seuil de détection des évolutions de classes. Quand le résidu de la méthode dépasse ce seuil pour une classe, on considère que l'évolution de cette classe a commencé. Si la valeur de $th1$ est trop importante, la détection de l'évolution d'une classe est retardée et des points représentatifs de cette évolution ne seront pas conservés dans la mise à jour de la classe. A l'inverse, une faible valeur de $th1$, entraîne une augmentation du nombre d'évolutions détecté et donc du nombre de formes obsolètes conservées dans la mise à jour des classes. Par expérience, $th1=0,1$ permet de bien détecter les évolutions de classe,
- $th2$ est le deuxième seuil utilisé avec le résidu. On peut dire que $th2$ va de paire avec $th1$. Une fois qu'une évolution de classe suffisamment importante est détectée, il est nécessaire de mettre à jour cette classe. C'est une fois que la valeur du résidu d'une classe atteint $th2$ qu'elle est considérée comme suffisamment évoluée. C'est pourquoi, si $th2$ est trop petit, toutes les nouvelles formes représentatives de l'évolution de la classe ne seront pas utilisées dans la mise à jour de la classe. Dans ce cas, l'évolution peut reprendre par la suite et induire de nouvelles adaptations de la classe. A l'inverse, si $th2$ est trop grand, on attend plus de formes représentatives pour adapter la classe. Le fait d'attendre ces formes conduit à un retard de détection. Nous préconisons l'utilisation d'une valeur de $th1=0,5$ par défaut afin d'attendre une évolution suffisante de la classe, mais n'entraînant généralement pas un retard d'adaptation trop important. Ainsi, si l'évolution d'une classe n'est pas complètement finie après son adaptation, soit la classe sera de nouveau adaptée avec les nouvelles formes représentatives s'il reste encore une partie suffisante de l'évolution de la classe, soit la fin de l'évolution ne sera pas considérée comme suffisante pour une nouvelle adaptation de la classe. Dans ce dernier cas, les nouvelles données sont classées normalement et la mise à jour incrémentale de la classe est quand même réalisée mais aucune forme n'est supprimée,
- $th3$ est le paramètre utilisé avec la mesure de la similarité entre les classes. Si cette mesure de similarité est supérieure à $th3$ entre deux classes, on considère que les deux classes sont suffisamment proches pour qu'elles soient fusionnées. Une valeur de $th3=0,2$ permet de fusionner les classes dont les formes sont proches (Figure 2.3).

Toutefois, si l'on met une valeur plus élevée à $th3$, les classes pourront quand même fusionnées, mais cela pourra engendrer un retard,

- n_1 est le nombre minimal de formes jugé nécessaire à la représentation d'une classe C_i . n_2 est le nombre maximal de formes qui peuvent être classées ailleurs que dans C_i avant que l'on considère C_i comme ne recevant plus de nouvelles formes. Si les critères concernant n_1 et n_2 sont atteints, on considère que C_i doit être supprimée. n_1 est défini par défaut à 5, tandis que n_2 est défini par défaut à 20. Si n_1 est défini avec une valeur trop faible aucune classe ne sera sans doute considérée comme obsolète alors qu'avec une valeur trop importante, trop de classes risquent d'être supprimées. De même, pour n_2 , il faut laisser à chaque classe suffisamment de temps pour recevoir au moins une nouvelle forme, avant de la considérer comme obsolète.

Dans la partie suivante, une autre approche de reconnaissance dynamique des formes est proposée (Hartert, 2010a) (Hartert, 2010c). Cette approche est basée sur la méthode K-Plus Proches Voisins Flous. Utilisant une méthode avec un principe et des paramètres différents de ceux de FPMD, ses résultats de classification sont donc différents.

2.2. K-Plus Proches Voisins Flous Dynamique (KPPVFD)

2.2.1. Versions antérieures des K-Plus Proches Voisins

Plusieurs versions de la méthode des K-Plus Proches Voisins (KPPV) (Cover, 1967) (Fix, 1951) ont été développées dans la littérature. Notre méthode repose sur la version Floue des K-Plus Proches Voisins (KPPVF) développée par (Keller, 1985). Plusieurs mécanismes ont été ajoutés à la méthode pour détecter les évolutions de classes et pour adapter ces dernières en fonction de la dynamique de leurs évolutions. Il était intéressant de développer cette méthode puisqu'elle tient compte des valeurs d'appartenance floues des points aux classes, qu'elle est simple et qu'elle reste très utilisée. De plus, ses capacités et limites ont déjà été présentées. En effet, KPPVF devient inefficace quand la taille de l'ensemble d'apprentissage est trop importante ou quand k n'est pas bien choisi. k est généralement déterminé par expérimentation mais il n'en reste pas moins un paramètre qui a une importance majeure dans la classification des formes. Un critère souvent utilisé est $k = \sqrt{N}$ (Dubuisson, 1990), où N est le nombre de formes de l'ensemble d'apprentissage. Parmi les autres techniques existantes basées sur KPPV, on peut citer la méthode KPPV avec prototype ou la méthode KPPV Adaptative (Roncaglia, 2004), etc. Dans (Law, 2005), une version de KPPV pré-assigne une

classe à plusieurs sous régions de l'espace de représentation dans le but de classifier plus rapidement les nouvelles formes. Dans (Zhang, 2002) un algorithme de recherche hiérarchique est développé pour trouver les k plus proches voisins en utilisant une mesure non métrique dans un espace de représentation binaire. Cette mesure est une mesure de similarité calculée entre les valeurs binaires qui représentent les formes. Dans (Kybic, 2006) et (Garcia, 2008), les auteurs utilisent respectivement un arbre de grande dimension et un arbre de k dimensions pour trouver les parties les plus intéressantes de l'espace de représentation où l'on risque de trouver les k plus proches voisins d'un nouveau point. Initialement seules quelques branches de ces arbres ont besoin d'être parcourues pour trouver ces k voisins, mais rapidement les branches de l'arbre peuvent devenir déséquilibrées. Une autre version de KPPV, développée par (Min, 2005), utilise une méthode de réduction de la dimensionnalité à l'aide d'un noyau afin d'améliorer les résultats de classification. Une autre version de KPPV, appelée Instance-Based Learning on Data Streams (IBL-DS) (Beringer, 2007), cherche à détecter les changements survenant dans les flux de données à l'aide d'une erreur de prédiction et de l'écart type des 100 dernières formes classées. Si un changement est détecté, les 20 dernières formes classées servent à estimer l'ampleur de l'évolution réalisée. A partir des indicateurs utilisés un pourcentage initialement défini de formes est supprimé par rapport à leur localisation spatiale et à leur comportement temporel. (Song, 2007) utilise deux mesures informatives pour trouver les formes susceptibles d'être les k plus proches voisins. Ces mesures sont basées sur des mesures de probabilité calculées localement et globalement.

Nous proposons deux méthodes de classification dynamique basées sur les KPPVF. La première méthode est supervisée, elle permet de suivre les évolutions des classes et d'adapter leurs caractéristiques au cours du temps. Lors d'une adaptation, seule la classe adaptée est conservée ; la classe à l'initiative de l'évolution est supprimée. Cette méthode a pour objectif d'actualiser le classifieur et de ne garder que les données représentatives des fonctionnements actuels d'un système. La deuxième méthode proposée est semi-supervisée. Elle a été développée dans le but de pouvoir utiliser la méthode KPPVFD sur les systèmes pour lesquels peu de données historiques sont disponibles. La méthode semi-supervisée met simplement à jour les classes quand peu de changements les concerne, par contre lorsqu'une évolution importante d'une classe a lieu la classe conserve ses caractéristiques d'avant évolution et une nouvelle classe est créée avec les nouvelles données représentatives du changement. De ce fait, le suivi des modes de fonctionnement du système est également réalisé, mais les anciennes formes ne sont pas supprimées, sauf si leur classe est considérée comme obsolète.

Des mesures de similarité et de validité intégrées à ces méthodes permettent de fusionner, de diviser ou de supprimer certaines classes de l'espace de représentation. Ces deux méthodes fonctionnent en plusieurs phases. Premièrement, la méthode KPPVF est utilisée pour réaliser la phase d'apprentissage des formes connues et pour réaliser la phase de classification de chacune des nouvelles formes. Après la classification de chaque nouvelle forme, les méthodes KPPVF Dynamique (KPPVFD) (Hartert, 2010a) proposées mettent à jour incrémentalement les paramètres des classes dans le but de suivre leurs évolutions. Si une évolution est détectée et confirmée en utilisant plusieurs indicateurs, les classes sont adaptées. Une phase de validation permet de conserver seulement les classes représentatives. De plus, une mesure de similarité est utilisée pour détecter les classes qui doivent être fusionnées suite à l'apparition de formes ambiguës entre elles.

2.2.2. KPPVFD en version Supervisée (KPPVFD-S)

Le développement de la version supervisée de K-Plus Proches Voisins Flous en version Dynamique (KPPVFD-S) permet de considérer l'évolution des formes même dans les zones de l'espace de représentation où aucune forme n'a été apprise. L'objectif de cette approche est de suivre les évolutions des classes en prenant en considération la représentativité des formes et en réduisant significativement l'ensemble de données après chaque adaptation d'une classe.

- dans la phase d'apprentissage, toutes les formes étiquetées sont apprises et les classes sont créées. Le centre de gravité actuel (CG_i^{jActu}) et l'écart type initial (σ_i^{jInit}) de la classe C_i sont calculés par rapport à chaque attribut j . A ce moment, l'écart type actuel (σ_i^{jActu}) de la classe est égal à son écart type initial,
- durant la phase de classification de KPPVFD-S, chaque nouvelle forme x est classée. Dans cette phase, KPPVFD-S réalise la détection des évolutions de classes en se basant sur deux indicateurs et l'adaptation de chaque classe lorsque son évolution est confirmée.

2.2.2.a. Détection des évolutions de classes

La classe C_i qui reçoit la forme x est la seule qui doit être mise à jour. Les valeurs actuelles de l'écart type (σ_i^{jActu}) et du centre de gravité (CG_i^{jActu}) de C_i sont alors mises à jour incrémentalement pour chaque attribut j par :

$$\sigma_i^{jActu} = \sqrt{\frac{N_i - 1}{N_i} \times (\sigma_i^{jActu-1})^2 + \frac{(x^j - CG_i^{jActu-1})^2}{N_i + 1}} \quad (2.16)$$

$$CG_i^{jActu} = \frac{CG_i^{jActu-1} \times N_i}{N_i + 1} + \frac{x^j}{N_i + 1} \quad (2.17)$$

où N_i est le nombre de formes présents dans C_i avant la classification de x . $(\sigma_i^{jActu-1})^2$ et $CG_i^{jActu-1}$ sont respectivement la variance et le centre de gravité de la classe, par rapport à l'attribut j , avant classification de x .

En se basant sur les valeurs calculées de CG_i^{jActu} , σ_i^{jActu} et σ_i^{jInit} , deux indicateurs sont utilisés pour suivre les changements temporels du système :

- ind_{1j} représente l'évolution de la compacité de la classe. Il est calculé à partir de l'écart entre σ_i^{jActu} et σ_i^{jInit} pour chaque attribut j :

$$ind_{1j} = \frac{\sigma_i^{jActu} \times 100}{\sigma_i^{jInit}} - 100 \quad (2.18)$$

ind_{1j} est donné en pourcentage. Si au moins un des attributs j obtient une valeur de ind_{1j} supérieure au seuil $th1$ alors la classe C_i a commencé à changer ses caractéristiques. $th1$ peut être fixé à une petite valeur, par exemple 5, quand il est intéressant de suivre les évolutions progressives de la classe. Au contraire, quand la classe peut évoluer de façon abrupte, une valeur plus importante peut être nécessaire.

- ind_{2j} représente l'écart d'un point par rapport à la dispersion moyenne de la classe. Il se base sur la distance entre x^j et CG_i^{jActu} en fonction de σ_i^{jActu} pour chaque attribut j de l'espace de représentation :

$$ind_{2j} = \frac{|x^j - CG_i^{jActu}| \times 100}{\sigma_i^{jActu}} - 100 \quad (2.19)$$

ind_{2j} est donné en pourcentage. Si au moins un attribut j obtient une valeur de ind_{2j} supérieure à $th1$ ($\max(ind_{2j}) \geq th1$) alors le point a une valeur d'appartenance faible à la classe C_i . Cependant, il ne suffit pas d'avoir un seul point loin de la classe pour considérer un changement de caractéristiques. Ce point peut être du bruit, c'est pourquoi il est nécessaire de définir un seuil $NbMin$ représentant le nombre de fois successives que ind_{2j} doit dépasser $th1$ pour confirmer une évolution. Si $NbMin$ est fixé à une valeur importante, le retard de détection de l'évolution de la classe peut être important. Ce nombre $NbMin$ doit être défini en fonction d'un compromis entre le

bruit présent dans les formes de l'espace de représentation et le retard maximal de détection d'évolution souhaité.

L'évolution de la classe est confirmée quand $NbMin$ valeurs successives des deux indicateurs ind_{1j} et ind_{2j} sont supérieures à $th1$.

2.2.2.b. Adaptation des classes évoluées

Dans sa phase d'adaptation la méthode KPPVFD-S intègre un mécanisme permettant d'ajuster les paramètres d'une classe évoluée lorsque suffisamment de changements de caractéristiques ont été détectés et confirmés lors de la phase de détection.

Quand une évolution est confirmée pour une classe C_i , elle doit être adaptée en se basant seulement sur les formes représentatives. Cette adaptation est réalisée en deux parties :

- les formes les plus représentatives de l'évolution sont sélectionnées. k formes seulement peuvent être sélectionnées pour représenter une classe évoluée à sa création puisque le paramètre k de la méthode correspond au nombre de formes jugées suffisamment représentative pour classer une nouvelle forme. La dernière forme x classée représente une de ces k formes puisqu'elle a permis de confirmer l'évolution, elle est donc sélectionnée. x représente également le changement le plus récent dans l'évolution de la classe. Les autres formes informatives sélectionnées par KPPVFD-S sont les $k-1$ dernières formes classées. Ainsi, même si les dernières formes classées représentent plusieurs évolutions différentes d'une même classe (cas d'une scission) elles seront toutes considérées. Aucune nouvelle distance n'a donc besoin d'être calculée pour trouver ces formes. L'ensemble de ces k formes représente les nouvelles formes de la classe C_i une fois adaptée,
- le nouveau centre de gravité CG_i^{jActu} et l'écart type initial σ_i^{jIni} de la classe sont calculés. Cette mise à jour est réalisée rapidement puisque le nombre de formes dans la classe évoluée est égal à k .

Cette phase d'adaptation permet de suivre en ligne l'évolution des classes avec un temps d'adaptation faible et constant. Puis, les nouvelles formes sont classées dans leur classe correspondante.

2.2.2.c. Validation des classes

Dans cette phase, KPPVFD-S réalise :

- la suppression des classes aberrantes et non représentatives,

- la fusion des classes qui ont une similarité forte,
- la scission des classes qui ne peuvent plus être considérées comme une seule classe.

Le bruit est déjà en partie pris en compte par le paramètre $NbMin$, cependant si une classe considérée comme du bruit est créée, il peut être intéressant de la supprimer. Dans le même ordre d'idée, une classe transitoire, qui correspond au passage d'un mode de fonctionnement à un autre, peut être considérée comme non représentative une fois le mode de fonctionnement atteint. Enfin le fait de supprimer les classes non représentatives permet également de tenir compte de la taille grandissante de l'ensemble des données. KPPVFD-S intègre un mécanisme de validité des classes pour supprimer les classes non représentatives répondant à deux cas :

- un nombre n_1 insuffisant de formes est contenu dans la classe ($n_1 > k$),
- aucune forme n'a été classée dans la classe depuis qu'un nombre n_2 suffisant de formes ont été classées dans d'autres classes.

Suite à l'évolution de certaines classes, il peut être nécessaire de les fusionner. Des classes évoluant vers une même direction devraient sans doute fusionner si leur chevauchement est suffisant à un moment donné. Pour décider de cette fusion, la mesure de similarité définie à l'équation (2.14) est utilisée entre toutes les paires de classes après la classification de chaque nouvelle forme. Si cette mesure dépasse un seuil th_{Fusion} entre deux classes alors ces dernières sont fusionnées.

La détection de la scission est plus difficile que pour FPMD. Pour savoir que la classe s'est divisée et qu'elle devrait être adaptée, plusieurs possibilités nous étaient offertes :

- vérifier à intervalles réguliers la validité de chaque classe,
- calculer la distance entre tous les points, et vérifier si plusieurs distances ne sont pas supérieures à un seuil,
- appliquer FCM et la mesure de validité prédéfinie en (2.14) après chaque adaptation d'une classe.

Nous avons choisi d'appliquer FCM avec la mesure de validité sélectionnée après chaque adaptation de la classe pour détecter une scission. Cette technique permet de faire gagner un gain de temps de calcul important par rapport aux autres techniques tout en permettant d'obtenir de bons résultats. Une fois le nombre optimal o de classes estimé par FCM, la scission est réalisée si chacune de ces o classes contient au moins k formes. En effet, une classe doit contenir au moins k formes pour être capable d'en recevoir de nouvelles. Si le

nombre de formes n'est pas suffisant au moment où le nombre de classes est estimé par FCM alors il faut attendre que ce nombre de formes soit classé pour de nouveau vérifier la scission de la classe.

La figure 2.5 présente l'algorithme général de la méthode KPPVFD-S.

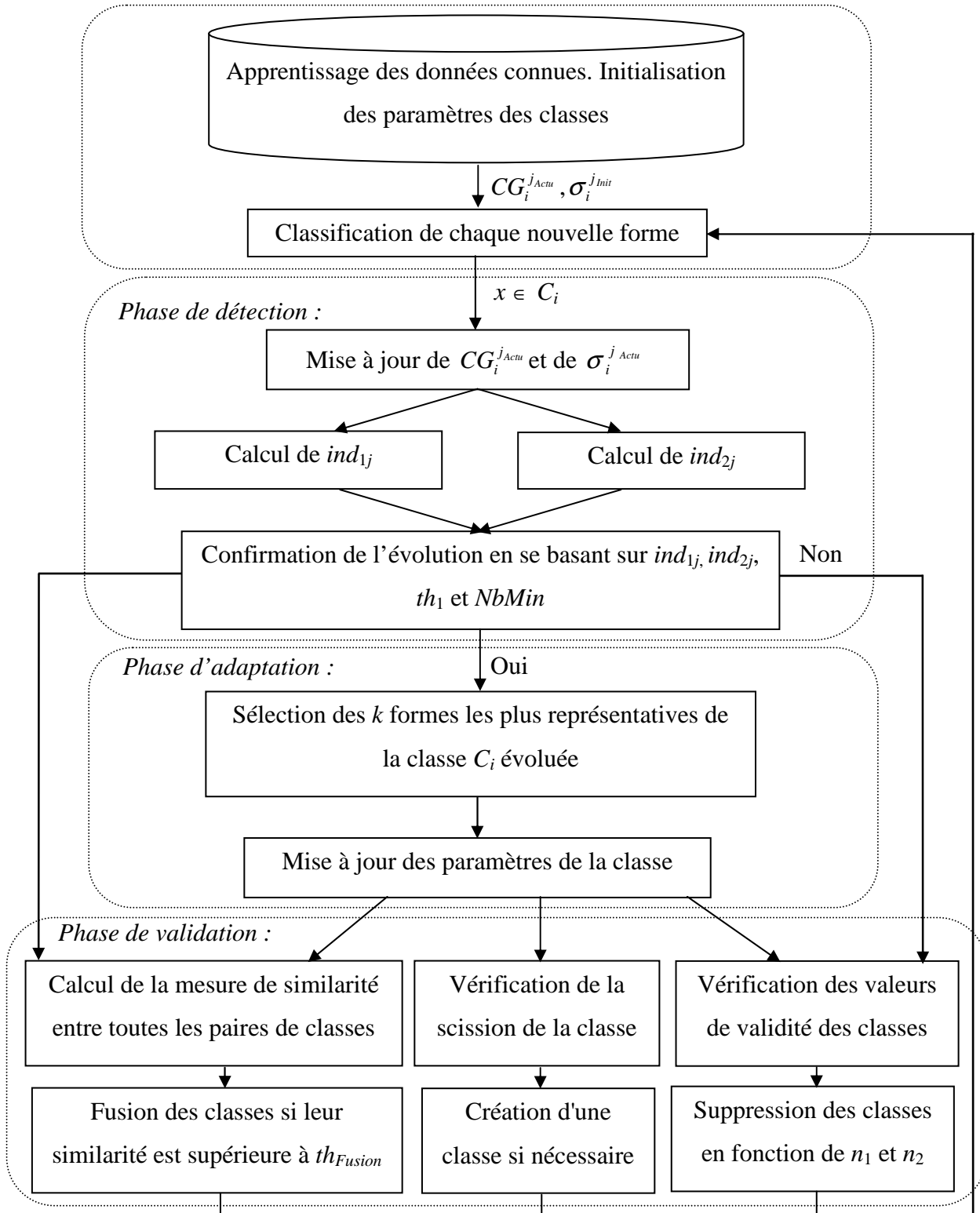


Figure 2.5. Description générale de la méthode supervisée KPPVF Dynamique (KPPVFD-S).

La version supervisée de la méthode KPPVFD vient d'être présentée. Dans la partie suivante, la version semi-supervisée de KPPVFD est détaillée.

2.2.3. KPPVFD en version Semi-Supervisée (KPPVFD-SS)

Dans cette section, une approche semi-supervisée de KPPVF capable de suivre les évolutions des classes dynamiques est présentée. Cette version (Hartert, 2010c) a été développée pour détecter en ligne l'évolution des classes dynamiques, pour adapter leurs caractéristiques, mais également pour détecter l'apparition de nouvelles classes. Cette version est semi-supervisée dans le but :

- de prendre en considération les informations initiales connues d'un système, tel que l'ensemble d'apprentissage X représentatif des classes connues,
- d'améliorer les caractéristiques des classes en utilisant les nouvelles formes classées dans les classes connues *a priori*,
- de détecter les nouvelles classes en fonction des évolutions des caractéristiques d'un système.

De la même façon que pour la version supervisée, les évolutions des formes peuvent être considérées même dans des parties de l'espace de représentation où aucune forme n'a été apprise. Le fait de détecter les nouvelles classes permet de conserver uniquement les formes qui sont représentatives d'une classe déjà connue. En effet dans cette version, une classe qui commence à se déplacer gardera sa forme initiale, tandis que les formes en évolution de cette classe représenteront une nouvelle classe. C'est cette nouvelle classe qui va permettre de mieux estimer le mode de fonctionnement actuel d'un système. Contrairement aux méthodes de classification dynamique précédemment détaillées (FPMD et KPPVFD-S), l'ensemble des formes apprises est ici conservé, aucun oubli de formes ou de classes n'a lieu sauf en cas de non représentativité d'une classe.

Les principales phases de fonctionnement de la méthode KPPVFD-SS sont présentées sur la figure 2.6.

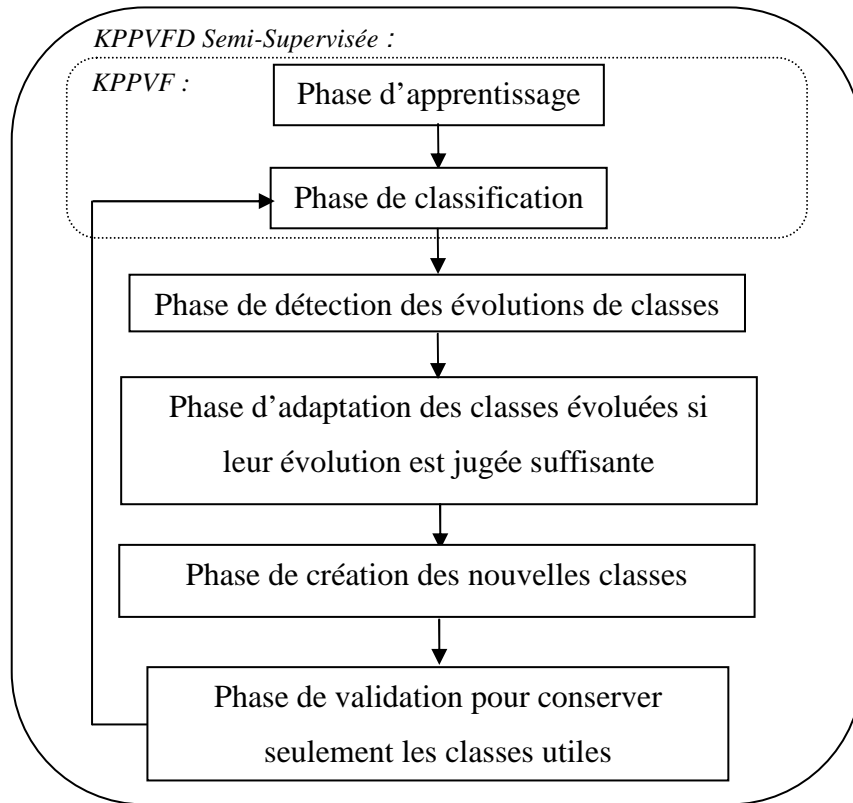


Figure 2.6. Les différentes phases de la méthode KPPVF Dynamique semi-supervisée.

2.2.3.a. Apprentissage et classification

Les phases d'apprentissage et de classification sont réalisées de la même façon que pour KPPVFD-S. L'ensemble d'apprentissage est appris initialement. Pour chaque classe apprise deux formes au minimum doivent être connues pour pouvoir calculer son centre de gravité initial ($CG_i^{j_{Actu}}$) ainsi que son écart type initial ($\sigma_i^{j_{mit}}$) par rapport à chaque attribut j . Ces deux valeurs ainsi que la valeur actuelle de l'écart type ($\sigma_i^{j_{Actu}}$) vont être prises en compte dans le calcul des deux indicateurs d'évolution de la méthode. Elles sont mises à jour incrémentalement (équations (2.16 et 2.17)). Il est également important de préciser que pour tout type de classe, $CG_i^{j_{Actu}}$, $\sigma_i^{j_{mit}}$ et $\sigma_i^{j_{Actu}}$ peuvent être calculés. Dans le cas des classes de formes complexes nous considérons que plusieurs sous classes gaussiennes seront créées puis fusionnées pour obtenir d'autres types de classes.

Dans la phase de classification de la méthode, chaque nouvelle forme est classée séquentiellement en fonction de ses k plus proches voisins. Comme pour la méthode KPPVFD-S, il est donc nécessaire de définir initialement le paramètre k .

2.2.3.b. Détection de l'évolution d'une classe

Deux indicateurs d'évolutions sont utilisés pour détecter le changement de caractéristiques de la classe recevant la nouvelle forme. Ces indicateurs sont les mêmes que pour la version supervisée. Le premier indicateur ind_{1j} (équation 2.18) permet d'obtenir une valeur de compacité de la classe. Le second indicateur ind_{2j} (équation 2.19) représente l'écart entre le point actuel et le centre de gravité de sa classe par rapport à l'écart type de la classe. Ces indicateurs sont mesurés pour chaque attribut j de l'espace de représentation.

Tout comme pour la version supervisée de KPPVFD, les deux seuils $th1$ et $NbMin$ permettent de valider une évolution lorsqu'un nombre successif et suffisant de formes ont réalisées cette évolution.

2.2.3.c. Adaptation d'une classe après validation de son évolution et création d'une nouvelle classe

Quand l'évolution d'une classe est confirmée, une nouvelle classe est créée par KPPVFD-SS en se basant seulement sur les formes représentatives de l'évolution. La classe à l'origine de l'évolution conserve toutes ses formes autres que celles de l'évolution.

Les différentes parties de cette phase d'adaptation sont :

- la création d'une nouvelle classe C' et la sélection des formes représentatives de l'évolution. Parmi ces formes, la dernière forme x classée est sélectionnée, ainsi que ses $k-1$ plus proches voisins. Aucune nouvelle distance ne doit être calculée puisque lors de la classification de x ses plus proches voisins ont déjà été déterminés. De cette façon, les points sélectionnés de l'évolution seront basés sur le changement le plus récent et sur une zone d'évolution précise,
- la dernière forme classée x et ses $k-1$ plus proches voisins déjà connus sont les seules formes retenues pour la création de C' ,
- les k formes sélectionnées sont supprimées de la classe C_i ,
- le centre de gravité $CG_i^{J_{Actu}}$ et l'écart type actuel $\sigma_i^{J_{Actu}}$ de la classe C_i sont mis à jour,
- $CG_i^{J_{Actu}}$ et $\sigma_i^{J_{Init}}$ sont calculés pour la classe C' . Ces valeurs sont calculées rapidement car seuls k points sont présents dans la classe,
- le nombre de classes de l'espace de représentation est mis à jour.

Cette phase permet de suivre en ligne les évolutions d'une classe. Si cette dernière est complexe, une succession de sous classes gaussiennes pourra apparaître lors de l'évolution.

En utilisant cette approche, toutes les formes et les classes sont conservées dans l'espace de représentation et une classe C qui évolue génère au moins une nouvelle classe ou sous-classe C' . En réalisant cette création de classes, la scission et le déplacement de classes sont pris en compte. Si aucune évolution n'a lieu, les formes sont classées normalement.

Si une classe C est considérée comme inutile, c'est à dire pas ou plus suffisamment représentative d'un mode de fonctionnement, il peut être intéressant de supprimer cette classe C pour solutionner le problème de taille grandissante de l'ensemble des données. De la même façon, certaines fois deux classes doivent être fusionnées puisqu'elles ont fini par mêler leurs caractéristiques. Les solutions pour traiter les cas de classes non représentatives et de classes devant être fusionnées sont présentées dans la section suivante.

2.2.3.d. Validation des classes

Le bruit des classes est déjà considéré par la méthode KPPVFD par le biais du seuil $NbMin$. Cependant, dans certains cas, le bruit ou d'autres facteurs peuvent faire que la suppression d'une ou de plusieurs classes devient nécessaire :

- lorsqu'une classe transitoire, c'est-à-dire reliant deux classes, est conservée après que la classe de destination soit atteinte. En effet, cette classe ne représente pas un mode de fonctionnement en tant que tel,
- lorsqu'une classe considérée comme bruitée est créée,
- lorsqu'une classe contenant très peu d'informations est conservée un certain temps.

Pour considérer ces cas, supprimer les classes et mettre à jour le classifieur, KPPVFD regarde deux critères de validité :

- un nombre n_1 insuffisant de formes est contenu dans la classe ($n_1 > k$),
- et aucune forme n'a été classée dans la classe depuis que n_2 formes ont été classées dans d'autres classes.

Toute classe ne vérifiant pas ces critères de validation sera supprimée. Il est nécessaire de préciser que cette suppression de classes non représentatives n'est pas nécessaire pour toutes les applications. Par exemple pour les applications considérant des données de fonctionnement au caractère important ou critique, il peut être préférable de conserver l'ensemble des données caractéristiques de toutes les classes.

La figure 2.7 présente un exemple de suppression de classe. Une classe est connue initialement et de façon abrupte un déplacement de la classe est réalisé par quelques nouvelles formes. Puis, une nouvelle évolution de la classe est réalisée et elle mène la classe vers sa

destination finale. Dans ce cas, les quelques formes de la classe transitoire doivent être supprimés. Les paramètres de la méthode KPPVFD-SS pour ce cas sont : $k=5$, $th_1 =5$, $NbMin=5$, $n_1=10$, $n_2=20$ et $th_{Fusion}=0,2$.

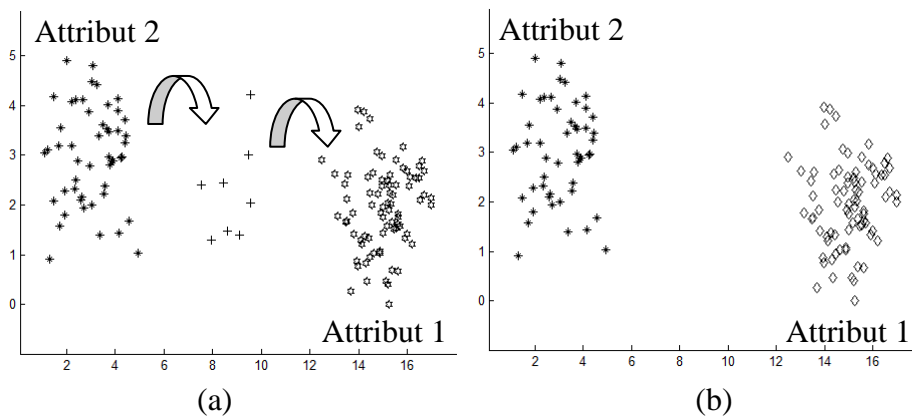


Figure 2.7. Suppression d'une classe par KPPVFD-SS. Dans (a), les formes de X sont notées '*', les formes transitoires sont notées '+', et les formes de la classe évoluée sont représentées par '*\'. (b) Résultats de classification obtenus par KPPVFD-SS.

KPPVFD-SS a considéré la première évolution de la classe comme une nouvelle classe. Puis, une seconde classe a été créée et la classe transitoire qui a alors été considérée comme non représentative a été supprimée.

Pour le cas de fusion des classes similaires, la mesure de similarité précédemment définie (équation 2.14) a été intégrée à KPPVFD-SS afin de vérifier si après chaque forme classée, il est nécessaire ou non de fusionner deux classes ayant une similarité supérieure ou égale à th_{Fusion} .

Sur la figure suivante 2.8, l'algorithme global détaillé de la méthode KPPVFD-SS est présenté.

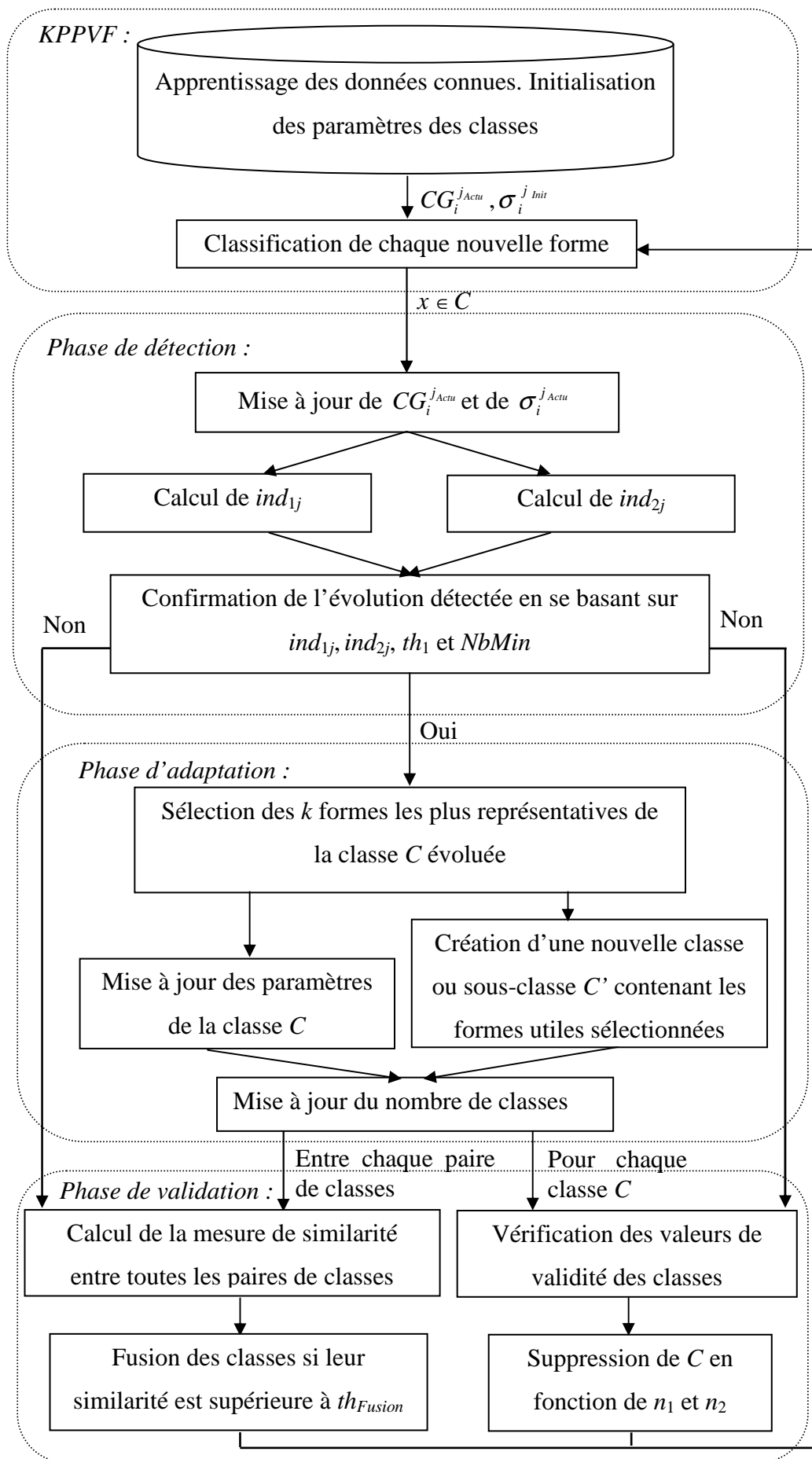


Figure 2.8. Description de la méthode KPPVFD-SS.

2.2.4. Définition des paramètres des méthodes KPPVFD-S et KPPVFD-SS

Les deux versions des méthodes KPPVFD ont besoin de plusieurs paramètres qui peuvent être définis facilement. Comme pour chaque méthode de reconnaissance des formes, les paramètres définis influencent les performances du classifieur. Cependant, nous proposons des valeurs par défaut qui sont généralement adaptées aux systèmes dynamiques :

- k correspond au nombre de voisins pris en compte dans les méthodes KPPV pour réaliser la classification d'un point. k est le paramètre commun des méthodes KPPV qui est le plus important. Il devrait être défini en fonction de la taille de la base de données, du bruit présent sur les observations du système et de la proximité des classes,
- th_1 est un des paramètres les plus importants de KPPVFD. Il est pris en compte par les deux indicateurs d'évolution de KPPVFD. Il permet de détecter les évolutions des classes. Une classe qui n'évolue pas aura toujours les mêmes caractéristiques, même si du bruit apparaît. Par contre les caractéristiques d'une classe changent dans le cas d'une évolution abrupte ou progressive. Une valeur de th_1 égale à 5 est un bon compromis permettant d'adapter les classes et leurs caractéristiques sans avoir besoin d'attendre la détection d'une évolution importante,
- $NbMin$ permet de valider une évolution. $NbMin$ influence les résultats au niveau du temps d'adaptation de la classe. Il devrait être défini au moins égal à k ($k \geq 1$) dans le but d'attendre suffisamment de formes représentatives pour bien estimer les caractéristiques d'une nouvelle classe. Il ne doit pas être non plus défini par une trop grande valeur pour ne pas retarder la détection de l'évolution. $NbMin$ devrait être défini entre k et $k+5$, respectivement si k est petit ou grand. Ces valeurs ont été déterminées par expérimentation. Si k et $NbMin$ sont petits, le risque de fausses alarmes devient plus élevé,
- th_{Fusion} est un paramètre d'optimisation utilisé pour fusionner les classes. En effet, même si aucune fusion n'apparaît, la simple apparition d'une classe signifie qu'une évolution du système a eu lieu. Dans ce cas, une alarme devrait être levée pour appeler un opérateur humain qui vérifierait l'état du système. Une valeur de th_{Fusion} proche de 0,2 permet de fusionner les classes qui ont commencé à avoir les mêmes caractéristiques,

- n_1 est un des paramètres utilisé dans la phase de validation des classes. Il devrait être défini supérieur à k ($n_1 > k$) puisqu'une classe contiendra au moins k formes (à sa création). Une valeur par défaut de n_1 peut être $k*2$,
- n_2 est un des paramètres utilisé dans la phase de validation des classes. Sa valeur ne devrait pas être définie trop petite puisqu'après la création d'une classe, il peut être nécessaire d'attendre pour classifier plus de formes dans la classe créée. Au contraire, si aucune forme n'est classifiée dans la nouvelle classe après un nombre important de points, la classe n'est pas ou plus représentative. Il s'agit probablement d'une classe de bruit ou d'un problème éphémère qui est apparu sur le système. La valeur de n_2 peut être définie par défaut à 20. Ce résultat a été obtenu par expérimentation. Cela signifie qu'une forme sur 20 devrait être classifiée dans la nouvelle classe, dans le but de confirmer progressivement son utilité. Pour les autres classes, même si elles ne reçoivent plus de formes après un certain temps, elles ne seront pas supprimées puisqu'elles ont déjà confirmé leur utilité en ayant un nombre suffisant de formes.

Dans la section suivante, des exemples sont utilisés pour illustrer et valider les méthodes de classification dynamiques proposées en utilisant un protocole expérimental.

2.3. Protocole expérimental pour la validation des méthodes proposées

Dans cette partie, nous détaillons les critères de validation des résultats utilisés dans le cadre de la classification dynamique. En effet, contrairement aux méthodes de classification statique il est nécessaire de mesurer l'apport des méthodes dynamiques sur la détection des évolutions des classes. Une fois ces critères détaillés, les simulations de bases de données dynamiques réalisées sont présentées pour être utilisées dans la partie suivante par chacune des méthodes dynamiques proposées et par trois méthodes incrémentales également.

Ces exemples de simulations ont pour but d'illustrer les résultats des méthodes. C'est dans le chapitre 4 que les méthodes proposées vont être évaluées autour d'exemples réels en utilisant le protocole expérimental et les critères d'évaluation définis.

Lorsqu'une classe a ses caractéristiques qui changent au cours du temps pour au moins un attribut, il est nécessaire de mettre à jour le classifieur pour suivre ces changements et ainsi correspondre au mieux à la classe en évolution. Pour ce faire, il est important de seulement considérer les nouvelles formes représentatives dans la mise à jour des fonctions d'appartenance des classes. Plus on utilise de formes représentatives pour la mise à jour des classes, plus les fonctions d'appartenance des classes sont bien estimées.

Afin de pouvoir comparer les résultats obtenus par les méthodes incrémentales avec ceux des méthodes de classification dynamique proposées, quatre critères sont utilisés dans ce manuscrit :

- le retard de détection d'une évolution de classe. Si la détection d'une évolution de classe n'est pas réalisée avec une méthode, cette information capitale doit être connue. A l'inverse si une évolution est détectée avec du retard, ce retard doit être communiqué et si une évolution est détectée à l'avance, la fausse alarme doit être signalée. Le retard de détection, noté t_{retard} , est obtenu par :

$$t_{retard} = t_{méthode} - t_{réel} \quad (2.20)$$

où $t_{méthode}$ représente l'instant où la méthode a détecté l'évolution d'une classe, et $t_{réel}$ représente l'instant où l'évolution a vraiment eu lieu. Si aucune détection d'une évolution n'a lieu, $t_{méthode}=\infty$ donc $t_{retard}=\infty$. t_{retard} peut être donné en secondes ou en nombre de formes. Dans ce dernier cas, il correspond au nombre de formes qu'il a fallu attendre pour réaliser la détection d'une évolution,

- une mesure d'erreur spécialement définie pour la classification dynamique. Cette mesure, notée ind_U , est le ratio entre le nombre de formes utiles et le nombre de formes inutiles conservées par une méthode de classification pour mettre à jour le modèle d'une classe (les fonctions d'appartenance). Cet indicateur est considéré comme un critère de qualité d'estimation des paramètres après la fin de l'évolution des classes :

$$ind_U = \frac{\sum_{i=1}^{c_{evol}} \frac{n_{évaluées}}{n_{évaluées} + n_{inutiles}} \times 100}{c_{evol}} \quad (2.21)$$

où $n_{évaluées}$ représente le nombre de formes classées dans C_i dont les caractéristiques ont changées. $n_{inutiles}$ représente le nombre de formes inutiles, i.e. celles qui ne sont plus représentatives du fonctionnement actuel du système, qui sont utilisées pour la mise à jour des paramètres des fonctions d'appartenance. c_{evol} correspond au nombre de classes concernées par les évolutions. Plus ind_U est élevé, plus la mise à jour des fonctions d'appartenance représente les changements apportés par les nouvelles formes,

- le taux d'erreur de classification, noté $err_{classif}$. En effet, les performances du classifieur se dégradent significativement quand les classes évoluent au cours du temps. Ce critère peut donc montrer l'intérêt de la mise à jour des paramètres du classifieur afin

de conserver ses performances. Cette valeur n'est donnée que lorsque l'espace de représentation contient au minimum deux classes. $err_{classif}$ est calculée par :

$$\begin{aligned}
 & \text{si } C_{estimée}(x_i) = C_{réelle}(x_i) \text{ alors } e(C_{estimée}(x_i), C_{réelle}(x_i)) = 0 \\
 & \text{sinon } e(C_{estimée}(x_i), C_{réelle}(x_i)) = 1 \\
 err_{classif} &= \frac{\sum_{i=1}^n e(C_{estimée}(x_i), C_{réelle}(x_i))}{n} \tag{2.22}
 \end{aligned}$$

où n est le nombre de formes classifiées, e permet de savoir si une forme est bien classifiée, $C_{estimée}(x_i)$ correspond à la classe estimée pour x_i et $C_{réelle}(x_i)$ correspond à la classe réelle de x_i ,

- l'erreur d'estimation des caractéristiques des classes évoluées, notée $err_{estimation}$. Quand les classes évoluent, les modèles des classes ne représentent plus les caractéristiques de ces classes. L'erreur d'estimation des caractéristiques d'une classe est calculée par :

$$err_{estimation} = \max \left(\frac{|\mu_{estimée}^j - \mu^j|}{|\mu^j|} \right), j \in \{1, \dots, d\} \tag{2.23}$$

où μ^j est la moyenne réelle d'une classe par rapport à l'attribut j après la fin de son évolution, $\mu_{estimée}^j$ est la moyenne finale estimée par la méthode de classification pour cet attribut et d correspond au nombre d'attributs de l'espace de représentation. La valeur d'erreur d'estimation retenue est la valeur maximale d'erreur obtenue par rapport à tous les attributs. Cette erreur n'est calculée que lorsque l'on connaît les caractéristiques attendues des classes réalisant une évolution.

Parmi ces quatre critères de qualité de la détection des évolutions des classes, le retard de détection est présenté pour chaque cas de classification réalisé, par contre, les trois autres critères sont donnés en fonction des caractéristiques de l'application (plusieurs classes dans l'espace de représentation, estimation possible des caractéristiques finales d'une classe, etc.).

Ces critères sont également utilisés pour quantifier la qualité de classification et de suivi des classes à l'aide des méthodes incrémentales de classification. En plus d'IFPM, les deux autres méthodes incrémentales utilisées dans les parties suivantes sont les Séparateurs à Vaste Marge Incrémentale (SVMI) (Cauwenberghs, 2001) et la méthode des K-Plus Proches Voisins en version Incrémentale (KPPVI) (Zhou, 2001). La comparaison des résultats de ces

méthodes incrémentales avec ceux obtenus par les méthodes de reconnaissance dynamique des formes proposées va permettre de voir l'intérêt majeur de ces dernières méthodes pour le suivi des systèmes évolutifs.

Dans les parties suivantes, de nombreuses simulations de classes dynamiques sont étudiées et les résultats de classification obtenus par les méthodes incrémentales et dynamiques sur ces bases de données sont présentés. Les valeurs des paramètres définis pour chaque méthode sont ceux ayant permis d'obtenir les meilleurs résultats de classification. La plupart des exemples illustratifs suivants sont basés sur des évolutions de classes gaussiennes. Une classe gaussienne est représentée par une distribution normale de deux variables indépendantes (moyenne μ et écart type σ). Dans un souci de simplicité de représentation, nous avons utilisé ces données dans un espace de deux ou trois dimensions. Tous les résultats obtenus peuvent être étendus à des dimensions supérieures. Nous supposons qu'une unité de temps correspond à la classification d'un point.

Les premiers résultats de classification présentés sont ceux concernant les méthodes d'apprentissage incrémental.

2.3.1. Limites des méthodes incrémentales pour les classes dynamiques

Classe en déplacement :

Ce premier cas de classe dynamique présente le déplacement progressif d'une classe généré suivant les équations suivantes :

- $t=0$: 150 formes sont utilisées comme ensemble d'apprentissage. Seule la classe initiale est connue. Les valeurs de moyenne et d'écart type de la classe sont pour l'attribut 1, $\mu^1=3$ et $\sigma^1=1$, et pour l'attribut 2, $\mu^2=3$ et $\sigma^2=1$ (Figure 2.9.a).
- $t=1$ jusque 50 : 50 nouvelles formes apparaissent avec les mêmes caractéristiques que celles de la classe initiale ; ainsi il n'y a pas d'évolution ou de déplacement.
- $t=51$ jusque 200 : un changement soudain apparaît dans les valeurs de moyenne de la classe par rapport à chaque attribut $j, j \in \{1,2\}$. Ce changement abrupt est suivi par un déplacement progressif de la moyenne de la classe par rapport à chaque attribut (Figure 2.9.b):

$$\left. \begin{aligned} \mu^1(t) &= \mu^1 + 2 + \frac{4 \times (t - 50)}{150} \\ \mu^2(t) &= \mu^2 + 2 + \frac{2 \times (t - 50)}{150} \end{aligned} \right\} 51 \leq t \leq 200 \quad (2.24)$$

- $t=201$ jusque 300 : 100 nouvelles formes apparaissent. Elles ont les mêmes caractéristiques que celles de la classe finale.

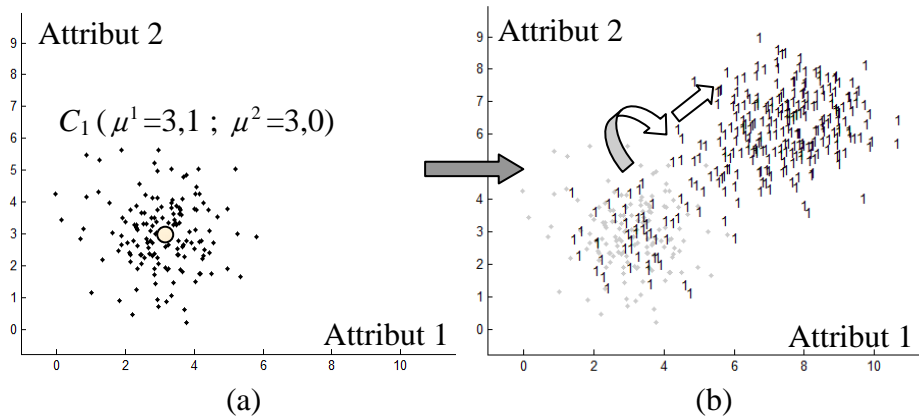


Figure 2.9. Ensemble d'apprentissage pour le cas de déplacement d'une classe (a). Evolution de la classe C_1 (b).

La figure 2.10 présente le résultat de classification obtenu par les trois méthodes incrémentales.

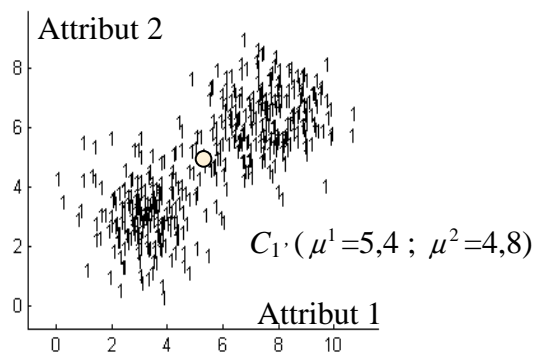


Figure 2.10. Classification des formes de la classe déplacée par FPMI, SVMi et KPPVI.

Toutes les formes classées par les méthodes incrémentales ont été considérées comme représentatives. Ici, seule une classe est considérée dans l'espace de représentation, le pourcentage de classification n'est donc pas diminué. Par contre, la valeur ind_U est loin d'être maximale (Tableau 2.1). Ainsi la classe C_1 actuelle, considérée par ces classifieurs n'est pas adaptée à la vraie classe dont les caractéristiques ont changé après déplacement. Ces résultats ne permettent pas de détecter une évolution du système, ni de bien estimer les paramètres de la classe.

Tableau 2.1. Résultats de classification obtenus par FPMI ($h=5$), SVM (noyau Gaussien avec variance=10 et constante de régularisation=5) et KPPVI ($k=5$), pour le déplacement d'une classe.

	FPMI	SVM	KPPVI
ind_U	66,6 %	66,6 %	66,6 %
$err_{estimation}$	29,9 %	29,9 %	29,9 %
t_{retard}	Aucune détection	Aucune détection	Aucune détection

Rotation de classe :

La rotation d'une classe a été créée afin de considérer un autre type d'évolution. La rotation d'une classe est présentée sur la figure 2.11. L'écart type de la classe par rapport à chaque attribut est initialement $\sigma_1=4$ et $\sigma_2=0,25$. Les 150 formes présentes dans la classe avant évolution sont utilisées comme l'ensemble d'apprentissage (Figure 2.11.a). Puis, entre $t=1$ et $t=150$, de nouvelles formes apparaissent et font évoluer la classe, pour finalement obtenir une rotation de la classe (Figure 2.11.b).

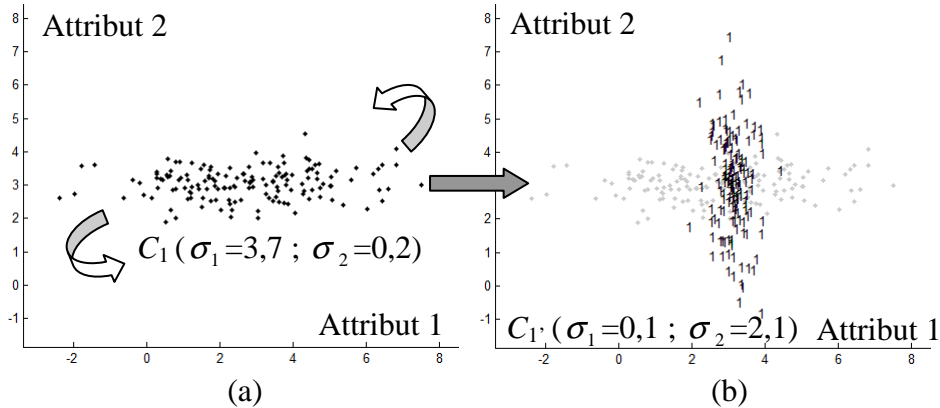


Figure 2.11. Ensemble d'apprentissage utilisé pour le cas de la rotation d'une classe (a).

Espace de représentation contenant les formes avant et après évolution de C_1 .

Ces changements sont générés en fonction de chaque attribut j , $j \in \{1,2\}$, par :

$$\left. \begin{aligned} \mu_1^t &= \mu_1^1 \times \sqrt{\frac{\sigma_2 \times t}{150}} \\ \mu_2^t &= \mu_2^1 \times \sqrt{\frac{\sigma_1 \times t}{150}} \end{aligned} \right\} 1 \leq t \leq 150 \quad (2.25)$$

La figure 2.12 présente les résultats obtenus par les trois méthodes d'apprentissage incrémental pour le cas de rotation d'une classe.

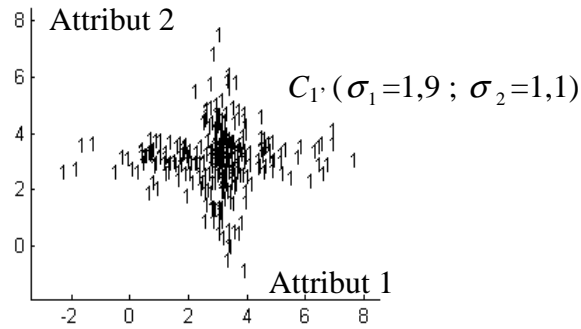


Figure 2.12. Résultats de classification par FPMI, SVM et KPPVI.

Les résultats de classification obtenus par les méthodes incrémentales pour ce cas sont présentés dans le tableau 2.2. Ces résultats montrent que la fonction d'appartenance de la classe est mal caractérisée puisqu'elle comprend 50 % de formes non représentatives de la classe. La rotation de la classe n'a donc pas été détectée.

Tableau 2.2. Résultats de classification obtenus par FPMI ($h=5$), SVM (noyau Gaussien avec variance=10 et constante de régularisation=5) et KPPVI ($k=5$), pour le cas de rotation d'une classe.

	FPMI	SVM	KPPVI
ind_U	50 %	50 %	50 %
$err_{estimation}$	3,3 %	3,3 %	3,3 %
t_{retard}	Aucune détection	Aucune détection	Aucune détection

L'erreur d'estimation est calculée à partir des valeurs de moyenne de chaque classe par rapport à chaque attribut. Elle ne permet pas pour ce cas de rotation de montrer les limites des méthodes incrémentales puisque la moyenne de la classe évolue peu.

Fusion de deux classes :

Ce cas présente deux classes se déplaçant vers une zone commune pour réaliser leur fusion. Ce cas est illustré par la figure 2.13. La fusion de deux classes (c'est-à-dire de deux modes de fonctionnement) peut représenter le cas d'une évolution temporelle d'un système vers un nouveau mode de fonctionnement (normal ou défaillant). La base de données des deux classes qui fusionnent a été créée comme suit :

- $t=0$: 125 formes sont utilisées pour apprendre les fonctions d'appartenance de la classe C_1 ($\mu^1=5$, $\mu^2=7$) et 125 autres formes sont utilisées pour la classe C_2 ($\mu^1=15$, $\mu^2=7$). L'écart

type de ces classes est égal à 1 pour chaque attribut. Ces formes d'apprentissage sont représentées sur la figure 2.13.a.

- $t=1$ jusque 150 : les nouvelles formes appartenant à chaque classe se dirigent vers une zone commune aux deux classes (Figure 2.13.b). Ces changements sont générées pour les deux classes en fonction de chaque attribut j , $j \in \{1,2\}$, par :

$$C_1: \left. \begin{aligned} \mu_1^{1'}(t) &= \mu_1^1 + 2 + \frac{3 \times t}{150} \\ \mu_1^{2'}(t) &= \mu_1^2 + 2 + \frac{t}{150} \end{aligned} \right\} 1 \leq t \leq 150, \quad (2.26)$$

$$C_2: \left. \begin{aligned} \mu_2^{1'}(t) &= \mu_2^1 - 2 - \frac{3 \times t}{150} \\ \mu_2^{2'}(t) &= \mu_2^2 + 2 + \frac{t}{150} \end{aligned} \right\} 1 \leq t \leq 150, \quad (2.27)$$

- $t=151$ jusque 275 : de nouvelles formes sont générées avec les mêmes caractéristiques que la classe évoluée.

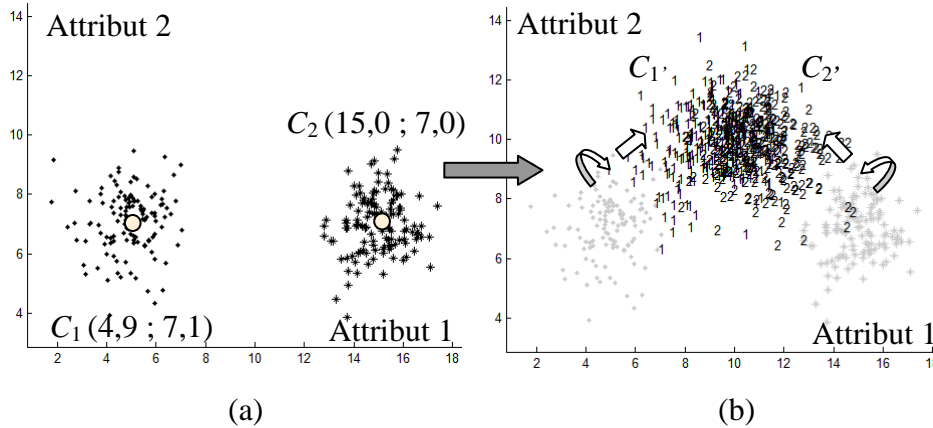


Figure 2.13. Ensemble d'apprentissage des deux classes (a). Fusion des classes C_1 et C_2 (b).

Les méthodes incrémentales ont déjà montrées leurs limites de suivi d'évolution concernant le suivi du déplacement d'une classe. Ici, le cas de la fusion de deux classes reprend en partie le même schéma (déplacements de deux classes), puis celles-ci doivent être fusionnées. Ainsi, les méthodes incrémentales conservent également ici l'ensemble des formes classées depuis la phase d'apprentissage de la méthode. Une mesure de similarité est ensuite utilisée pour décider si les classes sont similaires et si elles doivent être fusionnées. La mesure de similarité trouvée est suffisante pour réaliser cette fusion après la classification de 273 formes, ce qui permet d'obtenir le résultat de classification de la figure 2.14. C'est grâce à la fusion des classes que les méthodes réussissent pour ce cas à détecter l'évolution des classes.

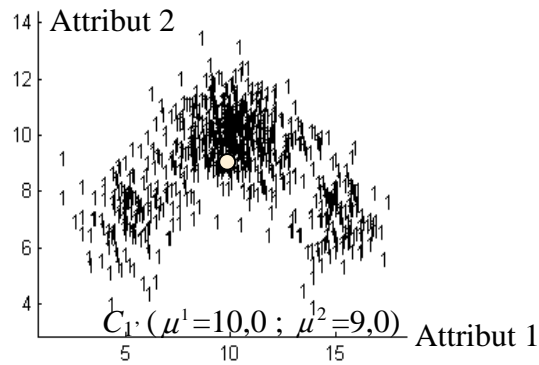


Figure 2.14. Résultat final de classification obtenue par les trois méthodes incrémentales après fusion des deux classes.

La classe obtenue sur la figure 2.14 contient l'ensemble des points classifiés. Parmi ces derniers, de nombreux points sont pourtant complètement non représentatifs des caractéristiques nouvelles de la classe (Tableau 2.3).

Tableau 2.3. Résultats de classification obtenus par FPMI ($h=5$), SVM (noyau Gaussien avec variance=10 et constante de régularisation=5) et KPPVI ($k=5$), pour le cas de fusion de deux classes dynamiques.

	FPMI	SVM	KPPVI
ind_U	68,8 %	68,8 %	68,8 %
$err_{estimation}$	10,9 %	10,9 %	10,9 %
t_{retard}	273 formes	273 formes	273 formes

Scission de classe :

La figure 2.15 présente le cas de scission d'une classe vers deux directions opposées.

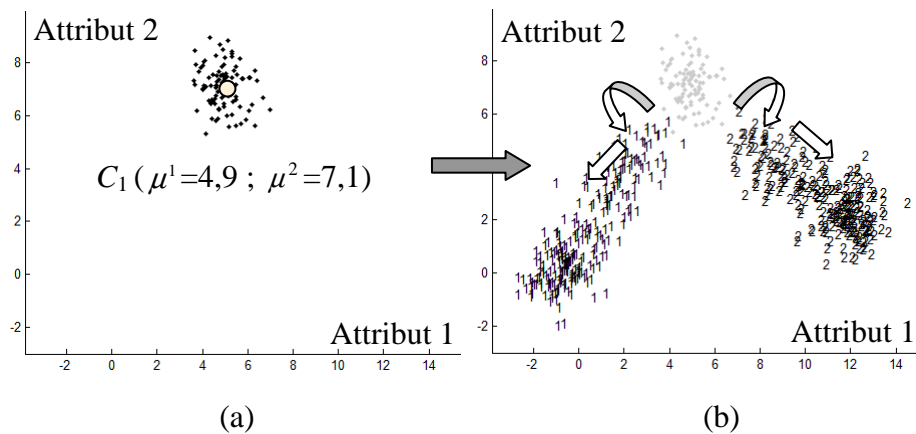


Figure 2.15. Classe initiale de l'ensemble d'apprentissage (a) avec l'ensemble des formes de l'évolution et leurs numéros de classes attendus (b).

La base de données de ce cas a été créée au cours du temps comme suit :

- $t=0$: 100 formes sont utilisées pour l'apprentissage du classifieur (Figure 2.15.a).
- $t=1$ jusque 150 : les nouveaux points évoluent dans deux directions différentes (Figure 2.15.b). Cette évolution est générée pour C_1 en fonction de chaque attribut j , $j \in \{1,2\}$, par :

$$\left. \begin{aligned} \mu_1^{1'} &= \mu_1^1 - 2 - \frac{4 \times t}{150} \\ \mu_1^{2'} &= \mu_1^2 - 2 - \frac{5 \times t}{150} \end{aligned} \right\} 1 \leq t \leq 150 \quad (2.28)$$

$$\left. \begin{aligned} \mu_1^{1'} &= \mu_1^1 + 2 + \frac{5 \times t}{150} \\ \mu_1^{2'} &= \mu_1^2 - 2 - \frac{3 \times t}{150} \end{aligned} \right\} 1 \leq t \leq 150 \quad (2.29)$$

- $t=151$ jusque 250 : de nouvelles formes sont classées avec les mêmes caractéristiques que les dernières formes de l'évolution.

La figure 2.16 présente le résultat de classification obtenu par les méthodes incrémentales. Ce résultat est le même peu importe la méthode.

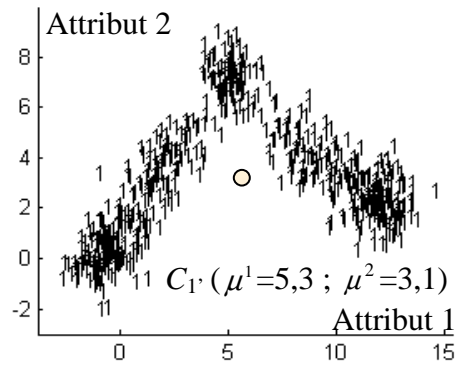


Figure 2.16. Résultat final de classification obtenue par chacune des méthodes incrémentales pour le cas de scission d'une classe.

Avec les méthodes incrémentales, toutes les formes ont été classifiées dans la même classe. Aucune détection de la scission n'est possible. Toutes les formes inutiles ont été conservées dans la classe finale (Tableau 2.4).

Tableau 2.4. Résultats de classification obtenus par FPMI ($h=5$), SVM (noyau Gaussien avec variance=10 et constante de régularisation=5) et KPPVI ($k=5$), pour le cas de scission d'une classe.

	FPMI	SVM	KPPVI
$err_{classif}$	50 %	50 %	50 %
ind_U	71,4 %	71,4 %	71,4 %
t_{retard}	Aucune détection	Aucune détection	Aucune détection

L'erreur de classification obtenue par chaque méthode incrémentale est due à la non prise en charge du changement de structure du classifieur, c'est-à-dire du nombre de classes.

Déplacements menant à un chevauchement partiel :

Pour cet exemple, les fonctions d'appartenance de deux classes initiales sont apprises par les trois méthodes incrémentales. Chacune de ces deux classes est représentée par 150 points. Puis, un déplacement de 200 formes est généré ; il mène chaque classe vers la classe opposée. Sur la figure 2.17, l'ensemble d'apprentissage et les formes de l'évolution sont représentées.

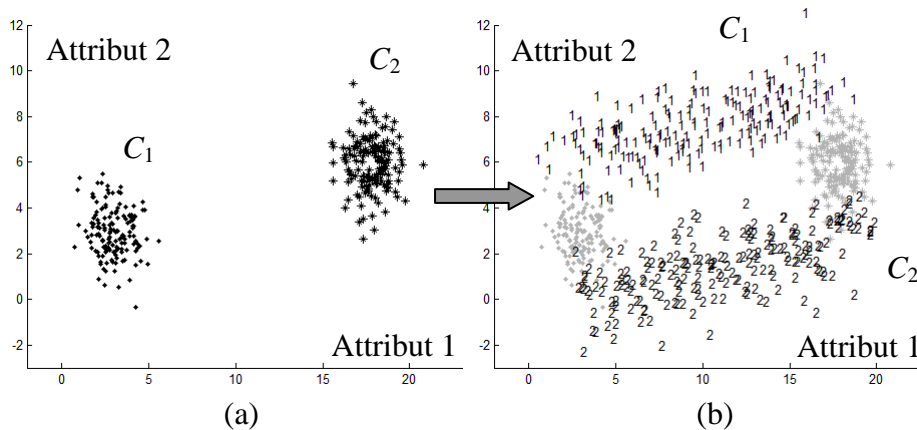


Figure 2.17. Les deux classes initiales (a) avec les formes de l'évolution à classifier. Dans (b) chaque nouvelle forme est représentée par son numéro de classe attendu.

Le déplacement de chaque classe est créé par :

$$C_1: \begin{cases} \mu^1(t) = \mu^1 + 2 + \frac{15 \times t}{200} \\ \mu^2(t) = \mu^2 + 6 + \frac{3 \times t}{200} \end{cases} \quad 1 \leq t \leq 200 \quad (2.30)$$

$$C_2: \begin{cases} \mu^1(t) = \mu^1 + 19 - \frac{15 \times t}{200} \\ \mu^2(t) = \mu^2 + 3 - \frac{3 \times t}{200} \end{cases} \quad 1 \leq t \leq 200 \quad (2.31)$$

Les résultats de classification obtenus par les trois méthodes incrémentales sont présentés dans le Tableau 2.5 et sur la figure 2.18.

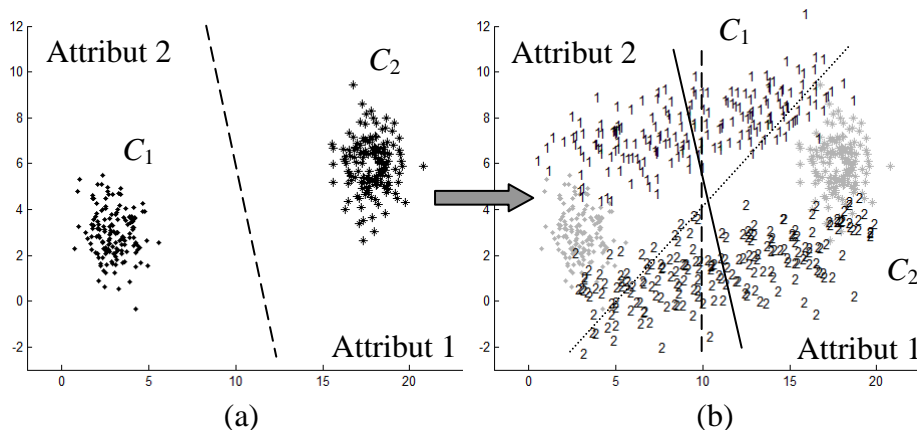


Figure 2.18. Frontière de décision initiale entre les deux classes (a). Frontière de décision obtenue par FPMI (--), SVM (-) et KPPVI (..) (b). Sur (b) chaque nouvelle forme est représentée par son numéro de classe attendu.

Tableau 2.5. Résultats de classification obtenus par FPMI ($h=5$), SVM (noyau Gaussien avec $\text{variance}=10$ et constante de régularisation=5) et KPPVI ($k=5$), pour le cas de déplacements de deux classes menant à un chevauchement partiel des classes.

	FPMI	SVM	KPPVI
$err_{classif}$	43,8 %	50,5 %	10,5 %
ind_U	57,1 %	57,1 %	57,1 %
t_{retard}	Aucune détection	Aucune détection	Aucune détection

Toutes les formes classifiées par FPMI, SVM et KPPVI ont été considérées comme représentatives des caractéristiques actuelles des classes. Cette considération ne permet pas de bien classifier les nouvelles formes et de bien estimer les paramètres des classes. Les frontières de décision de la figure 2.18.b montrent que la séparation des classes n'est pas bien réalisée par chacune des méthodes incrémentales. Si seules les formes représentatives avaient été conservées, il aurait été possible de séparer correctement les classes évoluées.

Les cas de simulations d'évolution de classes qui ont été présentés permettent de voir que les méthodes incrémentales de reconnaissance des formes se limitent à classer avec la même informativité chacune des nouvelles formes. En conservant les formes inutiles, ces méthodes ne peuvent pas détecter les évolutions des classes (ou avec un retard important) et peuvent même fausser le résultat de classification.

Pour solutionner ces limitations, et pour permettre le suivi des modes de fonctionnement des systèmes évolutifs, les méthodes de reconnaissance dynamique des formes proposées peuvent être utilisées. Les résultats de classification obtenus par la méthode FPMD sur les mêmes cas de simulations sont présentés dans la partie suivante.

2.3.2. Application de la méthode FPMD sur des classes dynamiques

FPMD a été testée sur les différents exemples de simulations présentés dans la partie 2.3.1. Les valeurs de $th1$ et $th2$ qui ont été sélectionnées pour chaque cas sont celles permettant d'obtenir les meilleures performances pour FPMD.

Classe en déplacement :

La méthode FPMD a été évaluée sur le déplacement d'une classe généré suivant les équations (2.24). Les valeurs de résidu obtenues progressivement après la classification de chaque forme dans la classe sont représentées sur la figure 2.19.

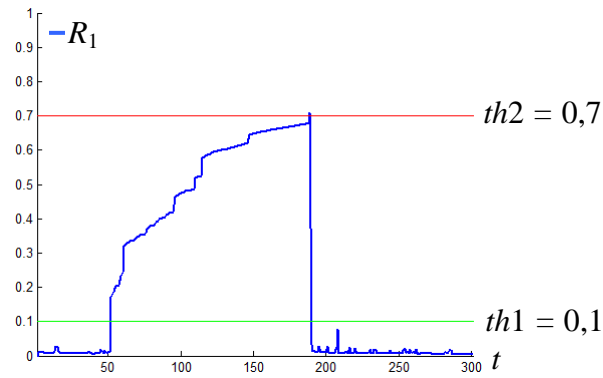


Figure 2.19. Valeurs de résidu obtenues par FPMD pour le déplacement d'une classe.

En se basant sur la figure 2.19, nous pouvons conclure que :

- entre $t=1$ et $t=50$, l'évolution de la classe n'a pas encore commencé,
- à $t=51$, l'évolution de la classe a commencé,
- à $t=189$, le déplacement de la classe est terminé. Pour détecter la fin de ce déplacement et pour valider l'évolution, $th2$ a été fixé à 0,7. Cette valeur a été déterminée par expérimentation. Toutefois, avec la valeur par défaut de $th2=0,5$, le modèle de la classe aurait également été adapté, i.e. mis à jour (Figure 2.22).
- finalement, entre $t=190$ et $t=300$ aucune évolution n'est observée et la valeur de résidu est faible et inférieure à $th1$ pour ces nouvelles formes. La classe est donc stabilisée et ces formes sont classées normalement.

La figure 2.20 présente les mesures de représentativité calculées pour l'évolution de la classe à $t=189$ pour chaque attribut.

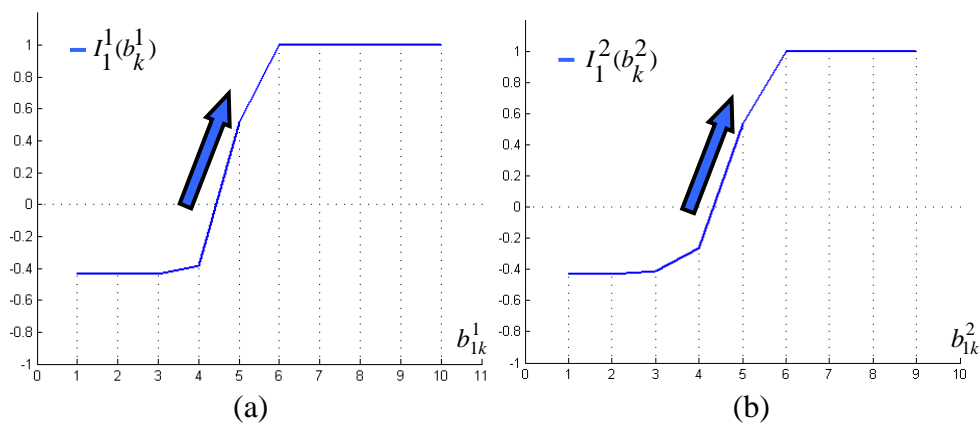


Figure 2.20. Mesures de représentativité pour les barres des histogrammes en fonction de l'attribut 1 (a) et de l'attribut 2 (b) pour le cas du déplacement de classe.

Les valeurs de représentativité sont négatives pour quatre des barres initiales par rapport à l'attribut 1 (barres 1 à 4) et à l'attribut 2 (barres 1 à 4). Au contraire, ces valeurs sont positives pour les nouvelles barres. Le déplacement de la classe a été bien détecté pour les deux attributs. La figure 2.21 décrit le résultat de classification obtenu par FPMD après évolution, lorsque l'adaptation des histogrammes a été réalisée.

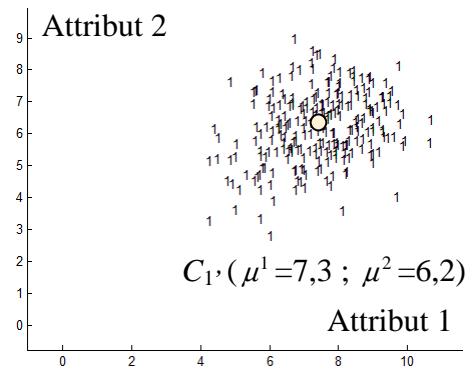


Figure 2.21. Résultat de classification obtenu par FPMD après classification de toutes les formes du cas du déplacement.

En utilisant la valeur par défaut de $th2$, 0,5, plusieurs adaptations du modèle de la classe sont réalisées. Sur la figure 2.22.a, on peut voir que suite à la première adaptation, l'évolution de la classe n'est pas finie, d'autres adaptations de la classe sont donc réalisées. En utilisant une valeur de $th2$ inférieure à celle utilisée sur la figure 2.19 ($th2=0,7$), moins de formes représentatives de l'évolution sont utilisées pour caractériser la classe lors de chacune de ses adaptations. Par conséquent, le nombre de formes transitoires conservées dans la classe adaptée est donc également moins important (Figure 2.22.b).

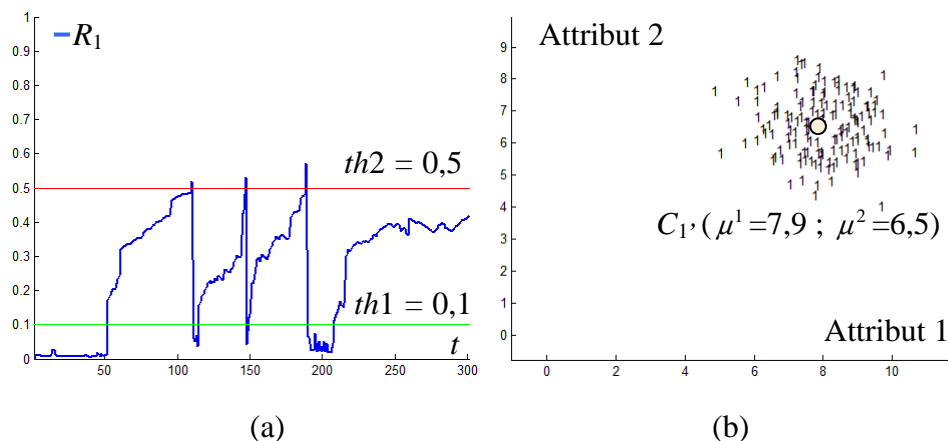


Figure 2.22. Résidu (a) et résultat de classification obtenu par FPMD pour $th2=0,5$ après classification de toutes les formes du cas du déplacement.

Le résultat de la figure 2.22 montre que le changement de modèle de la classe a été réalisé tout comme pour la figure 2.19. Plus d'adaptations ont été réalisées, c'est pourquoi le modèle de la classe n'est pas le même. A partir de $t=208$, une nouvelle évolution de classe est détectée, mais le résidu n'atteint pas $th2$ (avant $t=300$). Dans ce cas, le modèle de la classe est mis à jour incrémentalement, et le contour de la classe est adapté avec chacune des nouvelles formes. En réalisant plus d'adaptations la classe contient moins de formes représentatives mais elle contient également moins de formes transitoires.

Pour le cas du déplacement, avec les deux valeurs de paramètres utilisés, $th2=0,5$ ou $th2=0,7$, le classifieur s'est adapté aux formes représentatives. Les valeurs de moyenne sur chaque attribut sont proches de celles attendues, contrairement à celles que l'on obtient avec une méthode incrémentale. Cependant, avec $th2=0,7$, certaines formes réalisant le pont entre la classe initiale et l'emplacement final de la classe évoluée sont conservées par FPMD. Ces formes sont représentatives de l'évolution de la classe mais elles représentent une transition. Les résultats de classification de FPMD pour ce cas sont résumés dans le Tableau 2.6. Le temps maximal de classification d'une forme est égal à 2×10^{-2} s et le temps moyen de classification est égal à 4×10^{-3} s.

Tableau 2.6. Résultats de classification obtenus par FPMD ($h=5$, $th1=0,1$ et $th2=0,7$) et FPMD ($h=5$, $th1=0,1$ et $th2=0,5$) pour le cas de déplacement d'une classe (Figure 2.9).

	FPMD $th2=0,7$	FPMD $th2=0,5$
ind_U	100 %	100 %
$err_{estimation}$	5,2 %	2,6 %
t_{retard}	Pas de retard	Pas de retard

Les résultats de classification obtenus pour le déplacement permettent de constater que l'évolution de la classe est bien détectée par FPMD. De plus, seules les formes représentatives de l'évolution sont conservées. L'influence de $th2$ est ici démontrée au niveau du nombre de formes transitoires conservées dans la classe adaptée.

Rotation de classe :

Les valeurs de résidu obtenues pour ce cas sont représentées sur la figure 2.23.

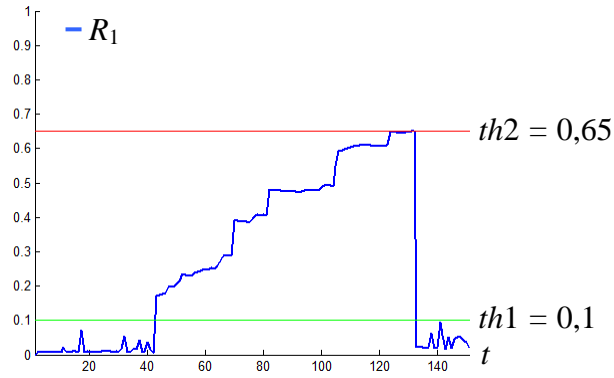


Figure 2.23. Valeurs de résidu obtenues par FPMD pour la classe réalisant une rotation.

En se basant sur la figure 2.23, nous pouvons conclure que l'utilisation de $th1=0,1$ et $th2=0,65$ permet de détecter la rotation de la classe. Ces valeurs ont été obtenues par expérimentation, elles permettent de contenir l'ensemble des formes représentatives de la classe évoluée à l'aide d'une seule adaptation. Aucun retard de détection n'a lieu, toutefois le résidu ne détecte pas d'évolution dès la classification des premières formes. Cela est dû au fait que les premières formes classées sont situées au centre de la classe. Aucune évolution n'a donc lieu à l'apparition des nouvelles formes. Sur la figure 2.24, les mesures de représentativité correspondent à l'évolution de la classe à $t=132$ pour les deux attributs.

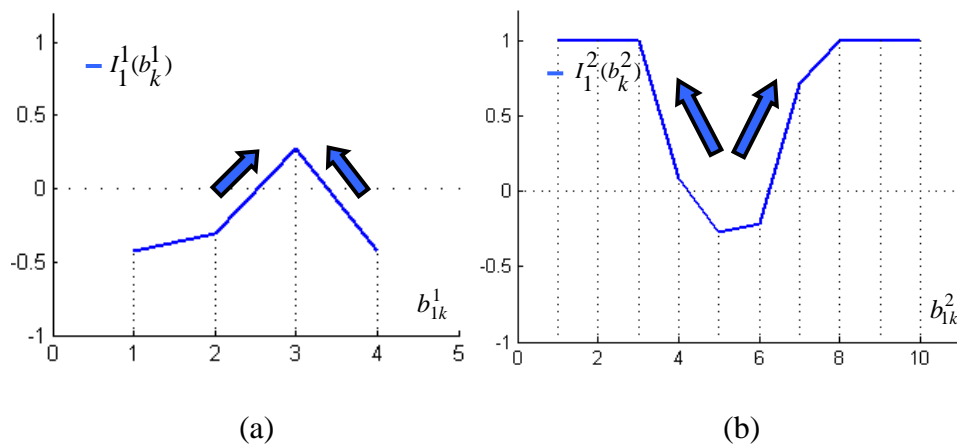


Figure 2.24. Mesures de représentativité des barres des histogrammes en fonction de l'attribut 1 (a) et de l'attribut 2 (b).

Les valeurs de représentativité montrent que la classe a réalisé une scission sur l'attribut 2 et qu'elle s'est resserrée sur l'attribut 1. Les mesures de représentativité des barres centrales de l'histogramme de la classe sont négatives concernant l'attribut 2 puisque le nombre de formes qui sont apparues dans ces barres reste faible par rapport au nombre de formes qui

étaient déjà présentes dans ces barres. Cependant, les valeurs de possibilité de ces barres restent différentes de 0 après adaptation (Figure 2.25.a et figure 2.25.b). Cela est dû au fait que de nombreuses nouvelles formes, conservées dans le bloc d'évolution, appartiennent à ces barres. Cette information est utile pour distinguer une rotation, d'une scission de classe comme nous allons le voir dans une prochaine sous-section.

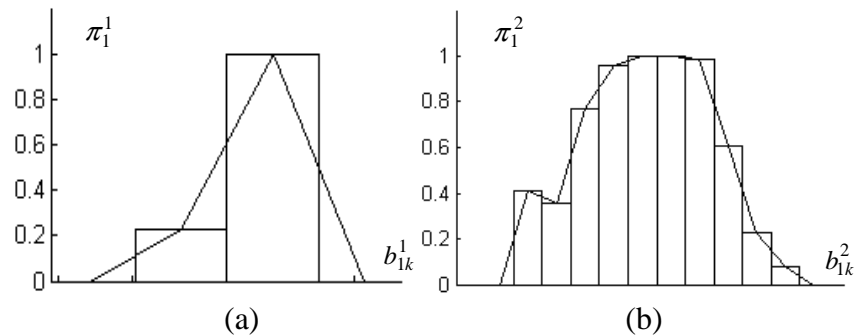


Figure 2.25. Histogrammes de possibilité pour C_1 en fonction de l'attribut 1 (a) et de l'attribut 2 (b) pour la rotation de la classe.

La figure 2.26 permet de voir le résultat de classification obtenu par FPMD pour le cas de la classe réalisant la rotation, une fois que l'adaptation des histogrammes a été réalisée.

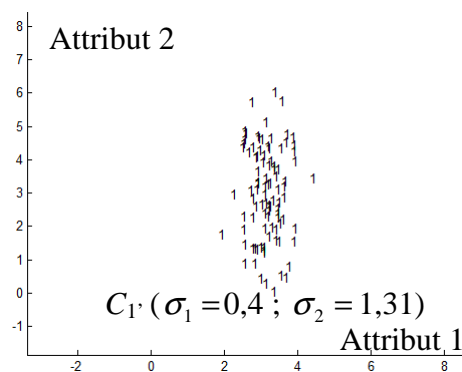


Figure 2.26. Résultat de classification obtenu par FPMD.

Le classifieur a été adapté avec toutes les formes représentatives conservées dans le bloc d'évolution depuis la détection de l'évolution de la classe. Les nouvelles formes représentatives de la classe sont les seules prises en compte dans le calcul de la fonction d'appartenance. Les résultats de classification de FPMD pour ce cas sont résumés dans le Tableau 2.7. Le temps maximal de classification d'une forme est égal à 2×10^{-2} s et le temps moyen de classification est égal à 3×10^{-3} s.

Tableau 2.7. Résultats de classification obtenus par FPMD ($h=5$, $th1=0,1$ et $th2=0,65$) pour le cas de rotation d'une classe (Figure 2.11).

	FPMD
ind_U	100 %
$err_{estimation}$	3,3 %
t_{retard}	Pas de retard

Fusion de deux classes :

FPMD a été appliquée au cas de deux classes se déplaçant l'une vers l'autre pour réaliser une fusion. Deux résidus sont observés en fonction du temps (Figure 2.27). Les valeurs de $th1$ et $th2$ sont les mêmes pour chaque classe puisque l'évolution des deux classes suit la même dynamique (équations (2.26) et (2.27)).

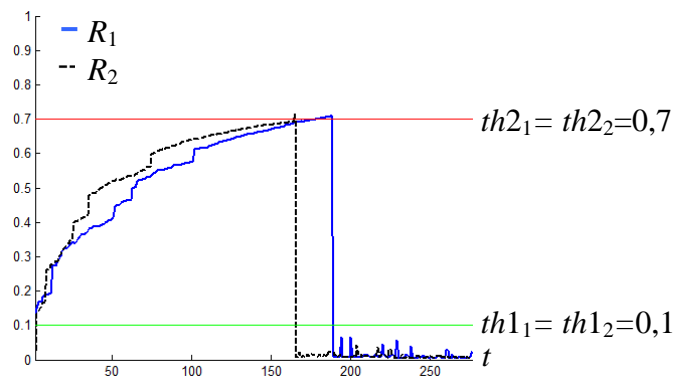


Figure 2.27. Valeurs de résidu pour les deux classes du cas de la fusion.

La figure 2.27 permet de conclure les remarques suivantes :

- à $t=1$, les évolutions de C_1 et C_2 sont détectées,
- à $t=165$, C_2 est adaptée puis la classe n'évolue plus,
- à $t=188$, C_1 est adaptée, et aucune évolution n'apparaît ensuite.

Les valeurs de représentativité de ces deux classes sont présentées sur la figure 2.28 entre le commencement de l'évolution de chaque classe et le moment où les classes cessent d'évoluer. La figure 2.28 permet de voir les valeurs de représentativité pour les barres des histogrammes de chaque classe. Pour les attributs 1 et 2, plusieurs des sept barres initiales des deux classes ne sont plus utiles. Ainsi, un déplacement est bien détecté pour chaque classe en fonction de tous les attributs.

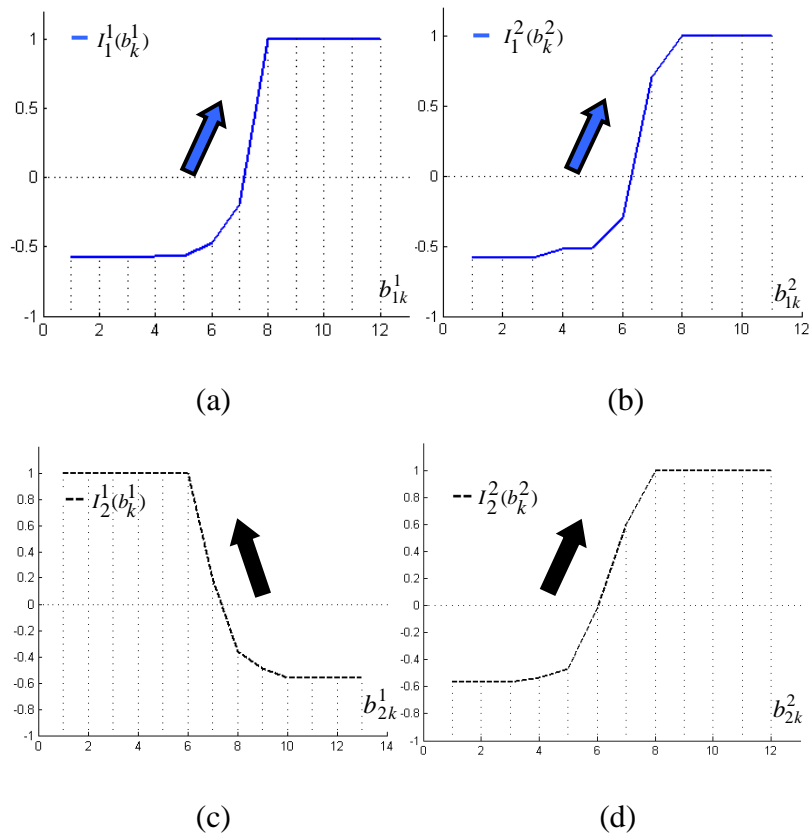


Figure 2.28. Valeurs de représentativité des barres des histogrammes de C_1 (a et b, respectivement pour l'attribut 1 et 2) et de C_2 (c et d, respectivement pour l'attribut 1 et 2).

Dans le but de voir si le déplacement de ces deux classes mène à leur fusion, la mesure de similarité δ_{12} est calculée entre C_1 et C_2 après classification de chaque forme. A l'instant $t=0$, δ_{12} est égal à 0. Après adaptation, due au déplacement des deux classes, cette mesure devient supérieure à $th3=0,2$, les classes peuvent alors être fusionnées. Le résultat final de classification avant (Figure 2.29.a) et après (Figure 2.29.b) fusion est présenté sur la figure 2.29.

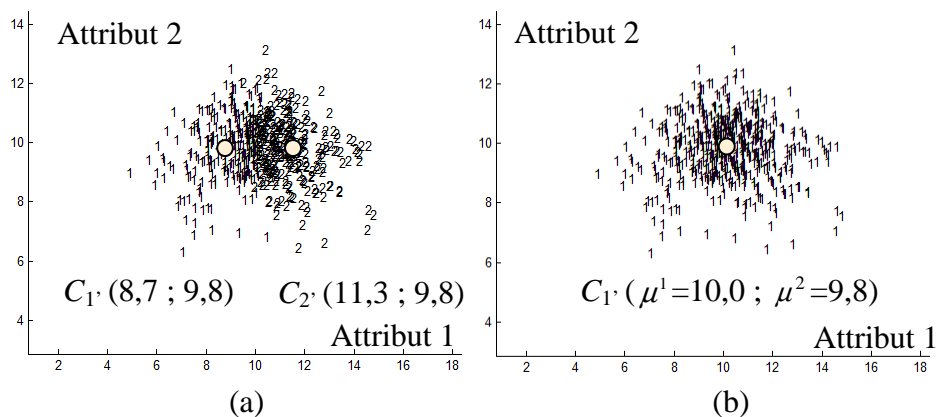


Figure 2.29. Résultat de classification obtenu par FPMD avant (a) et après (b) fusion.

FPMD a bien suivi le déplacement des deux classes et la fusion a été réalisée. La classe finale est proche de celle attendue, cependant un certain nombre de formes transitoires ont été conservées dans la classe adaptée. Les résultats de classification de FPMD pour ce cas sont résumés dans le Tableau 2.8. Le temps maximal de classification d'une forme est égal à 3×10^{-2} s et le temps moyen de classification est égal à 7×10^{-3} s.

Tableau 2.8. Résultats de classification obtenus par FPMD ($h=7$, $th1=0,1$ et $th2=0,7$ pour chaque classe, $th3=0,2$) pour le cas de fusion des classes (Figure 2.13).

	FPMD
ind_U	100 %
$err_{estimation}$	3,0 %
t_{retard}	Pas de retard

Scission de classe :

La méthode FPMD a été évaluée sur la scission d'une classe générée par les équations (2.28) et (2.29). Les valeurs de résidu obtenues jusqu'à la première adaptation de la classe sont représentées sur la figure 2.30.

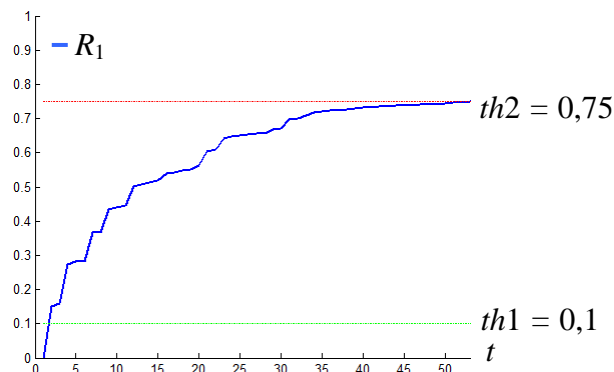


Figure 2.30. Valeurs de résidu obtenues par FPMD pour le cas de scission d'une classe.

La figure 2.30 montre que l'évolution est détectée rapidement. La classe est adaptée à $t=53$. A ce moment, les valeurs de probabilité sont analysées pour chaque attribut. La probabilité de deux barres, au centre de l'histogramme, est égale à 0 concernant l'attribut 1 (Figure 2.31.a), ce qui permet de détecter une scission à $t=53$. Aucune nouvelle forme conservée dans le bloc d'évolution n'appartient à ces barres. La figure 2.31 montre les mesures de représentativité pour la scission d'une classe après adaptation des histogrammes.

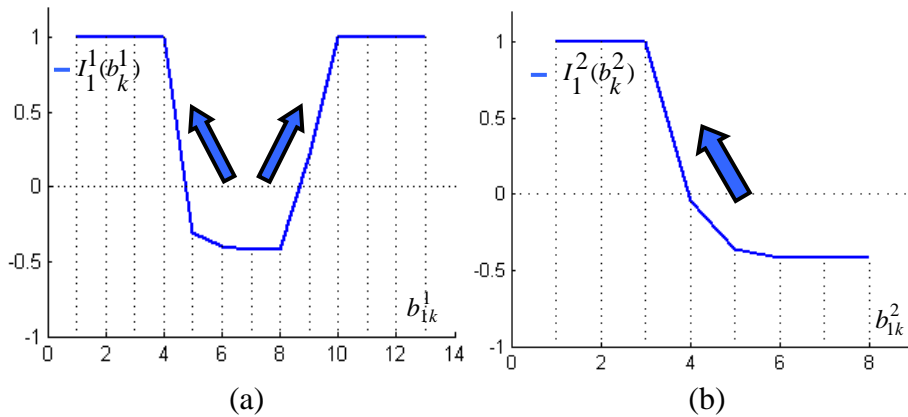


Figure 2.31. Valeurs de représentativité pour les barres des histogrammes de l'attribut 1 (a) et de l'attribut 2 (b) pour la scission de classe.

La figure 2.31 permet de voir les barres avec des valeurs de représentativité négative au centre de la classe par rapport à l'attribut 1. Les valeurs de possibilité de certaines de ces barres sont égales à 0 après adaptation (Figure 2.32.a), contrairement au cas de la rotation d'une classe. Concernant l'attribut 2, un déplacement a été détecté.

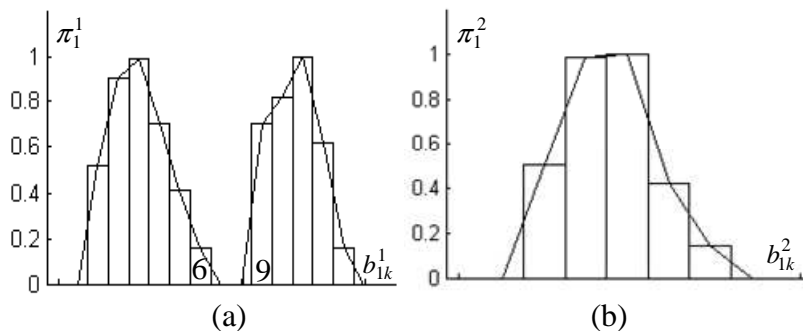


Figure. 2.32. Histogrammes de possibilité pour la classe C_1 en fonction de l'attribut 1 (a) et de l'attribut 2 (b) pour le cas de scission d'une classe.

La figure 2.32 permet de voir pour l'attribut 1 que deux barres (barre 7 et barre 8) positionnées au centre de l'histogramme de la classe, ont des valeurs de possibilité égales à 0. Cela indique qu'aucune nouvelle forme conservée dans le bloc d'évolution n'appartient à ces deux barres. La figure 2.33 permet de voir le résultat de classification obtenu par FPMD après adaptation du classifieur.

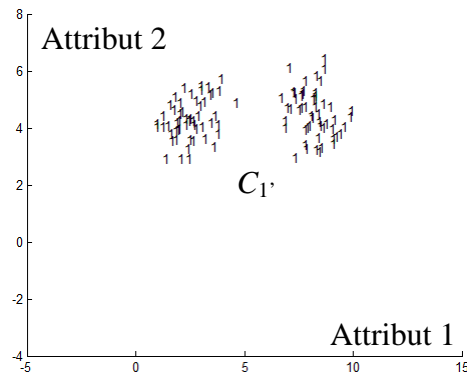


Figure 2.33. Résultat de classification obtenu par FPMD après adaptation de la classe ayant subi une scission.

La scission de la classe a été détectée par FPMD mais le nombre de classes résultantes et leur partitionnement n'a pas encore été déterminé. Ainsi, la méthode FCM a été appliquée avec l'indice de validité défini en (2.15), pour trouver le nombre optimal de classes. Le tableau 2.9 donne la mesure de validité obtenue pour plusieurs nombres de classes.

Tableau 2.9. Mesures de validité pour la classe obtenue après scission.

Nombre de classes	Mesure de validité correspondante
2	0,03
3	0,34
4	0,21
5	0,33

La valeur minimale est obtenue pour $c=2$ ce qui indique que le meilleur nombre de classes est deux. La figure 2.34 montre le résultat de classification obtenu par FCM avec 2 classes.

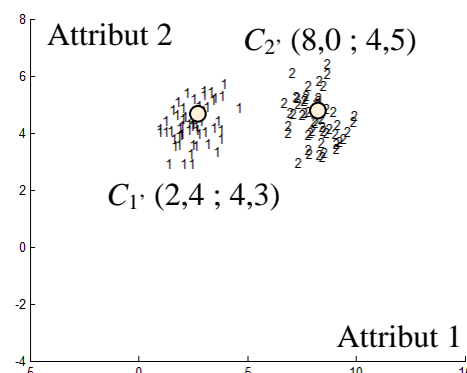


Figure 2.34. Résultat de classification obtenu par FCM pour la classe de scission.

Chacune de ces deux classes suit ensuite un déplacement. Le résultat final de classification des nouvelles formes par FPMD est présenté sur la figure 2.35.

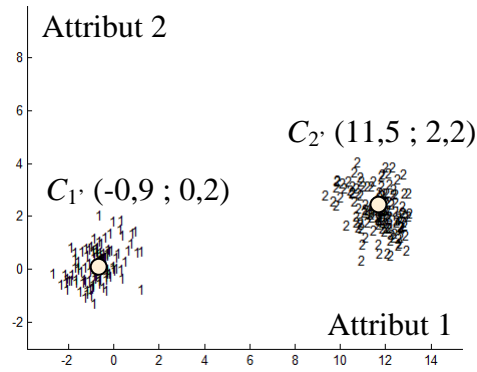


Figure 2.35. Résultat final de classification obtenu par FPMD pour le cas de la scission d'une classe.

Les deux classes présentées sur la figure 2.35 montrent que FPMD a permis de suivre l'évolution des deux classes obtenues. Les valeurs des moyennes de ces deux classes finales sont proches de celles souhaitées, cependant une partie des formes sont transitoires. Les résultats de classification de FPMD pour ce cas sont résumés dans le Tableau 2.10. Le temps maximal de classification d'une forme est égal à 9×10^{-2} s (FCM compris) et le temps moyen de classification est égal à 3×10^{-3} s.

Tableau 2.10. Résultats de classification obtenus par FPMD ($h=6$, $th1=0,1$ et $th2=0,75$), pour le cas de scission d'une classe (Figure 2.15).

	FPMD
$err_{classif}$	0 %
ind_U	100 %
t_{retard}	Pas de retard

Déplacements de deux classes menant à un chevauchement partiel des classes :

Les valeurs de résidu obtenues par FPMD après la classification de chacune des formes de ce cas sont représentées sur la figure 2.36.

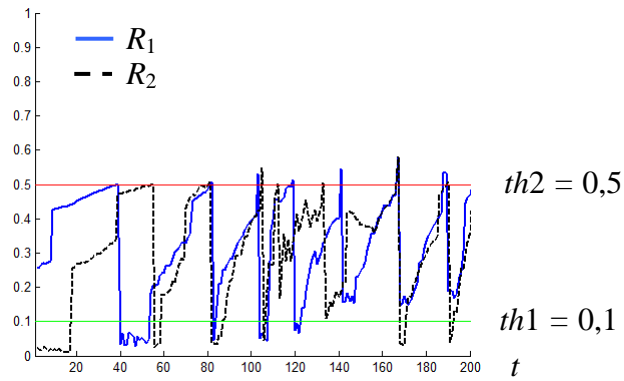


Figure 2.36. Résidu obtenu par FPMD pour le déplacement de deux classes.

En se basant sur la figure 2.36, on voit que plusieurs évolutions de classes ont été détectées et confirmées par le résidu de FPMD. Les mesures de représentativité de ces classes ne sont pas représentées compte tenu du nombre d'adaptations réalisées. Après chacune de ces évolutions, les paramètres des classes ont été mis à jour. Les classes finales évoluées et adaptées obtenues par FPMD sont présentées sur la figure 2.37 et les résultats de classification sont détaillés dans le tableau 2.11.

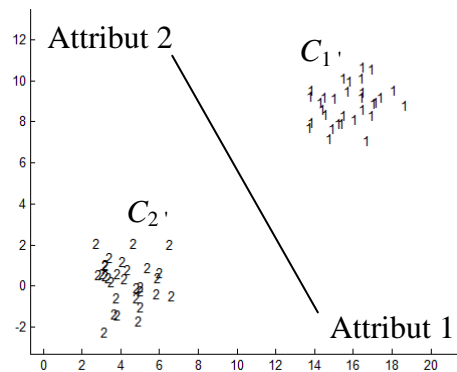


Figure 2.37. Résultat final de classification obtenu par FPMD. La frontière de décision obtenue par FPMD est représentée par '-'.

Tableau 2.11. Résultats de classification obtenus par FPMD ($h=3$, $th1=0,1$ et $th2=0,5$).

	FPMD
$err_{classif}$	0 %
ind_U	100 %
t_{retard}	Pas de retard

L'utilisation des formes récentes et représentatives a permis de bien estimer les fonctions d'appartenance des classes et ainsi de réaliser une bonne classification des formes au fur et à mesure de l'évolution de la classe. Le temps maximal de classification d'une forme est égal à

2×10^{-2} s et le temps moyen de classification est égal à 4×10^{-3} s. La frontière de décision de la figure 2.37 montre que la prise en compte de l'évolution des classes a bien été réalisée et qu'elles sont bien séparées.

Les cas de simulations d'évolution de classes qui ont été présentés, permettent de voir que la méthode FPMD permet de conserver les seules formes représentatives d'une évolution de classe. FPMD a montré ses capacités à suivre les évolutions des classes dynamiques, mais quelques formes transitoires représentatives des évolutions peuvent quand même être conservées dans les classes adaptées. Les seuils de détection et de confirmation d'une évolution, respectivement $th1$ et $th2$ doivent être bien déterminés dans le but de détecter les évolutions de la classe et de bien adapter leurs fonctions d'appartenance. Toutefois, les valeurs définies par défaut pour ces seuils permettent généralement de détecter les évolutions de classes et de les adapter. Ces seuils influenceront le retard de détection des évolutions ainsi que le nombre de formes représentatives utilisées pour la mise à jour d'une classe.

Dans la partie suivante, les résultats de classification de la méthode KPPVFD-S sont présentés sur les mêmes exemples d'évolutions de classes dynamiques.

2.3.3. Application de la méthode KPPVFD-S sur des classes dynamiques

KPPVFD-S a été testée sur les différentes bases de données présentées dans la partie 2.3.1. Les valeurs des paramètres définis pour l'utilisation de la méthode sont celles permettant d'obtenir les meilleures performances.

Classe en déplacement :

La méthode KPPVFD Supervisée a été évaluée sur le déplacement d'une classe généré suivant les équations (2.24). La classe qui évolue est adaptée deux fois par la méthode pour arriver finalement au résultat présenté sur la figure 2.38.

- La première adaptation de la classe a lieu à $t=64$,
- La seconde adaptation de la classe a lieu à $t=193$.

Puis les formes suivantes sont classées dans la classe sans qu'une nouvelle évolution suffisamment importante soit détectée.

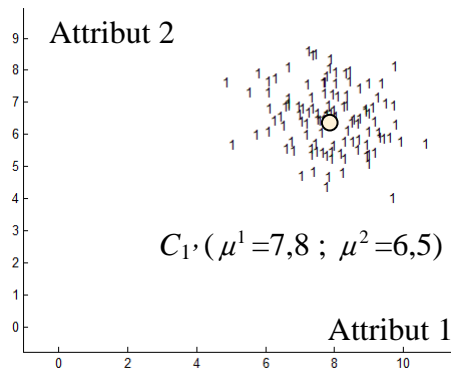


Figure 2.38. Résultat de classification obtenu par KPPVFD-S après classification de toutes les formes du cas du déplacement.

Le classifieur a réalisé le suivi de la classe en utilisant seulement les formes représentatives de l'évolution. Une fois l'évolution terminée, les nouvelles formes sont classifiées normalement. Très peu de formes transitoires ont été conservées. Les résultats de classification de KPPVFD-S sont résumés pour le cas du déplacement dans le Tableau 2.12.

Tableau 2.12. Résultats de classification obtenus par KPPVFD Supervisée ($k=5$; $th1=5$; $NbMin=10$; $th_{Fusion}=0,2$; $n_1=10$ et $n_2=20$) pour le cas de déplacement d'une classe (Figure 2.9).

	<i>KPPVFD Supervisée</i>
ind_U	100 %
$err_{estimation}$	1,3 %
t_{retard}	4 formes

Le temps maximal de classification d'une forme est égal à 2×10^{-2} s et le temps moyen de classification est égal à 2×10^{-3} s.

Rotation de classe :

KPPVFD-S a également été testée sur le cas de la rotation d'une classe. La figure 2.39 permet de voir le résultat de classification obtenu par la méthode une fois que la dernière adaptation de la classe a été réalisée et que les formes suivant l'adaptation ont été classifiées.

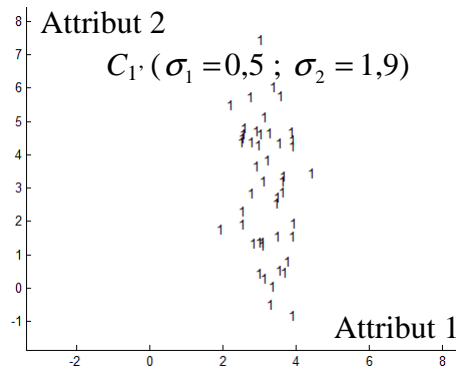


Figure 2.39. Résultat de classification obtenu par KPPVFD Supervisée.

Pour ce cas, une seule adaptation a eu lieu à $t=136$. Puis les formes restantes ont été classées dans la classe adaptée. L'ensemble des résultats de classification obtenu par la méthode est présenté dans le Tableau 2.13. Le temps maximal de classification d'une forme est égal à 3×10^{-2} s et le temps moyen de classification est égal à 2×10^{-3} s.

Tableau 2.13. Résultats de classification obtenus par KPPVFD Supervisée ($k=7$; $th1=4$; $NbMin=7$; $th_{Fusion}=0,05$; $n_1=10$ et $n_2=20$) pour le cas de rotation d'une classe (Figure 2.11).

	KPPVFD Supervisée
ind_U	100 %
$err_{estimation}$	3,3 %
t_{retard}	Pas de retard

Fusion de deux classes :

Le cas de la fusion de deux classes est présenté dans cette partie en utilisant la méthode KPPVFD Supervisée. Plusieurs adaptations des deux classes ont eu lieu ce qui signifie que l'évolution de chaque classe a été détectée. Puis, petit à petit les classes adaptées se sont suffisamment chevauchées pour devoir être fusionnées. La figure 2.40.a représente les classes trouvées avant leur fusion et sur la figure 2.40.b le résultat de la fusion des classes et de la classification de l'ensemble des formes est présenté.

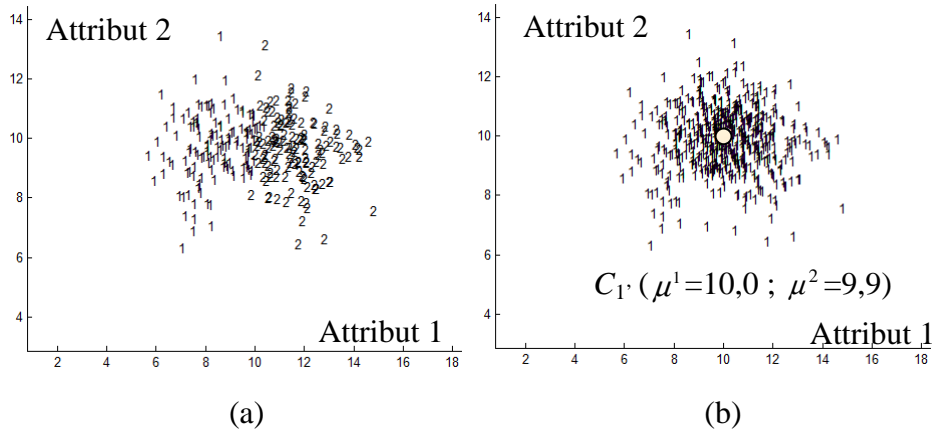


Figure 2.40. Résultat de classification obtenu par KPPVFD Supervisée avant fusion (a) et après fusion et classification de l'ensemble des formes (b).

Les résultats de classification obtenus par la méthode sont présentés dans le tableau 2.14. Le temps maximal de classification d'une forme est égal à 9×10^{-2} s et le temps moyen de classification est égal à 2×10^{-3} s.

Tableau 2.14. Résultats de classification obtenus par KPPVFD Supervisée ($k=5$; $th1=5$; $NbMin=10$; $th_{Fusion}=0,2$; $n_1=10$ et $n_2=20$) pour le cas de fusion de deux classes (Figure 2.13).

	<i>KPPVFD Supervisée</i>
ind_U	100 %
$err_{estimation}$	2,0 %
t_{retard}	7 formes

Scission de classe :

La méthode KPPVFD-S a été utilisée pour le cas de scission d'une classe dynamique également. Plusieurs adaptations de la classe ont eu lieu pour ce cas. Suite à la première adaptation de la classe à $t=8$, une scission a été détectée par FCM pour un nombre de classe optimal trouvé égal à 2. Seulement en se basant sur le modèle de FCM à ce moment, trop peu de formes (moins de k) sont présentes dans les 2 classes estimées. Il est donc nécessaire d'attendre que k nouveaux points apparaissent afin de vérifier si la scission est toujours nécessaire. A $t=18$, la classe adaptée obtenue par KPPVFD-S est celle de la figure 2.41.

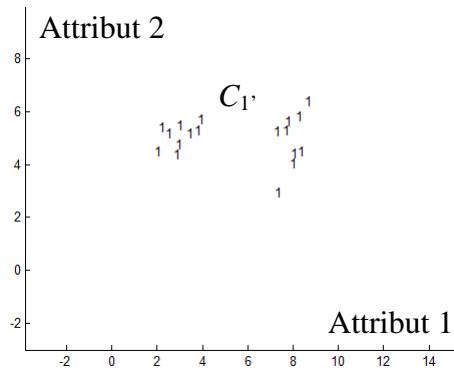


Figure 2.41. Résultat de classification obtenu par KPPVFD Supervisée à $t=18$.

A $t=18$, la classe est de nouveau étudiée par FCM et les mesures de validité correspondant à plusieurs nombres de classes sont données dans le tableau 2.15.

Tableau 2.15. Mesures de validité pour la classe obtenue à $t=18$.

Nombre de classes	Mesure de validité correspondante
2	0,03
3	0,18
4	0,20
5	0,40

La valeur minimale obtenue dans le tableau 2.15 pour $c=2$ indique que le meilleur nombre de classe trouvé est deux. La figure 2.42 montre le résultat de classification obtenu par FCM avec 2 classes.

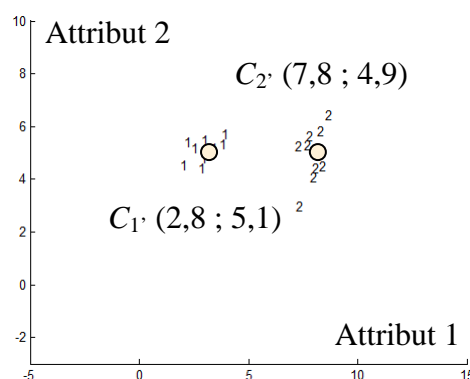


Figure 2.42. Résultat de classification obtenu par FCM pour le cas de scission.

La scission d'une classe en deux classes a été détectée. Ainsi, deux classes sont obtenues et le reste de la classification peut avoir lieu. Chacune des deux classes suit un déplacement et

est adaptée plusieurs fois pour obtenir le résultat final de classification par KPPVFD-S présenté sur la figure 2.43.

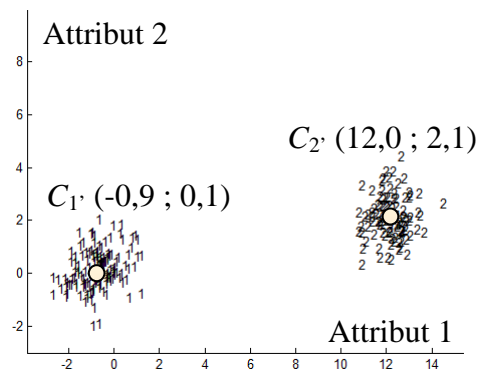


Figure 2.43. Résultat final de classification obtenu par KPPVFD-S pour le cas de la scission d'une classe.

Les deux classes présentées sur la figure 2.43 montrent que KPPVFD-S a détecté les évolutions des classes et qu'avec son mécanisme de scission, deux classes ont été obtenues. Les résultats de classification de la méthode KPPVFD-S pour ce cas sont résumés dans le Tableau 2.16. Le temps maximal de classification d'une forme est égal à 9×10^{-2} s (FCM compris) et le temps moyen de classification est égal à 2×10^{-3} s.

Tableau 2.16. Résultats de classification obtenus par KPPVFD Supervisée ($k=9$; $th1=5$; $NbMin=15$; $th_{Fusion}=0,2$; $n_1=10$ et $n_2=20$) pour le cas de scission d'une classe (Figure 2.15).

	KPPVFD Supervisée
$err_{classif}$	0 %
ind_U	100 %
t_{retard}	Pas de retard

Déplacements de deux classes menant à un chevauchement partiel des classes :

Ce dernier cas d'illustration présente le déplacement simultané de deux classes dans une direction opposée. Cette évolution de classe présente un cas où les nouvelles classes dynamiques évoluées vont chevauchées les classes initialement apprises. La figure 2.44 et le tableau 2.17 montrent les résultats de classification obtenus par la méthode KPPVFD Supervisée pour cette base de données.

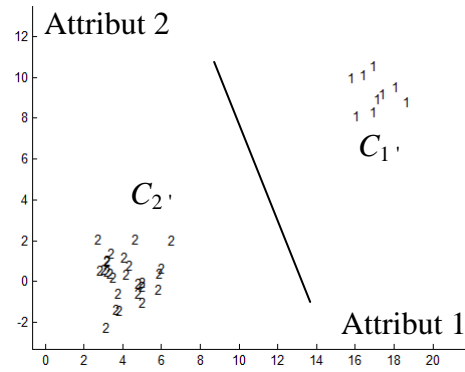


Figure 2.44. Résultat final de classification obtenu par KPPVFD Supervisée. La frontière de décision obtenue par KPPVFD-S est représentée par '-'.

Tableau 2.17. Résultats de classification obtenus par KPPVFD Supervisée ($k=5$; $th1=5$; $NbMin=10$; $th_{Fusion}=0,2$; $n_1=10$ et $n_2=20$) pour le cas de déplacements de deux classes menant à un chevauchement (Figure 2.17).

	KPPVFD Supervisée
$err_{classif}$	0 %
ind_U	100 %
t_{retard}	5 formes

Chaque classe a été adaptée plusieurs fois au cours du temps. De cette façon les transitions successives des modes de fonctionnement ont pu être mises à jour jusqu'à arriver à la localisation finale de chacune des classes. Grâce à ces multiples adaptations, le suivi des classes a pu être réalisé, aucune erreur de classification n'a eu lieu et seules les formes représentatives des modes de fonctionnement ont été conservées. La frontière de décision de la figure 2.44 montre que les classes obtenues sont bien séparées et que leurs modèles sont bien adaptés. Le temps maximal de classification d'une forme est égal à 4×10^{-2} s et le temps moyen de classification est égal à 3×10^{-3} s.

Dans la partie suivante, la méthode KPPVFD Semi-Supervisée est également appliquée sur les mêmes exemples de classes dynamiques.

2.3.4. Application de la méthode KPPVFD-SS sur des classes dynamiques

Les résultats obtenus avec la méthode KPPVFD-SS sur les bases de données présentées dans la partie 2.3.1 sont détaillés dans cette partie. Les valeurs des paramètres définis pour l'utilisation de la méthode sont celles permettant d'obtenir les meilleures performances. Les numéros des classes montrés par KPPVFD-SS ne correspondent pas forcément à l'ordre de

leur apparition. En effet, avec les procédures de fusion et de suppression des classes, les numéros des classes évoluent en permanence.

Classe en déplacement :

Lors de la classification des formes évoluant dans le cadre de la classe se déplaçant (équation 2.24), plusieurs classes ont été créées. Puis, certaines d'entre elles ont été fusionnées et d'autres supprimées. Le résultat final de classification obtenu par la méthode KPPVFD Semi-Supervisée est présenté sur la figure 2.45.

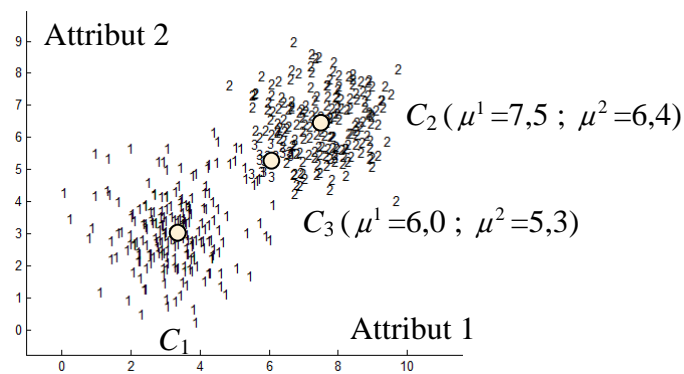


Figure 2.45. Résultat de classification obtenu par KPPVFD-SS après classification de toutes les formes du cas du déplacement.

La méthode a réussi à détecter l'évolution de la classe. Une partie des formes transitoires n'a pas encore été éliminée, mais une partie l'a déjà été, en effet sur les 450 formes de l'espace de représentation, il en reste 415. La classe initiale C_1 a conservé ses caractéristiques et la classe C_2 correspond bien à la classe attendue. Les résultats de classification de KPPVFD-SS sont résumés pour le cas du déplacement dans le Tableau 2.18.

Tableau 2.18. Résultats de classification obtenus par KPPVFD Semi-Supervisée ($k=5$; $th_1=5$; $NbMin=5$; $th_{Fusion}=0,2$; $n_1=10$ et $n_2=20$) pour le déplacement d'une classe (Figure 2.9).

	KPPVFD Semi-Supervisée
ind_U	100 %
$err_{estimation}$	2,6 %
t_{retard}	4 formes

Le temps maximal de classification d'une forme est égal à 5×10^{-2} s et le temps moyen de classification est égal à 5×10^{-3} s.

Rotation de classe :

KPPVFD Semi-Supervisée montre ses résultats de classification pour le cas de la rotation d'une classe sur la figure 2.46.

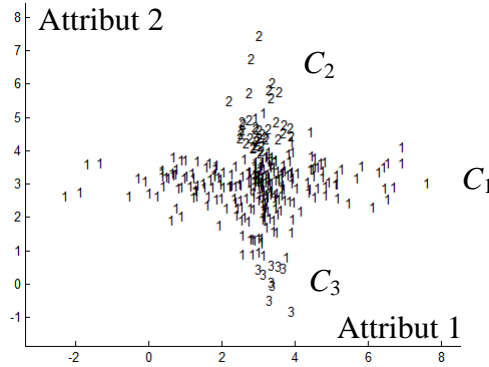


Figure 2.46. Résultat de classification obtenu par KPPVFD Semi-Supervisée pour la rotation.

Seules les nouvelles formes ont été utilisées pour la création des classes. Les classes C_2 et C_3 montrent bien que la classe a évolué, cependant puisque C_1 a été conservée et que ses caractéristiques n'ont pas beaucoup changées, les classes C_2 et C_3 sont séparées. Le temps maximal de classification d'une forme est égal à 5×10^{-2} s et le temps moyen de classification est égal à 3×10^{-3} s.

Tableau 2.19. Résultats de classification obtenus par KPPVFD Semi-Supervisée ($k=7$; $th1=4$; $NbMin=7$; $th_{Fusion}=0,05$; $n_1=10$ et $n_2=20$) pour le cas de la rotation d'une classe (Figure 2.11).

	KPPVFD Semi-Supervisée
ind_U	100 %
t_{retard}	Pas de retard

La valeur d' $err_{estimation}$ n'est pas donnée puisque deux classes sont obtenues au lieu de celle définie (Figure 2.11). Il est donc difficile de faire un comparatif des résultats de ce cas pour cette méthode et ce critère.

Fusion de deux classes :

Les résultats de classification de KPPVFD Semi-Supervisée pour le cas de fusion sont présentés sur la figure 2.47 et dans le tableau 2.20.

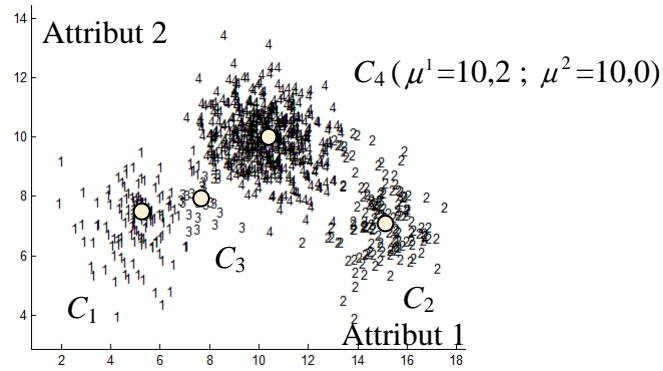


Figure 2.47. Résultat de classification obtenu par KPPVFD-SS.

Plusieurs classes ont été créées par la méthode dynamique. Une classe de fusion C_4 a été créée et elle contient les formes présentant des caractéristiques similaires venant de classes initialement différentes. Cette classe C_4 permet de bien estimer les caractéristiques finales obtenues. Le temps maximal de classification d'une forme est égal à 3×10^{-2} s et le temps moyen de classification est égal à 6×10^{-3} s.

Tableau 2.20. Résultats de classification obtenus par KPPVFD Semi-Supervisée ($k=5$; $th1=5$; $NbMin=5$; $th_{Fusion}=0,1$; $n_1=10$ et $n_2=20$) pour le cas de fusion de deux classes (Figure 2.13).

	KPPVFD Semi-Supervisée
ind_U	100 %
$err_{estimation}$	2,0 %
t_{retard}	7 formes

Scission de classe :

KPPVFD Semi-Supervisée peut traiter les cas de scission de part sa faculté à créer de nouvelles classes dès qu'une évolution est détectée. Dans ces conditions, qu'une ou plusieurs évolutions aient lieu ne change pas le fait qu'elles seront détectées par la méthode et qu'un certain nombre de classes sera alors créé. La figure 2.48 présente le résultat de classification obtenu par la méthode à la fin de l'apparition de toutes les nouvelles formes.

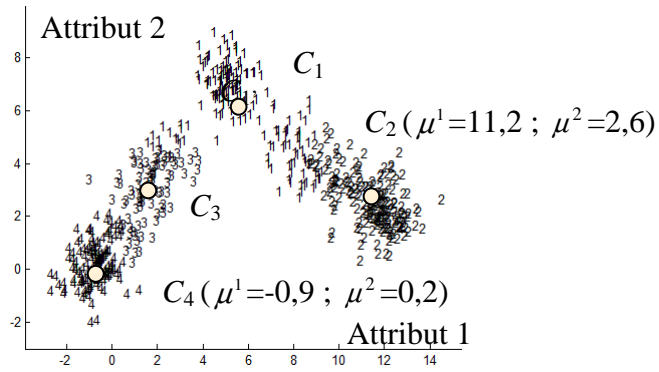


Figure 2.48. Résultat de classification obtenu par KPPVFD Semi-Supervisée.

Quatre classes sont obtenues à la fin de la classification de toutes les formes. Les classes C_2 et C_4 représentent les classes finales d'évolution. Leurs caractéristiques correspondent bien à celles des classes attendues. Les résultats de classification de la méthode KPPVFD Semi-Supervisée pour ce cas sont résumés dans le Tableau 2.21. Le temps maximal de classification d'une forme est égal à 6×10^{-2} s et le temps moyen de classification est égal à 5×10^{-3} s.

Tableau 2.21. Résultats de classification obtenus par KPPVFD Semi-Supervisée ($k=5$; $th1=5$; $NbMin=7$; $th_{Fusion}=0,1$; $n_1=10$ et $n_2=20$) pour la scission d'une classe (Figure 2.15).

	KPPVFD Semi-Supervisée
$err_{classif}$	4 %
ind_U	100 %
t_{retard}	12 formes

Déplacements de deux classes menant à un chevauchement partiel des classes :

Les résultats de KPPVFD Semi-Supervisée pour ce dernier cas dynamique de déplacements de deux classes sont présentés sur la figure 2.49 et dans le tableau 2.22. Les deux classes évoluent simultanément jusqu'à chevaucher la classe initialement opposée.

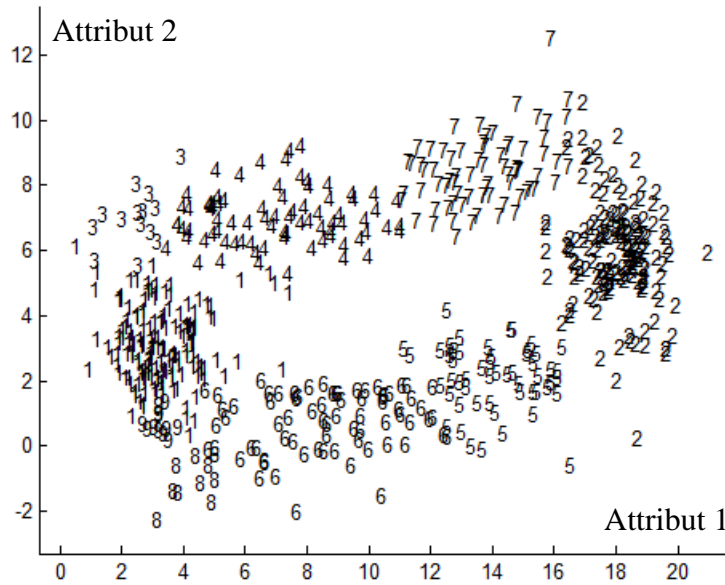


Figure 2.49. Résultat final de classification obtenu par KPPVFD Semi-Supervisée.

Chaque classe a été adaptée plusieurs fois au cours du temps. Plusieurs classes ont été obtenues, ont fusionnées et ont été supprimées. Ainsi, les classes C_1 , C_3 , C_4 et C_7 permettent de suivre l'évolution de la classe C_1 initiale. De la même façon, les classes C_2 , C_5 , C_6 , C_8 et C_9 montrent l'évolution de la classe C_2 initiale. Ce suivi permet de réaliser une classification des formes sans faute et de tenir compte de l'ensemble des nouvelles et des anciennes formes représentatives. De plus, il serait possible d'utiliser l'ensemble de ces classes d'évolution pour prédire la classe vers laquelle on se dirige au fur et à mesure du temps. Le temps maximal de classification d'une forme est égal à 3×10^{-2} s et le temps moyen de classification est égal à 9×10^{-3} s.

Tableau 2.22. Résultats de classification obtenus par KPPVFD Semi-Supervisée ($k=5$; $th1=5$; $NbMin=10$; $th_{Fusion}=0,2$; $n_1=10$ et $n_2=20$).

	<i>KPPVFD Semi-Supervisée</i>
$err_{classif}$	4 %
ind_U	100 %
t_{retard}	5 formes

2.3.5. Comparatif des résultats obtenus

Dans cette partie les résultats obtenus par l'ensemble des méthodes sont regroupés dans le tableau 2.23 afin de pouvoir les comparer.

Tableau 2.23. Récapitulatif des résultats de classification obtenus pour les évolutions des classes dynamiques présentées.

		<i>FPMI</i>	<i>SVMI</i>	<i>KPPVI</i>	<i>FPMD</i>	<i>KPPVFD-S</i>	<i>KPPVFD-SS</i>
Déplacement	<i>ind_U</i>	66,6 %	66,6 %	66,6 %	100 %	100 %	100 %
	<i>erreur_{estimation}</i>	29,9 %	29,9 %	29,9 %	2,6 % 5,2 %	1,3 %	2,6 %
	<i>t_{retard}</i>	Aucune détection	Aucune détection	Aucune détection	0	4	4
Rotation	<i>ind_U</i>	50 %	50 %	50 %	100 %	100 %	100 %
	<i>erreur_{estimation}</i>	3,3 %	3,3 %	3,3 %	3,3 %	3,3 %	-
	<i>t_{retard}</i>	Aucune détection	Aucune détection	Aucune détection	0	0	0
Fusion	<i>ind_U</i>	68,8 %	68,8 %	68,8 %	100 %	100 %	100 %
	<i>erreur_{estimation}</i>	10,9 %	10,9 %	10,9 %	3,0 %	2,0 %	2,0 %
	<i>t_{retard}</i>	273	273	273	0	7	7
Scission	<i>err_{classif}</i>	50 %	50 %	50 %	0 %	0 %	4 %
	<i>ind_U</i>	71,4 %	71,4 %	71,4 %	100 %	100 %	100 %
	<i>t_{retard}</i>	Aucune détection	Aucune détection	Aucune détection	0	0	12
Déplacement de deux classes	<i>err_{classif}</i>	43,8 %	50,5 %	10,5 %	0 %	0 %	4 %
	<i>ind_U</i>	57,1 %	57,1 %	57,1 %	100 %	100 %	100 %
	<i>t_{retard}</i>	Aucune détection	Aucune détection	Aucune détection	0	5	5

Comme on peut le voir dans le tableau 2.23, les méthodes de classification dynamique proposées donnent de bien meilleurs résultats que les méthodes incrémentales concernant les classes dynamiques. Chacune des trois méthodes proposées donne des résultats satisfaisants et puisqu'elles possèdent chacune des caractéristiques qui leurs sont propres (nombre de paramètres à définir, histogrammes de possibilité, distances entre points, etc.), n'importe quelle méthode peut être sélectionnée pour réaliser le suivi d'un système évolutif.

Dans le cas des classes de formes complexes (non-convexes) KPPVFD sera généralement meilleure puisqu'elle tient compte de la forme non-convexe des classes et de la non-linéarité entre les attributs des classes. Sur ces exemples illustratifs de classes gaussiennes, on peut dire que :

- FPMD a réussi à détecter les évolutions des classes sans retard de détection. La méthode a conservé plus de formes représentatives que KPPVFD-S, ce qui lui permet de conforter l'estimation du modèle de ses classes mais qui augmente l'erreur d'estimation obtenue pour certains cas,
- KPPVFD-S a permis d'obtenir l'erreur d'estimation la plus faible pour certains cas, puisque le nombre de formes représentatives conservées par la méthode était plus faible. Son temps de retard de détection des évolutions de classe est faible par rapport à chaque cas présenté. Par contre son temps de classification des nouvelles formes est généralement supérieur à celui de FPMD,
- KPPVFD-SS a détecté l'apparition de nouvelles classes, tout en détectant les évolutions successives des classes. Toutefois la méthode a un pourcentage d'erreur de classification supérieur à FPMD et KPPVFD-S pour certains des cas présentés.

2.4. Conclusion

Dans ce chapitre, trois méthodes de reconnaissance dynamique des formes ont été proposées. Ces méthodes permettent de traiter les différentes évolutions que des classes dynamiques peuvent rencontrer (déplacement, scission, fusion, etc.). Chaque méthode possède ses paramètres propres qui influencent les résultats de classification. Toutefois, plusieurs conseils permettant de guider la définition de ces paramètres ont été donnés. De plus, plusieurs exemples de simulations ont été présentés afin d'illustrer les méthodes proposées. A titre de comparaison, trois méthodes incrémentales de la littérature ont été également testées sur ces exemples afin de montrer l'intérêt des méthodes de classification dynamique. Ces méthodes incrémentales ont été utilisées puisqu'elles intègrent les nouvelles données mais qu'elles ne tiennent pas compte de la représentativité des points au cours du temps. Ainsi, il est possible de voir l'intérêt et l'apport des méthodes de classification dynamique.

Deux des trois méthodes proposées sont des classifieurs supervisés. Elles permettent de suivre les évolutions progressives et brutales mais ne détectent pas les nouvelles classes. Par contre, le fait d'oublier les données devenues obsolètes leur permet de mettre à jour le classifieur et de coller au mieux aux changements des paramètres des modes de fonctionnement du système au cours du temps. La troisième méthode est un classifieur semi-supervisé, elle permet de suivre tout type d'évolution et de détecter l'apparition de nouvelles classes ou sous-classes. Ces nouvelles classes correspondent aux nouveaux modes de

fonctionnement du système qui peuvent être des modes de fonctionnement défailants ou transitoires. La détection des modes transitoires est utile et intéressante puisqu'elle permet de réduire le temps de retard de détection d'une évolution.

Les trois méthodes, développées dans ce chapitre, tiennent compte de l'aspect évolutif du système de manières différentes (fonctions d'appartenance, indicateurs d'évolution, validation des évolutions, adaptation des classes, etc.). Elles ont pour objectif d'aider les opérateurs humains dans l'exploitation, le diagnostic et le suivi des modes de fonctionnement des systèmes évolutifs et non pas de les remplacer. Ce sont des méthodes applicables facilement, avec une information limitée sur le fonctionnement d'un système. Les temps de classification de ces méthodes et leurs résultats ont montrés qu'elles sont applicables en ligne sans générer un retard de détection important. Dans le chapitre 4, ces méthodes seront évaluées à l'aide de plusieurs exemples de simulations et d'applications réelles.

Le chapitre suivant répond au problème de classification des formes dynamiques. Il présente deux approches proposées pour résoudre ce problème de classification tout en tenant compte de la structure des informations observées. Une première approche structurelle de classification est proposée pour permettre de classer les formes dynamiques. Puis, une approche mixte, c'est-à-dire mélangeant les propriétés statistiques et structurelles des formes, est développée pour réaliser la classification des données dynamiques tout en permettant d'obtenir un résultat interprétable par les opérateurs humains.

Chapitre 3. Contributions aux méthodes de reconnaissance des formes dynamiques

Dans ce chapitre, nous nous intéressons au problème de classification des formes dynamiques. Une forme dynamique est définie par une trajectoire représentant son développement temporel durant une certaine période. La caractérisation de cette forme est réalisée par la définition de sa structure. La détermination de la classe d'une forme dynamique n'est pas seulement basée sur sa valeur instantanée à un instant donné mais également sur son développement temporel. Cette détermination nécessite donc l'utilisation des méthodes de reconnaissance structurelles des formes qui tiennent compte de la structure d'une forme dynamique. Par ailleurs, la combinaison des propriétés statistiques et structurelles des formes dynamiques donnent naissance à des méthodes mixtes de classification. Les performances de ces méthodes sont supérieures à celles des méthodes structurelles. De plus, elles facilitent l'interprétation des résultats par les opérateurs humains.

Les principes des méthodes structurelle et mixte de reconnaissance des formes dynamiques sont détaillés dans ce chapitre. Notre objectif est de positionner les approches développées dans ce manuscrit pour la classification des formes dynamiques, par rapport à celles existantes de la littérature. Ces approches sont expliquées et illustrées en utilisant plusieurs exemples de simulations. Dans le chapitre 4 (section 4.2.1), les méthodes développées seront évaluées sur le cas de caractérisation et de discrimination de la coordination inter-segmentaire de la marche des patients hémiparétiques de l'hôpital de Garches.

3.1. Introduction

Les méthodes de reconnaissance structurelle des formes (Fu, 1982) (Friedman, 1999) (Bunke, 1990) sont particulièrement adaptées au cas des formes dynamiques. Les formes dynamiques sont composées d'informations historiques concernant l'évolution spatiale et temporelle d'une observation. Cette évolution permet de caractériser la structure et le contexte d'une forme. Décrite par les paramètres les plus informatifs d'une application, une forme dynamique est définie par une trajectoire représentant son développement temporel et son évolution dans le temps peut être infinie. La description retenue peut alors rendre compte des composants élémentaires d'une forme (les primitives), de leurs propriétés temporelles et/ou spatiales (nombre de primitives, orientation, courbure) et de leurs inter-relations (Armand,

2005). La différence notable avec les méthodes statistiques est l'utilisation de l'évolution des observations pour classer une forme dynamique. En effet, contrairement aux méthodes statistiques, la classification d'une forme ne dépend pas uniquement de sa valeur à l'instant courant mais également de son développement temporel (sa trajectoire dans le passé). Cependant, les méthodes structurelles sont particulièrement adaptées aux formes dynamiques dont une structure caractéristique peut être extraite. Dans le cas de signaux bruités ou de signaux dont l'évolution n'est pas progressive mais fortement discontinue, il peut être difficile de distinguer une ou plusieurs allures caractéristiques d'un mode de fonctionnement normal ou anormal. Les méthodes de classification statistiques peuvent alors être utilisées en complément d'une méthode structurelle. Ces méthodes, dites mixtes, seront détaillées dans la section 3.4 de ce chapitre.

Dans le cas où la forme est observée continuellement, une méthode capable de traiter les informations en ligne, i.e. arrivant au fur et à mesure, est recommandée (utilisation d'une fenêtre glissante éventuellement). Dans le cas contraire, si seulement un ensemble d'informations historiques est disponible concernant l'évolution d'un paramètre au cours du temps, il est alors possible d'utiliser une méthode structurelle hors ligne. Dans les deux cas, la méthode analyse les informations disponibles jusqu'à un moment donné, par partie ou de façon globale, puis une représentation simplifiée de la forme dynamique est obtenue par segmentation. Sur la figure 3.1, les six types de modulations principalement utilisés pour caractériser les différentes structures composant une forme dynamique sont présentés.

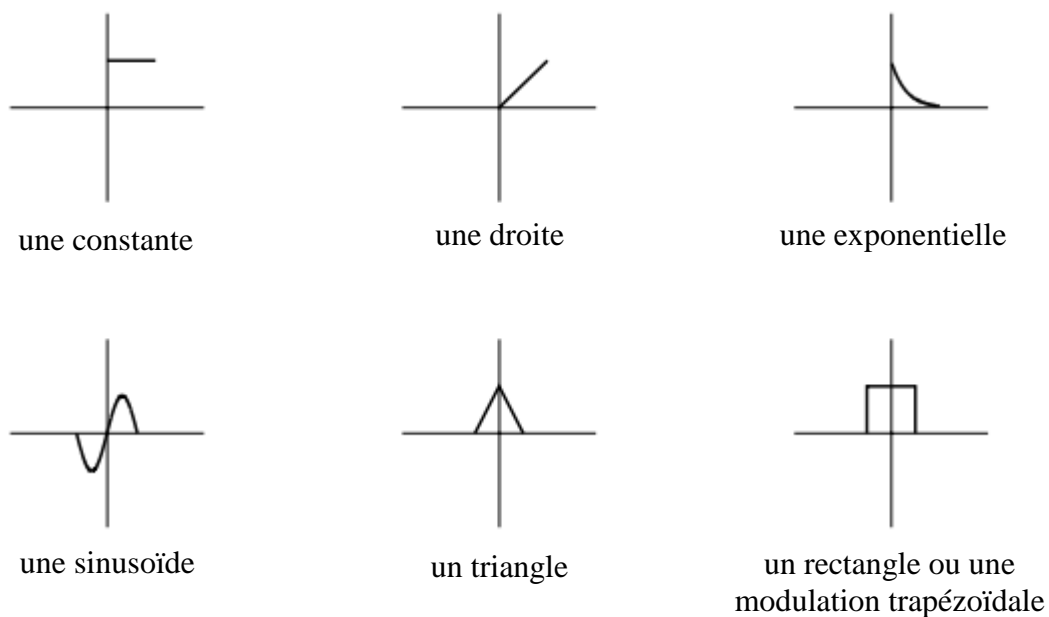


Figure 3.1. Les six principaux types de modulations utilisés pour caractériser une forme dynamique (Olszewsky, 2001).

De plus, comme nous l'avons évoqué au chapitre 1, ces modulations sont également caractéristiques de la pente d'un épisode ou du signe de ses dérivées ; on les appelle des primitives. C'est donc en fonction de l'ensemble des primitives disponibles (d'ordre 1 ou 2), du seuil d'erreur maximal autorisé et de la précision souhaitée que l'on choisit les primitives les plus adaptées à une application. En effet, la représentation structurelle est basée sur la perception humaine (Biederman, 1987). Si la caractérisation d'une forme dynamique devient trop compliquée en utilisant certaines primitives, il peut être préférable de se limiter à quelques types de primitives seulement. Soit on va chercher la primitive la plus adaptée à chaque épisode parmi l'ensemble des primitives connues afin d'obtenir un maximum de précision, soit parmi une liste de primitives sélectionnées on va limiter la précision de la représentation structurelle pour obtenir une représentation simplifiée pouvant être plus facilement interprétée. La représentation structurelle obtenue est ensuite utilisée pour classer la forme dynamique. Plusieurs techniques existent, avec parmi les plus utilisées :

- celles de la théorie des langages formels telles que les automates ou grammaires (Tanaka, 1995) qui utilisent les liaisons et le sens entre les primitives pour caractériser une forme. Pour éviter une trop grande sensibilité aux erreurs dues à l'incomplétude d'une grammaire, une recherche de similarité entre signature décrivant les formes peut être menée en évaluant ce qu'il faut modifier pour qu'elle devienne correcte au sens de la grammaire mise en comparaison. Quand on cherche à discriminer plus de deux classes, une grammaire est nécessaire pour chaque classe,
- les mesures de similarité qui calculent la ressemblance entre formes dynamiques en utilisant les primitives obtenues dans la signature caractérisant chaque forme ou en utilisant des mesures de distances entre formes dynamiques.

Certaines méthodes ne passent pas par la phase de segmentation, elles utilisent la tendance des données connues pour détecter une anomalie (Shewart, etc..). Cependant, les données à utiliser avec ces méthodes doivent avoir un comportement très spécifique en mode de fonctionnement normal mais doivent dériver rapidement suite à l'apparition d'un problème. Lors de la phase de classification, comme pour toute méthode de reconnaissance des formes, il est nécessaire de détecter le plus précisément possible la ou les classes de fonctionnement normal. Pour cela, on commence généralement par analyser les signaux issus d'un processus de fonctionnement normal. Cette analyse faite au cas par cas montre qu'une structure de

données est généralement propre à une application particulière. Par exemple, pour le domaine médical on peut citer les données issues d'Electrocardiogrammes (ECG) (Koski, 1995), les données issues de l'analyse quantifiée de la marche, les données issues du monitoring du rythme cardiaque de patients, etc. Les méthodes structurelles nécessitent donc la connaissance du domaine dans lequel elles vont être appliquées.

Les approches de reconnaissance structurelle des formes dynamiques suivent donc un principe d'extraction des connaissances identique à celui des méthodes de reconnaissance statistique des formes (Fayyad, 1996). Elles commencent par une phase de prétraitement à base de filtres et/ou de transformations, puis l'extraction et la sélection des paramètres sont réalisées pour mieux caractériser le système sur lequel les formes dynamiques sont obtenues. Il est alors possible de réaliser la phase de caractérisation des formes par segmentation. Puis la classification de ces formes (identification des classes et du mode actuel) est réalisée en ligne ou hors ligne selon les contraintes du système.

Suite à l'extraction des connaissances à partir des données, le déroulement d'une méthode de reconnaissance structurelle des formes reproduit le principe de perception humaine en trois étapes (Biederman, 1987) :

- segmentation des objets ou d'une forme dynamique en fonction des différences de pente, luminance, texture, couleur, etc.,
- approximation de chaque partie ou épisode par une primitive,
- identification en fonction de la représentation géométrique globale de la forme dynamique.

Dans la section suivante, plusieurs méthodes de reconnaissance structurelle des formes sont présentées et les deux approches proposées sont expliquées.

3.2. Approches de reconnaissance structurelle des formes dynamiques

Chaque méthode de reconnaissance des formes a ses spécificités, que ce soit dans l'utilisation d'une méthode de segmentation, dans l'utilisation des primitives, dans la phase de classification ou dans l'utilisation de certains paramètres (seuils, etc.). La définition d'une méthode de classification structurelle mène donc à des résultats différents, mais également à des objectifs de représentation d'une forme différents (représentation simplifiée ou complexe). Dans cette partie, nous détaillons deux méthodes structurelles dont les objectifs sont différents.

Pour son approche de classification structurelle, (Olszewski, 2001) utilise un détecteur de structures. Ce détecteur de structures est basé sur l'ensemble des fonctions de modulation définies (constante, droite, exponentielle, etc. cf. figure 3.1). Un nombre d'épisodes est défini initialement, puis il est incrémenté jusqu'à obtenir une erreur d'approximation proche de zéro permettant de considérer qu'il y a suffisamment d'épisodes pour caractériser une forme. L'algorithme récursif proposé permet de détecter les zones minimisant l'erreur d'approximation en fonction de l'ensemble des modulations utilisables. La fonction de modulation retenue est celle qui permet d'obtenir l'erreur d'approximation minimale pour chaque épisode. Avec cette méthode, l'objectif de (Olszewski, 2001) est de caractériser avec le minimum d'erreur possible, la globalité d'une forme dynamique (Figure 3.2).

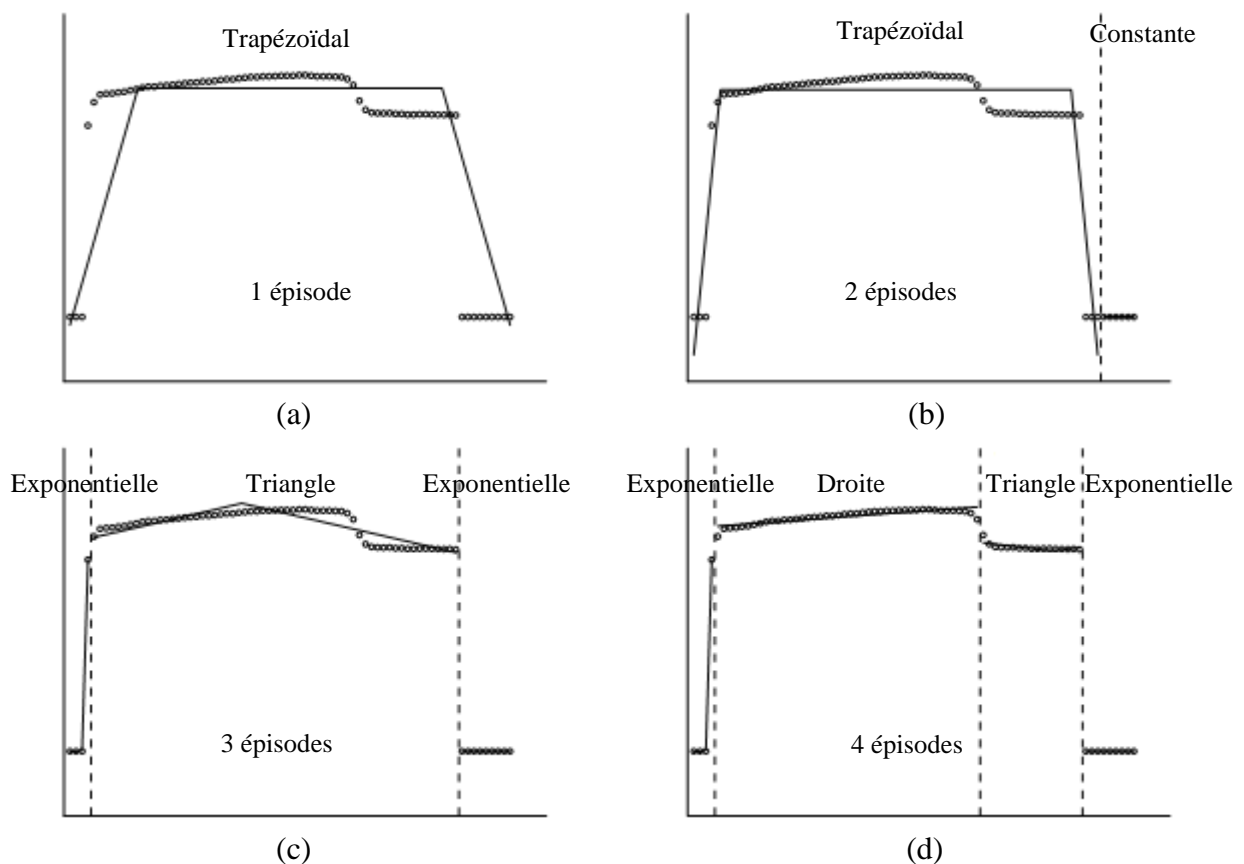


Figure 3.2. Caractérisation d'un signal par un (a), deux (b), trois (c) puis quatre (d) épisodes, en utilisant les modulations les plus adaptées (Olszewski, 2001). Les types de modulation retenus sont indiqués pour chaque épisode.

Avec cette méthode, pour obtenir une erreur d'approximation proche de zéro, il est souvent nécessaire d'utiliser un grand nombre de primitives. Bien sûr, cela est lié à la forme du signal étudié. Une fois un grand nombre de primitives obtenu, chaque forme risque d'avoir une signature spécifique. Le problème de la méthode de (Olszewski, 2001) se pose donc au niveau

de la mesure de similarité entre formes. Pour cette mesure, il est nécessaire de choisir entre comparer le nombre d'épisodes optimal pour une forme dynamique, comparer le type des primitives utilisées pour sa représentation, ou comparer le nombre d'épisodes et le type des primitives de chaque forme dynamique.

Parmi les autres méthodes structurales, nous avons choisi de parler de la méthode du codage symbolique d'une forme dynamique (Daw, 2003) qui conserve l'aspect temporel des données. Le principe de cette méthode est de diviser une forme dynamique en plusieurs parties à l'aide d'un seuil l'amplitude pour obtenir une signature. La figure 3.3 présente un exemple de signature obtenue pour un signal.

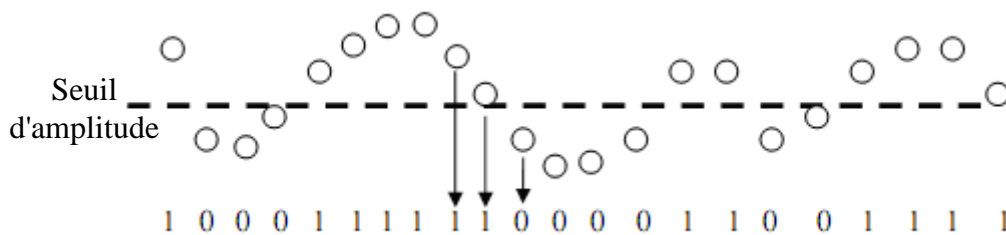


Figure 3.3. Caractérisation d'un signal par codage symbolique.

En fonction de l'amplitude correspondant à chaque valeur du signal, et du seuil fixé, la représentation symbolique diffère beaucoup. Cette méthode très simple utilise généralement un codage binaire. (Daw, 2003) propose ensuite de regrouper dans une fenêtre glissante, trois valeurs binaires obtenues (ex : 1 0 0) et de leur faire correspondre leur valeur en base décimale. Puis, la fenêtre glissante est décalée d'un pas de 1 pour calculer la valeur suivante. De cette façon, la structure de la forme dynamique peut être représentée par huit valeurs différentes (000=>0, 001=>1, ..., 111=>7). Enfin, le nombre de fois où chacune de ses valeurs est apparue, est comptabilisé et présenté dans un histogramme. La similarité entre formes dynamiques est alors réalisée par comparaison de leur histogramme. Le problème de la méthode vient de la définition du ou des seuils d'amplitude et de la perte d'informations engendrée par la signature obtenue. En effet, seule la caractéristique d'amplitude évoluant au cours du temps est conservée.

Notre objectif dans la définition d'une méthode structurale est de proposer une approche conservant une représentation caractéristique des données en fonction de l'évolution temporelle réalisée. Cette méthode a pour but d'être appliquée hors ligne sur les patrons de marche de la coordination inter-segmentaire des hémiparétiques (chapitre 4, section 4.2.1).

C'est en collaboration avec le laboratoire d'Analyse de la marche de l'hôpital de Garches doté d'un système d'analyse de la marche, que nous avons réalisé cette étude. Nous avons alors défini qu'il était intéressant de proposer une méthode tirant compte des informations interprétables des données. Parmi les contraintes de la méthode, il doit être possible d'identifier, quantifier et interpréter la marche d'un patient hémiparétique. Comme nous l'avons vu avec les méthodes présentées ci-dessus ou avec celles du chapitre 1, les méthodes structurelles ont tendance à obtenir une signature approximant le plus possible la forme dynamique étudiée, ou dans le cas de la méthode de (Daw, 2003) d'obtenir une représentation très simplifiée perdant d'importantes caractéristiques structurelles d'une forme pour se concentrer sur ses caractéristiques statistiques (amplitude). C'est pourquoi, nous avons choisi de définir une méthode structurelle, qui ne nécessite pas la définition d'un seuil d'erreur d'approximation et permettant une comparaison simple des signatures. Cette méthode, basée sur la similarité entre primitives obtenues, permet de conserver les informations nécessaires à l'interprétabilité des données. Cette méthode est présentée dans la partie suivante.

3.3. Approche proposée de reconnaissance structurelle des formes dynamiques

Dans cette partie, nous nous intéressons à la caractérisation des formes dynamiques à l'aide d'une approche de reconnaissance structurelle des formes dynamiques. Sur la figure 3.4 un exemple de forme dynamique est présenté. La forme dynamique est divisée en deux phases que l'on souhaite caractériser.

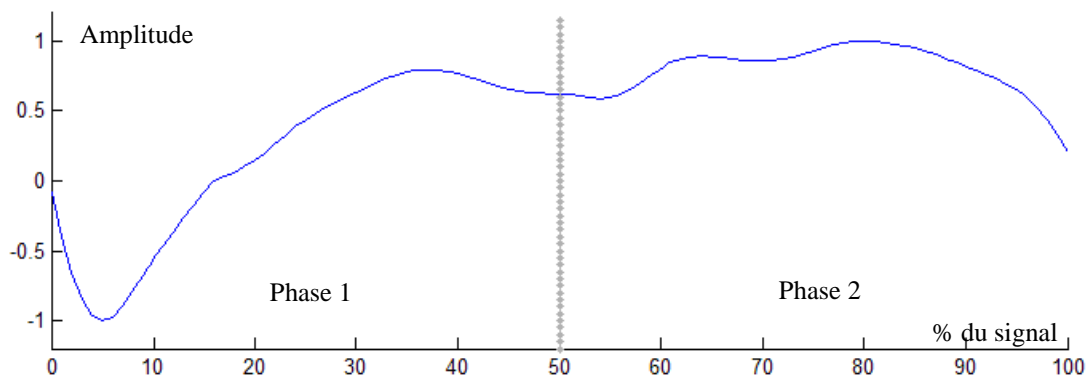


Figure 3.4. Exemple de signal à classifier.

L'approche proposée dans cette partie cherche à caractériser de façon simple les formes dont les caractéristiques structurelles sont similaires ou très proches les unes des autres. Pour cela, elle consiste en :

- l'adaptation du nombre de primitives caractéristiques de chaque phase d'une forme dynamique. Si une forme dynamique ne contient pas de phase caractéristique, l'adaptation du nombre de primitives se fait par rapport à la forme dynamique globale,
- la détermination des points de rupture de chaque phase d'une forme dynamique en utilisant une technique de segmentation. Les primitives sélectionnées doivent représenter tous les épisodes délimités par les points de rupture,
- le rassemblement des signatures obtenues, pour obtenir une signature globale caractéristique,
- la classification de chaque signature globale.

La plupart des méthodes de segmentation utilisent une erreur d'approximation qui est un paramètre difficile à définir. Les signatures obtenues par ces méthodes ont des longueurs généralement différentes, i.e. elles ont un nombre différent de primitives. Il devient alors plus difficile de comparer ces signatures pour obtenir une signature caractéristique, représentative d'une classe. C'est pourquoi, nous avons préféré utiliser un nombre fixe de primitives pour représenter chaque forme dynamique. Plus le nombre de primitives est adapté à une forme dynamique, moins l'erreur d'approximation générée par ces primitives est grande. Il n'est donc plus nécessaire de fixer de seuil d'erreur d'approximation et la comparaison des signatures peut devenir plus interprétable.

Un schéma représentant le principe général de la méthode est présenté sur la figure 3.5.

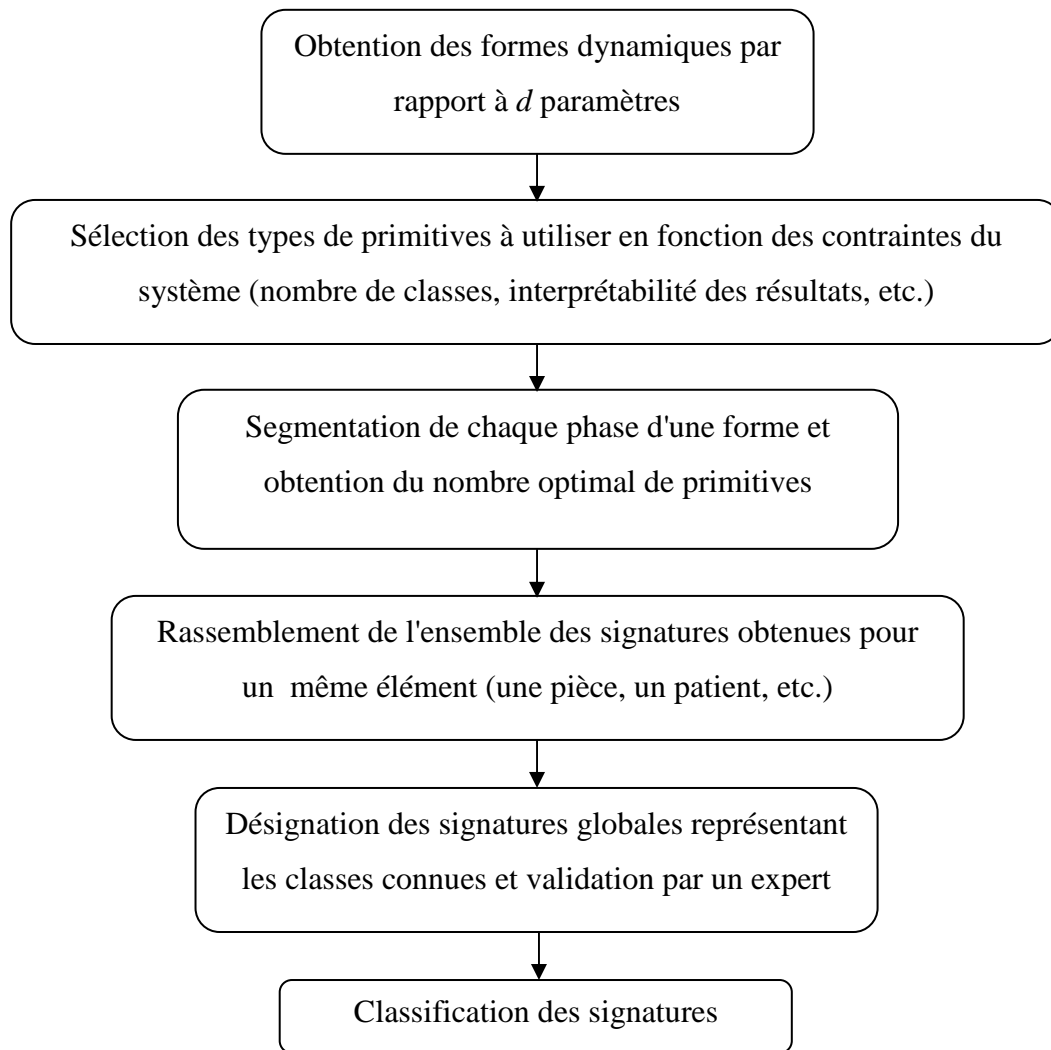


Figure 3.5. Déroulement de l'approche structurelle de reconnaissance des formes proposée.

Dans la partie suivante, la phase d'adaptation du nombre de primitives d'une signature est présentée.

3.3.1. Adaptation du nombre de primitives

Il est important d'adapter le nombre de primitives par rapport à chaque phase que peut contenir une forme dynamique. On peut trouver par exemple pour une même forme, différentes phases de production, des phases du cycle de la marche, etc. En fonction de la longueur de chaque phase d'une forme et de la dynamique de cette phase, le nombre de primitives à utiliser doit être bien défini. Pour cela nous proposons de définir une limite au nombre de primitives qui peuvent être utilisées pour chaque phase d'une forme dynamique. Par exemple, prenons le cas d'un système industriel qui aurait une première phase de conception d'une pièce durant 1mn puis une autre phase durant 20s. D'une part, le nombre

d'observations acquises durant ces deux phases va être complètement différent. D'autre part, en considérant que le système génère plus d'activités dans la première phase, i.e. plus de variations dans le signal, la première phase devra être caractérisée par plus de primitives. Il ne faut donc pas regarder avec la même précision et avec le même nombre de primitives ces deux phases. La première phase pourrait être limitée à une vingtaine de primitives tandis que pour la deuxième phase, quatre ou cinq primitives pourraient suffire. C'est en fonction d'un nombre de primitives insuffisant ou trop important que l'on obtient respectivement une précision d'approximation d'un signal trop faible ou trop grande. Par ailleurs, il est plus facile de caractériser une classe lorsque peu de primitives sont utilisées pour une signature, mais dans ce cas l'inconvénient peut être l'obtention d'une même signature pour des classes différentes.

Afin de trouver le nombre de primitives le plus adapté à chaque phase, nous proposons de définir un nombre maximal N_{max} de primitives défini initialement par expérimentation pour ne pas continuer de façon illimitée la segmentation d'un signal. Une fois ce nombre maximal N_{max} de primitives défini pour chaque phase ou pour un signal complet, en fonction de la longueur des phases, du nombre de classes et de la dynamique du signal, nous utilisons la méthode de segmentation. Pour chacune des phases d'un signal, la méthode de segmentation obtient donc initialement 1 primitive, puis 2, jusqu'à N_{max} primitives par phase si c'est nécessaire. Le nombre de primitives conservées pour représenter chaque phase, est celui qui permet d'obtenir la même signature pour les formes d'une même classe. Ce nombre de primitives sélectionné peut être inférieur ou égal à N_{max} pour chaque phase. Si plusieurs primitives rendent possible cette caractérisation, c'est le nombre maximal de primitives qui est sélectionné pour chaque phase, afin d'accroître la précision de la signature d'une classe et de discriminer au mieux les classes. A l'inverse si aucun nombre de primitives ne permet de bien caractériser une phase, cette phase n'est pas considérée comme représentative de la signature d'une forme et elle n'est pas conservée. Par exemple, cela peut se produire si le problème d'un système n'apparaît jamais dans cette phase. L'obtention des primitives (et donc des épisodes) est réalisée de façon récursive, ce qui permet de ne lancer qu'une seule fois la méthode de segmentation avec le seul paramètre N_{max} . Cette approche permet de garder la connexion entre les primitives tout en trouvant une signature représentative d'une ou de plusieurs classes. Toutefois, le paramètre N_{max} a une certaine importance. Il est préférable de le choisir assez grand (par rapport à la longueur d'une phase) afin de laisser suffisamment de flexibilité à la méthode pour trouver une segmentation adaptée. A l'inverse, si le nombre N_{max} est trop petit, le résultat de segmentation ne sera pas suffisant pour discriminer les formes de classes

différentes. Outre le fait de n'avoir que ce seul paramètre N_{max} à définir, l'avantage de la méthode proposée est de commencer par une segmentation de faible précision (1 segment par phase) afin de vérifier dans un premier temps si la caractérisation des formes d'une même classe est possible avec les types de primitives sélectionnées.

3.3.2. Choix des primitives

Comme on l'a vu dans la section 1.2.4.a, plusieurs types de primitives peuvent être choisis pour représenter une forme dynamique. Ce choix se fait en fonction de la précision que l'on souhaite obtenir pour décrire chaque épisode. Par exemple, lorsque l'on cherche à caractériser plusieurs défauts différents sur un système, on doit être capable de différencier ces défauts sur une forme dynamique. Outre la liaison entre les épisodes provenant de la segmentation d'une forme, c'est le type des primitives que l'on souhaite utiliser qui a une importance majeure dans les résultats de classification. De la même façon puisque l'on cherche à interpréter chaque primitive, il peut devenir trop compliqué d'interpréter une forme dynamique si de nombreuses primitives sont utilisées. C'est également en fonction des caractéristiques des signaux (bruités ou débruités) que l'on utilise certaines primitives. On utilise généralement entre 2 (constante et droite) et 13 (celles de Colomer, Figure 1.10 du chapitre 1) types différents de primitives pour caractériser une forme dynamique. Toutefois, un même type de primitives peut être décliné en de nombreuses variantes ; notamment en fonction de la pente de chaque primitive utilisée. Dans la section suivante, la méthode de segmentation utilisée dans l'approche structurale de reconnaissance des formes proposée, est détaillée.

3.3.3. Méthode de segmentation

La méthode que nous employons n'utilise comme seul paramètre que le nombre de primitives maximal N_{max} autorisé pour chaque phase d'une forme dynamique, qui est défini par expérimentation. La méthode est une variante des approches descendantes de segmentation. Elle commence par segmenter une phase entière de forme dynamique pour segmenter ensuite les épisodes obtenus. Les Points de Rupture (PtR) sont définis par interpolation en utilisant l'algorithme suivant :

- le premier point de rupture (PtR_1) est le point de commencement de la phase,
- le dernier point de rupture (PtR_n) est le point final de la même phase,
- puis pour chaque couple de points de rupture consécutifs (PtR_i et PtR_j) :

- on calcule l'erreur d'approximation générée par le segment leur correspondant. Cette erreur est calculée comme la somme cumulée des valeurs absolues des écarts d'amplitude de tous les points du segment $[PtR_i; PtR_j]$ par rapport à leur valeur réelle,
- durant le calcul d'erreur d'approximation, on cherche le nouveau point de rupture entre PtR_i et PtR_j qui fait décroître le plus l'erreur d'approximation du segment. En effet, en créant un point de rupture, l'erreur d'approximation décroît puisque la précision de la segmentation s'accroît. Chaque point situé entre PtR_i et PtR_j est testé comme candidat pour être le nouveau point de rupture,
- c'est entre les points PtR_i et PtR_j pour lesquels l'erreur d'approximation est maximale qu'on ajoute le nouveau point de rupture. Ce dernier est celui obtenu durant le calcul de l'erreur d'approximation du segment,
- jusqu'à obtention du nombre de primitives désiré pour une phase, les nouveaux points de rupture sont trouvés récursivement par minimisation globale de l'erreur d'approximation. L'erreur d'approximation est donc considérée dans la phase de segmentation, mais elle ne nécessite pas la définition de seuils.

Un exemple de segmentation est détaillé sur la figure 3.6 en utilisant cet algorithme pour chacune des étapes de création d'un point de rupture.

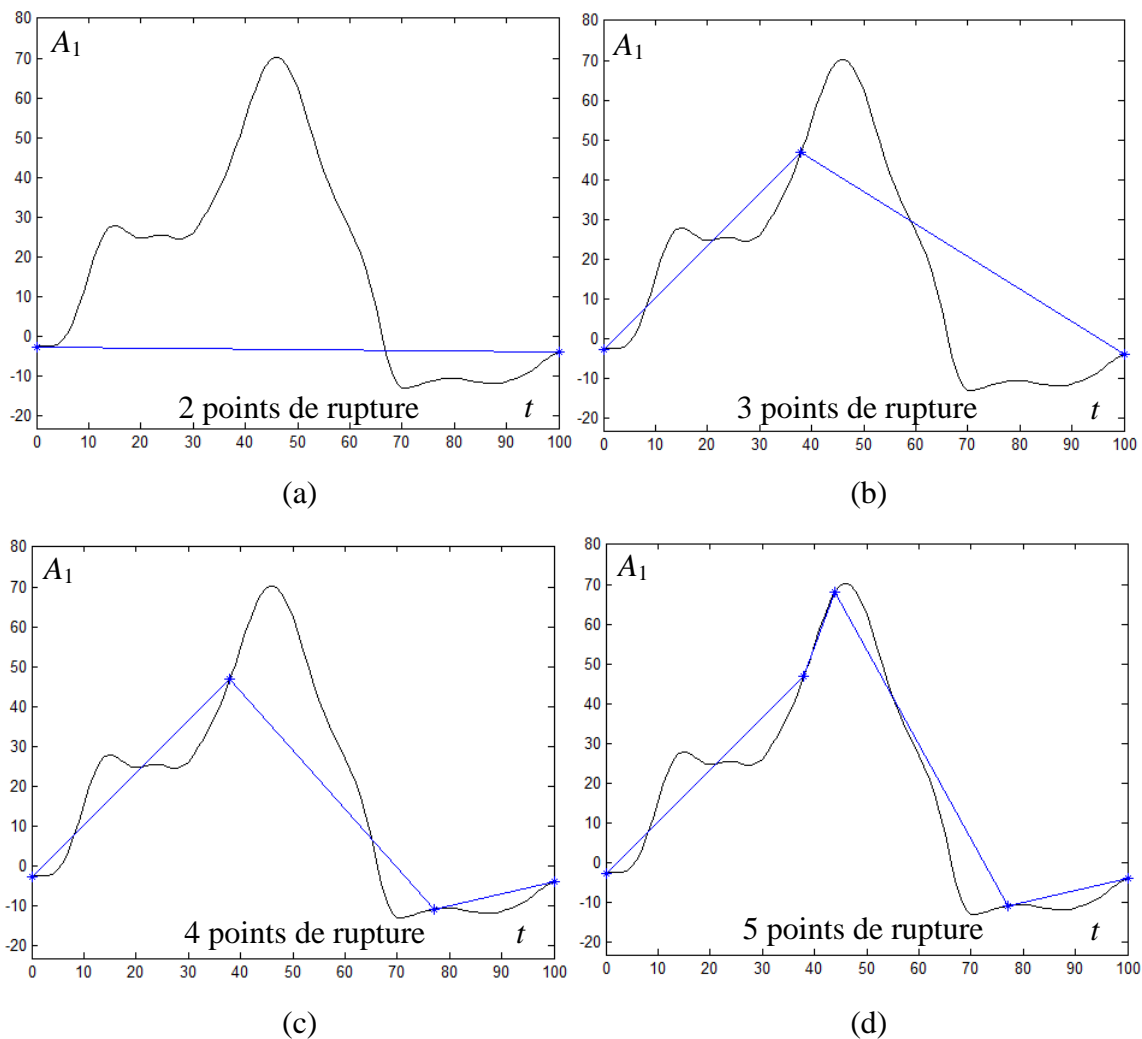


Figure 3.6. Exemple de segmentation détaillé étape après étape. 2 points de rupture '*' sont obtenus sur (a), 3 sur (b), 4 sur (c) et 5 sur (d). Le signal traité est représenté par '—' et les segments obtenus sont représentés par '—'.

L'exemple de segmentation de la figure 3.6 montre l'utilisation de 5 points de rupture pour la dernière segmentation obtenue. L'erreur d'approximation diminue avec l'augmentation du nombre de points de rupture utilisé. Toutefois, un trop grand nombre de points de rupture ne permet pas de bien caractériser l'ensemble des formes d'une même classe. Sur la figure 3.6, il est donc possible de continuer d'augmenter le nombre de points de rupture, et donc le nombre de primitives, jusqu'à obtenir la précision d'approximation désirée ou la caractérisation des formes d'une même classe.

En utilisant cet algorithme, aucune erreur d'approximation ni aucun seuil n'a besoin d'être défini. Une fois la signature obtenue pour l'ensemble des parties caractérisant un système, nous proposons de regrouper les différentes signatures obtenues. Prenons l'exemple simulé d'un

système fonctionnant en deux parties, par exemple un système réalisant un perçage puis une soudure. Si l'on réussit à caractériser la phase de perçage d'une pièce à l'aide d'une signature, puis la phase de soudure par une autre signature, on peut ensuite dire que la réalisation globale d'une pièce peut être réalisée par le rassemblement des deux signatures obtenues. En effet, si l'on considère qu'un système est dans son fonctionnement normal quand toutes ses signatures correspondent à un fonctionnement normal, alors on peut regrouper l'ensemble des signatures correspondant au fonctionnement normal de ce système en une seule signature globale. Il en va de même pour les autres modes de fonctionnement. Un exemple de rassemblement de deux signatures est présenté sur la figure 3.7 en utilisant les primitives définies auparavant.

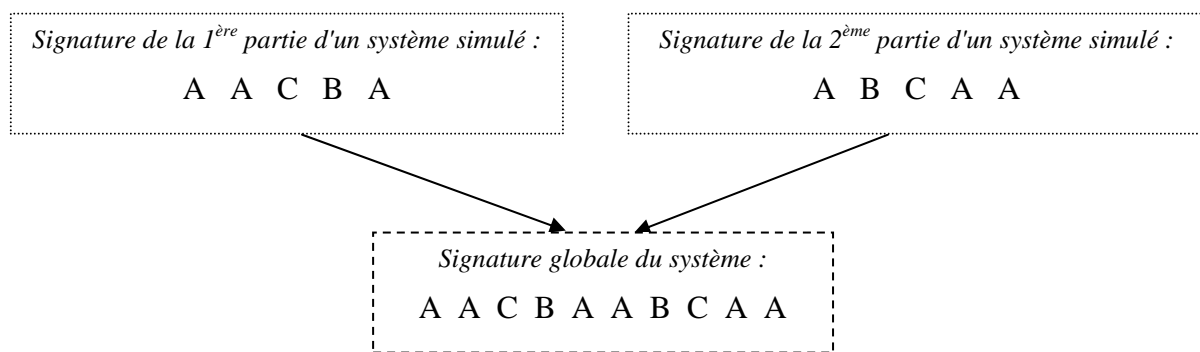


Figure 3.7. Exemple de rassemblement de deux signatures obtenues pour caractériser un mode de fonctionnement d'un système.

Dans la section suivante, la signature globale obtenue pour chaque forme dynamique du fonctionnement d'un système est classifiée.

3.3.4. Classification des formes dynamiques par mesure de similarité

Lorsque la signature globale d'une forme dynamique caractérise un mode de fonctionnement, cela revient à dire que c'est la liaison entre les primitives utilisées qui caractérise ce mode de fonctionnement. Plus précisément, c'est l'évolution temporelle des primitives, i.e. leur enchaînement dans un ordre précis, qui va permettre de classer une forme dans une classe. On peut donc dire que chaque primitive caractérise partiellement le mode de fonctionnement.

Quand on cherche à discriminer seulement deux classes sur un système, il suffit de caractériser une classe avec un maximum de précision pour réussir à discriminer les formes de l'autre classe. Lorsque cela est possible, on peut aussi chercher à caractériser chaque classe par une signature propre. Dans cette partie, nous proposons de classer chaque forme par

comparaison (Dormido-Canto, 2008) avec la signature globale obtenue pour chaque classe. Pour cela, on considère que la $i^{\text{ème}}$ primitive de la signature globale d'une classe doit être identique à la $i^{\text{ème}}$ primitive de la signature d'une forme. Le nombre total de primitives identiques entre la signature d'une forme et la signature d'une classe est alors utilisé pour mesurer la similarité envers chaque classe. L'ordre des primitives est ainsi respecté. Si les primitives sont exactement les mêmes alors elles auront un coefficient égal à 1 dans la mesure de la similarité, si elles sont complètement différentes le coefficient est égal à 0 et si elles se ressemblent (par exemple : droite croissant faiblement et droite croissant très faiblement) elles auront une valeur de similarité prédéfinie (Dash, 2003). On obtient alors un coefficient global flou de similarité entre formes dynamiques :

$$sim_{struct}(Sig1, Sig2) = \frac{\sum_{i=1}^n S(p_i^{Sig1}, p_i^{Sig2})}{n} \quad (3.1)$$

où $Sig1$ et $Sig2$ sont deux signatures de longueur n . sim_{struct} représente la mesure de similarité structurelle entre $Sig1$ et $Sig2$. S est la valeur floue de similarité qui doit être définie entre chaque couple de primitives. p_i^{Sig1} représente la $i^{\text{ème}}$ primitive de $Sig1$. La valeur de similarité entre deux signatures est donc obtenue par la somme totale des similarités entre toutes leurs primitives, divisé par leur nombre. Il est également possible d'utiliser cette mesure pour calculer la similarité entre une signature $Sig1$ et une classe. Dans ce cas, il faut mesurer la similarité entre $Sig1$ et la signature caractéristique d'une classe.

Prenons un exemple pour calculer la mesure de similarité entre deux signatures. Nous allons considérer l'utilisation des 7 primitives (allant de A à G) définies dans (Dash, 2003) et dont la matrice de valeurs de similarité de ces primitives est donné dans (Dash, 2003) par :

Tableau 3.1. Exemple de valeurs de similarité définies entre 7 primitives (Dash, 2003).

	A	B	C	D	E	F	G
A	1	0	0,25	0	0	0,25	0
B	0	1	0,75	0,5	0	0	0
C	0,25	0,75	1	0,75	0	0	0
D	0	0,5	0,75	1	0	0	0
E	0	0	0	0	1	0,75	0,5
F	0,25	0	0	0	0,75	1	0,75
G	0	0	0	0	0,5	0,75	1

Si l'on considère la première signature définie par « *A B G E D* », et la deuxième signature définie par « *A C D E C* », le déroulement de la mesure de similarité calculée par (3.1) est égal à :

$$sim_{struct} = \frac{S(A,A) + S(B,C) + S(G,D) + S(E,E) + S(D,C)}{5}$$

$$sim_{struct} = \frac{1 + 0,75 + 0 + 1 + 0,75}{5} = \frac{3,5}{5} = 0,7$$

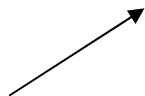
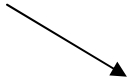
La similarité structurelle entre les deux signatures est donc égale à 0,7. La mesure de similarité est une valeur floue [0;1], avec une précision tenant compte de la longueur de la signature utilisée. Plus la longueur d'une signature est importante, plus la précision de la mesure de similarité peut être importante. Cette mesure permet de quantifier l'évolution que réalise un système après une modification de son réglage, du changement d'une pièce ou pour l'application clinique lorsqu'un patient a eu un traitement thérapeutique. Dans ce dernier cas, la quantification peut permettre aux cliniciens de valider un traitement pour certains patients ou de la changer pour d'autres.

Dans la partie suivante, l'approche proposée est appliquée à deux exemples illustratifs.

3.3.5. Exemples illustratifs

Les deux exemples suivants illustrent l'utilisation de l'approche de segmentation proposée qui caractérise la méthode structurelle de classification proposée. Nous cherchons à présenter dans cette partie les capacités de caractérisation des formes dynamiques à l'aide de la méthode structurelle proposée. Les signaux ne sont pas séparés en phases spécifiques afin de montrer l'application de la méthode sur des signaux complets. Les primitives que l'on utilise sont celles du tableau 3.2 (*A*, *B* et *C*).

Tableau 3.2. Les trois primitives sélectionnées.

<i>Episode</i>			
<i>Primitive</i>	A	B	C

Premier exemple de segmentation :

Le signal traité dans cette partie concerne l'évolution d'un paramètre p en fonction du temps. Le type de signal que l'on traite dans cette partie est non bruité et est de la forme de celui présenté sur la figure 3.8.

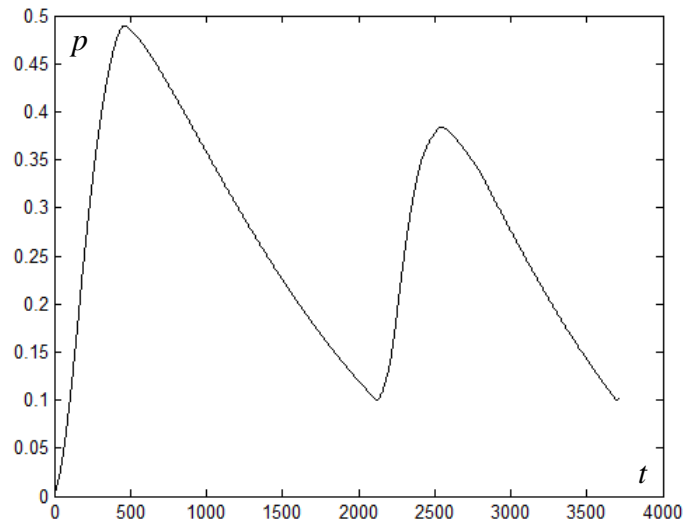


Figure 3.8. Signal représentant l'évolution d'un paramètre p .

La réalisation de la segmentation du signal est présentée sur la figure 3.9, en utilisant la méthode de segmentation proposée dans la partie 3.3.3 avec un nombre de primitives allant de 1 à 7.

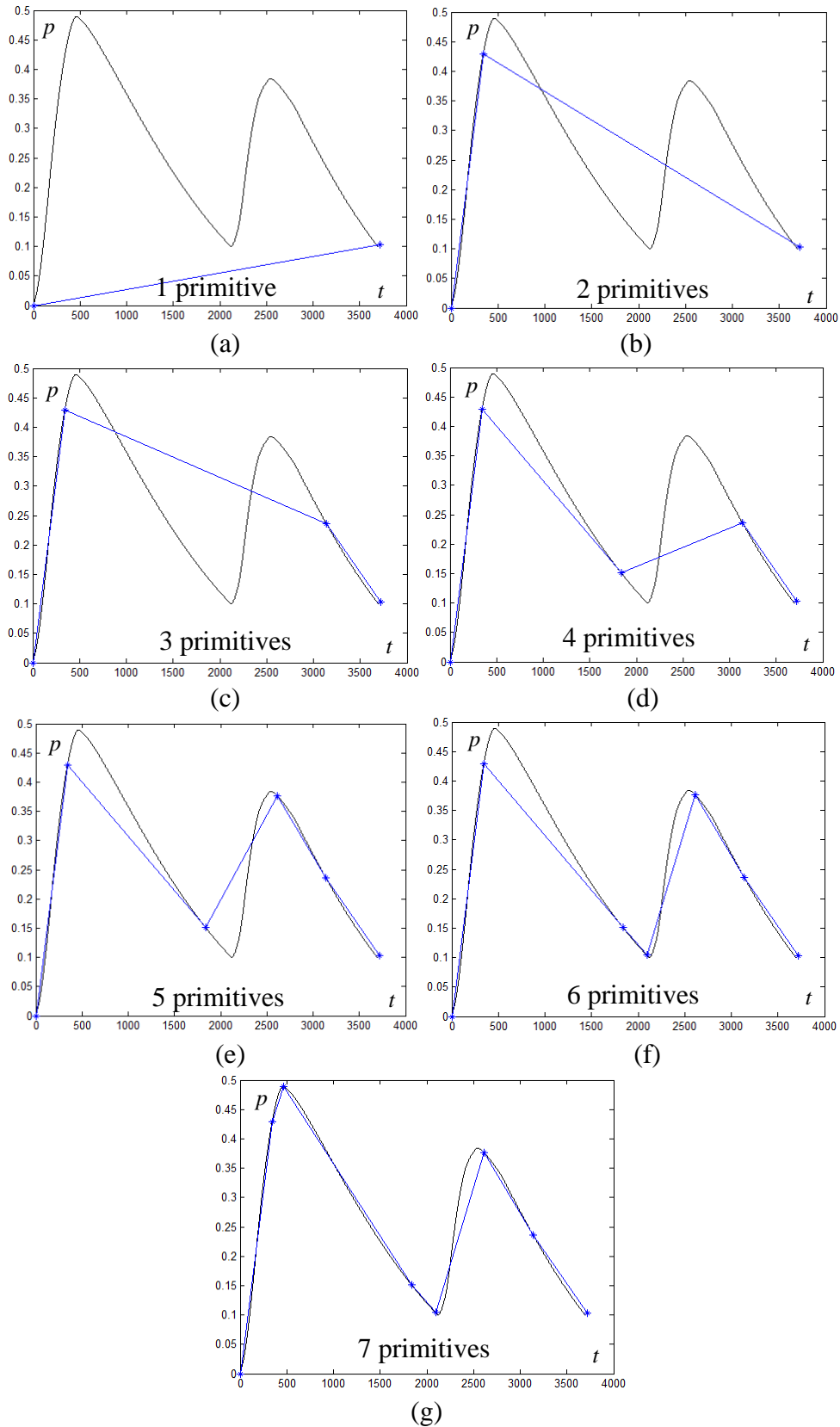


Figure 3.9. Segmentation d'un signal avec de 1 à 7 primitives, respectivement de (a) à (g). Les points de rupture sont représentés par des '*'. Le signal traité est représenté par '—' et les segments obtenus sont représentés par '—'.

Avec l'utilisation de 7 primitives, le résultat de segmentation obtenu permet de caractériser ce signal et sa classe. En utilisant seulement 5 primitives, il était déjà possible d'obtenir une signature caractéristique du signal. La signature que l'on obtient pour le signal de la figure 3.9 avec 7 primitives est égale à : $A A C C A C C$. Cela nous permet de considérer que la deuxième montée en amplitude de p peut être de plus faible ampleur que la première montée en amplitude de p puisqu'il y a une seule primitive A . Un changement survenant dans l'établissement des primitives obtenues permet donc de bien caractériser un changement sur un système. En effet sur ce signal au début de la deuxième montée en amplitude de p , un défaut a été généré ; il est donc normal que ce changement soit perceptible et qu'il se retrouve dans la signature obtenue.

Sur cet exemple ce sont 3750 valeurs qui ont été utilisées pour caractériser la forme dynamique du système. Cette forme dynamique pourrait contenir beaucoup plus de données. Dans ce cas il serait important de réaliser la segmentation des autres données avec le même nombre de primitives optimal, i.e. 7 primitives, et une fenêtre glissante de 3750 valeurs. On pourrait alors réaliser une comparaison des signatures obtenues pour chaque fenêtre glissante et une classification des formes dynamiques de ce système. Pour la définition de la fenêtre glissante à utiliser, il est important que celle-ci ait une taille au moins égale à la durée d'un cycle du système observé, de telle façon que la structure d'un cycle puisse être déterminée. Les résultats de segmentation dépendent en effet de cette taille de fenêtre. Si le signal observé s'arrête au milieu d'un cycle, il est important qu'il en soit de même pour le prochain signal observé.

Deuxième exemple de segmentation :

Sur ce deuxième exemple, nous nous intéressons au signal de la figure 3.10, qui permet de suivre l'évolution d'un paramètre p en fonction du temps.

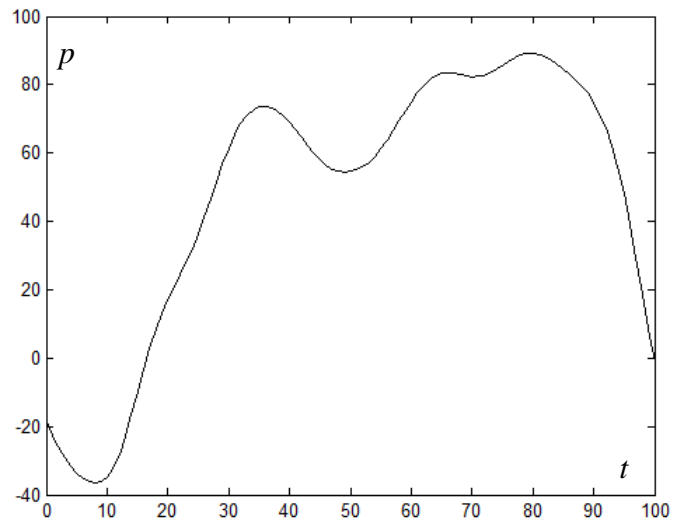


Figure 3.10. Signal représentant l'évolution d'un paramètre p .

La réalisation de la segmentation du signal de la figure 3.10 est présentée sur la figure 3.11. La méthode de segmentation utilisée est celle de la partie 3.3.3 avec un nombre de primitives allant de 1 à 6.

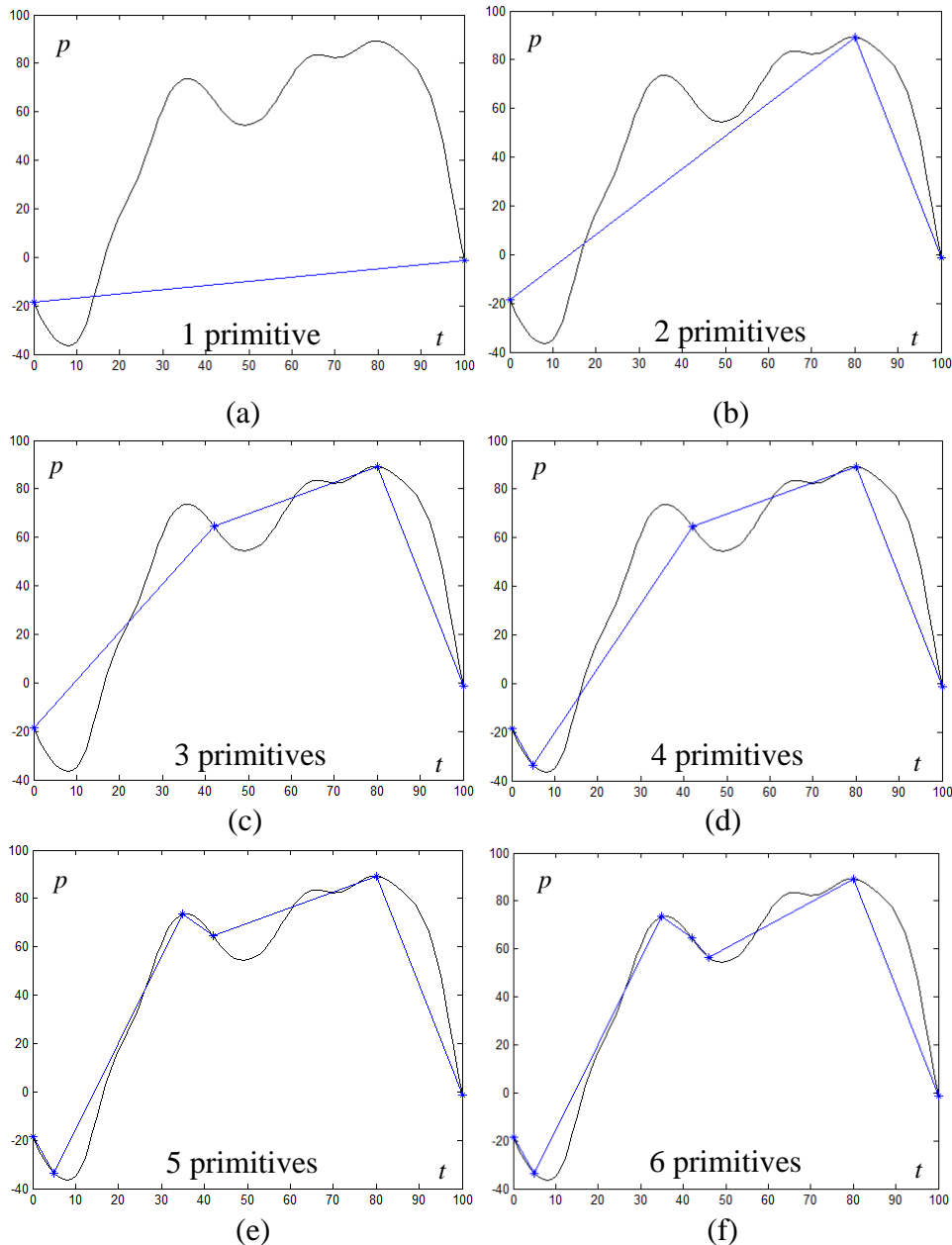


Figure 3.11. Segmentation d'un signal avec de 1 à 6 primitives, respectivement de (a) à (f). Les points de rupture sont représentés par des '*'. Le signal traité est représenté par '—' et les segments obtenus sont représentés par '—'.

Les six primitives utilisées sur la figure 3.11.f pour représenter la signature de cet exemple permettent d'obtenir une caractérisation suffisante des formes de la même classe. Six primitives sont donc suffisantes pour caractériser ces signaux. La signature obtenue est égale à : $C A C C A C$. Toutefois, on pourrait avoir des primitives semblables dans le cas d'un changement de dynamique très faible (vers un mode normal ou défaillant). Dans ce cas il faudrait utiliser plus de types de primitives ou attendre suffisamment longtemps pour détecter l'accumulation des changements réalisés. En effet, c'est la répétition des primitives de la

signature obtenue qui nous permet de détecter un changement si la surveillance du système se fait en ligne. Comme nous l'avons dit précédemment, plus le nombre de primitives est élevé plus il sera rapide de détecter un changement sur le système (puisque la précision est plus grande). Le problème d'un nombre élevé de primitives est qu'il devient difficile de caractériser un mode de fonctionnement.

L'utilisation de la méthode structurelle de reconnaissance des formes dynamiques développée dans ce chapitre peut se faire de façon conjointe à une méthode de reconnaissance statistique des formes. On appelle alors cette approche une méthode mixte de reconnaissance des formes. Dans la partie suivante, plusieurs approches mixtes de la littérature sont présentées, dont une approche basée sur la méthode structurelle que l'on vient de développer. L'approche de classification mixte proposée repose sur la définition d'une signature mixte permettant d'interpréter les résultats. Dans la première partie de la méthode, i.e. la phase de recherche des paramètres informatifs, la segmentation des formes dynamiques est réalisée et le nombre d'épisodes (ou de primitives) optimal est sélectionné. C'est alors en fonction de ces épisodes que vont être calculés les paramètres statistiques. C'est cette caractérisation mixte des formes dynamiques qui va permettre par la suite à la méthode structurelle (précédemment développée) et à la méthode statistique sélectionnée de classifier chacune des formes.

3.4. Combinaison de classifieurs basée sur les caractéristiques statistiques et structurelles des formes dynamiques

Dans la partie précédente, nous avons détaillé les approches de reconnaissance structurelle des formes dynamiques. L'approche proposée a été appliquée à des signaux non bruités et la caractérisation de ces signaux a été réalisée. Toutefois, il peut arriver que les signaux ne comportent pas suffisamment de caractéristiques structurelles propres à chaque classe, pour pouvoir correctement les discriminer. Dans ce cas, on peut faire appel en complément, aux méthodes de reconnaissance des formes basées sur les propriétés statistiques des formes. On peut alors utiliser n'importe quelle méthode statistique existante, qu'elle soit supervisée, non supervisée, semi-supervisée, paramétrique, non paramétrique, etc.

L'ajout des informations statistiques issues des formes doit permettre de compléter la représentation structurelle obtenue pour chaque forme dynamique. Les paramètres statistiques peuvent alors apporter des informations concernant la longueur d'un épisode, l'amplitude des épisodes, ou d'autres valeurs comme la moyenne, la valeur RMS, ou le nombre de pics contenus dans le signal complet ou dans certaines de ses parties. En ajoutant ces paramètres

on cherche à caractériser d'avantage chaque forme et ainsi à améliorer le taux de classification.

Dans la partie suivante, nous détaillons les différentes structures qu'un multi-classifieur peut avoir et quelques approches de multi-classifieur voient également leur principe étudié.

3.4.1. Méthodes existantes

3.4.1.a. Principe de fonctionnement des multi-classifieurs

Les multi-classifieurs peuvent utiliser les informations statistiques et structurelles des données (Bunke, 1990) de façon répétée ou complémentaire. Certaines approches utilisent plusieurs classifieurs statistiques, d'autres approches utilisent plusieurs classifieurs structurels et d'autres utilisent encore des classifieurs structurels et statistiques pouvant tirer parti de la complémentarité des informations.

L'utilisation combinée de plusieurs classifieurs dans un multi-classifieur a pour but d'obtenir :

- une plus grande précision lors de la classification des données, i.e. un meilleur taux de classification,
- une plus grande fiabilité des résultats. En effet, plus les décisions des classifieurs mènent aux mêmes résultats, plus le résultat global de classification est sûr,
- une gestion différente du problème de classification. On peut en effet décomposer un problème initial en sous problèmes moins complexes pour réduire la complexité de décision de classification d'une forme,
- une exploitation des informations complémentaires des données.

Chaque méthode de classification d'un multi-classifieur peut être considérée (Giacinto, 1998) comme faisant soit partie d'une approche par fusion ou combinaison (Ho, 2002) (Cremer, 2001), ou faisant partie d'une approche par coopération ou sélection (Auda, 2000) (Aksela, 2003) (Ruta, 2000).

On parle de fusion des classifieurs quand l'association de classifieurs est réalisée parallèlement (Giani, 1999) ou séquentiellement (Gunes, 2003). Une association parallèle de classifieurs utilise chaque classifieur indépendamment et généralement en même temps, puis la fusion des résultats de chaque classifieur est réalisée pour obtenir un compromis de classification entre l'ensemble des résultats obtenus par les classifieurs. La fusion des résultats

peut ne favoriser aucun classifieur ou au contraire on peut attribuer à chaque classifieur un poids dans la considération de ses résultats (en fonction de ses performances, etc.). Parmi les méthodes de combinaison des résultats les plus utilisées on peut citer le système de vote majoritaire (Bahler, 2000), la combinaison des résultats qui utilise la théorie Bayésienne (Bahler, 2000), une combinaison qui utilise la théorie de Dempster-Shafer pour fusionner les mesures d'évidence obtenues par différents classifieurs (Al-Ani, 2002), les règles d'agrégation fixes comme le maximum, le minimum, la somme, la moyenne ou le produit (Chen, 1997), etc.

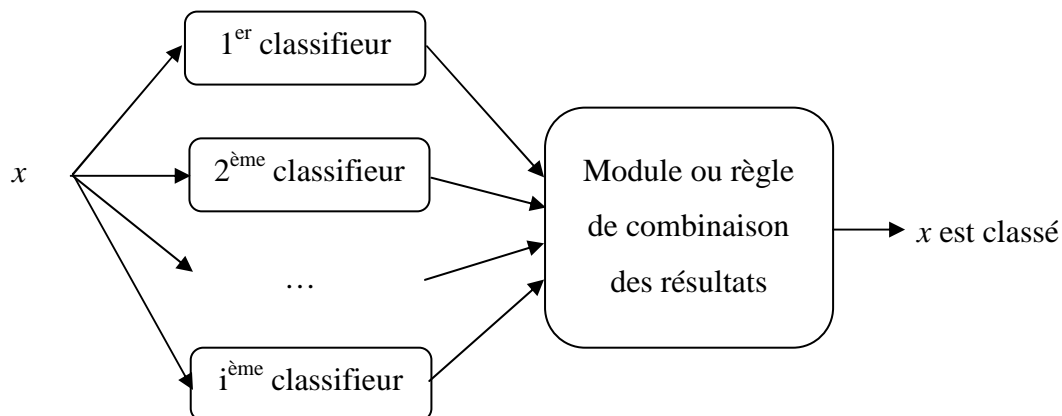


Figure 3.12. Multi-classifieur possédant une architecture parallèle.

A l'inverse, un multi-classifieur séquentiel utilise successivement chaque classifieur (Figure 3.13). Un multi-classifieur séquentiel peut donc réduire progressivement le nombre de classes possibles pour une forme, ou le nombre de données ambiguës à classer. L'objectif de cette technique est d'utiliser les décisions d'un premier classifieur pour obtenir de meilleurs résultats de classification par la suite. On peut choisir de ne classer successivement que les données ambiguës ou l'ensemble des données. L'importance de l'ordre des classifieurs peut alors avoir une place majeure dans les résultats obtenus.

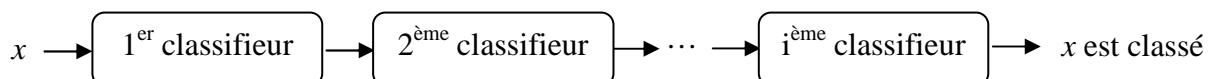


Figure 3.13. Multi-classifieur possédant une architecture séquentielle.

On peut également combiner les deux types d'approches pour obtenir un multi-classifieur séquentiel et parallèle. On tient alors compte de la réduction progressive des données à classer et du consensus au niveau de la décision à prendre.

On parle de sélection de classifieurs quand un ou plusieurs classifieurs sont utilisés par d'autres classifieurs pour obtenir le résultat global de classification d'un multi-classifieur. Ce type de multi-classifieur cherche à sélectionner les classifieurs les mieux adaptés à la classification de chaque forme. La sélection des classifieurs peut se faire de façon statique (Aksela, 2003) ou dynamique (Giacinto, 2000). Dans le cadre d'une sélection statique des classifieurs, le multi-classifieur désigne dans la phase d'apprentissage les classifieurs à utiliser. Quand la sélection des classifieurs est dynamique, le multi-classifieur choisit dans la phase de classification le classifieur le plus adapté à l'échantillon à classer.

Dans la section suivante, plusieurs exemples de multi-classifieur sont présentés.

3.4.1.b. Exemples de multi-classifieur

(Ayad, 2010) propose d'utiliser deux classifieurs statiques afin d'améliorer le taux de classification dans les zones d'ambiguïté. Pour cela, l'auteur utilise la méthode Fuzzy Pattern Matching Semi-Supervisée (FPMSS) (Sayed Mouchaweh, 2010) et la méthode des Séparateurs à Vaste Marge (SVM) (Vapnik, 1995). Dans la phase d'apprentissage, le multi-classifieur utilise FPMSS pour construire les fonctions de décision (fonctions d'appartenance des classes) et pour déterminer ensuite les zones d'ambiguïté entre les classes. Les zones d'ambiguïté sont définies par les formes de l'espace de représentation possédant une valeur d'appartenance sensiblement proche pour plusieurs classes. Puis ces formes sont utilisées par la méthode SVM pour discriminer les classes connues. Dans la phase de classification, chaque nouvelle forme est d'abord classifiée par FPMSS puis son rapport d'ambiguïté en appartenance par rapport à plusieurs classes est calculé. Si ce rapport est inférieur à un seuil de tolérance d'ambiguïté fixé alors la forme est classée dans la classe trouvée par FPMSS sinon la forme est classifiée par SVM. Enfin, si la forme a des valeurs d'appartenance nulles lors de sa classification, un nouveau prototype de classe est créé par FPMSS. Les avantages principaux dérivant de la combinaison de ces classifieurs pour les zones d'ambiguïté sont d'obtenir un meilleur taux de classification et de détecter l'apparition de nouvelles classes par rapport à l'utilisation seule de SVM.

Dans (Wang, 2003b), l'auteur précise que quand les résultats de plusieurs classifieurs sont utilisés par un multi-classifieur, ils ont généralement un poids équivalent. Cependant, en fonction de l'application sur lequel le multi-classifieur est utilisé, la qualité de classification de chaque classifieur peut fortement varier. C'est pourquoi (Wang, 2003b) choisit de définir un poids propre à chaque classifieur utilisé. Le poids donné aux résultats de classification obtenus par chaque classifieur est équivalent à la précision de prédiction que chaque

classifieur aura obtenu sur des données tests. Puis pour chaque nouvelle donnée, les classifieurs vont être utilisés dans un ordre donnant la priorité à celui qui a eu les meilleurs résultats jusqu'à celui qui a eu les moins bons résultats. Tous les classifieurs ne sont pas nécessairement utilisés pour chaque forme. Il peut être possible d'arrêter la classification d'une forme, dès lors qu'un taux de confiance suffisant est obtenu. Ce taux de confiance est calculé après la classification de chaque forme par chaque classifieur, en fonction de la précision de chaque classifieur, ainsi que de la moyenne et de la variance des données. Les classifieurs utilisés par (Wang, 2003b) sont C4.5 (Quinlan, 1993), Ripper (Cohen, 1995) et l'approche Bayésienne.

(Littlestone, 1994) utilise le vote majoritaire sur un ensemble de classifieurs afin de créer son multi-classifieur. Pour cela chaque classifieur commence avec un poids de 1 dans la décision finale de classification d'une forme. Ce poids est diminué quand la prédiction du classifieur est mauvaise. L'établissement de ces poids est réalisé avec un ensemble d'apprentissage. La décision globale du multi-classifieur repose sur le poids finalement estimé pour chaque classifieur, et c'est la classe qui obtient le poids maximal qui est désignée pour la classification d'une forme.

Dans la section suivante, nous proposons une approche mixte de reconnaissance des formes dynamiques dont le résultat se veut entièrement interprétable.

3.4.2. Approche mixte proposée de reconnaissance des formes dynamiques

Dans cette partie nous détaillons le principe de fonctionnement de l'approche multi-classifieur (mixte) proposée (Hartert, 2010e). Les informations statistiques et structurelles issues des formes dynamiques sont ici utilisées de façon complémentaire. En effet, la décision prise par le classifieur global repose sur le résultat obtenu pour la classification de chaque forme par rapport à chaque classifieur utilisé. C'est l'utilisation parallèle d'un classifieur statistique et d'un classifieur structurel qui rend mixte l'approche proposée.

Le multi-classifieur proposé utilise l'approche structurelle précédemment développée. Cette méthode se veut donc plutôt utilisable hors ligne. De plus, nous avons choisi d'utiliser chaque classifieur pour réaliser la classification de chaque forme, afin de privilégier la double validation des résultats plutôt que le temps de classification (qui importe peu hors ligne). La classification peut dans ce cas être réalisée en parallèle. De ce fait le problème de sélection des formes devant être reclassées une deuxième fois dans le cadre d'une combinaison série, est également supprimé. Toutes les formes sont donc classées deux fois, en utilisant les avantages propres à chaque type de méthode, c'est-à-dire en utilisant la structure historique

des formes et leurs caractéristiques statistiques particulièrement informatives. Le multi-classifieur tient alors compte des informations temporelles et statistiques pour valider la classification de chaque forme dynamique.

Dans un premier temps nous allons décrire l'approche statistique, puis l'approche mixte permettant de classifier les données, puis nous présenterons une mesure de similarité mixte. Le principe de l'approche multi-classifieur mixte proposée est résumé sur la figure 3.14.

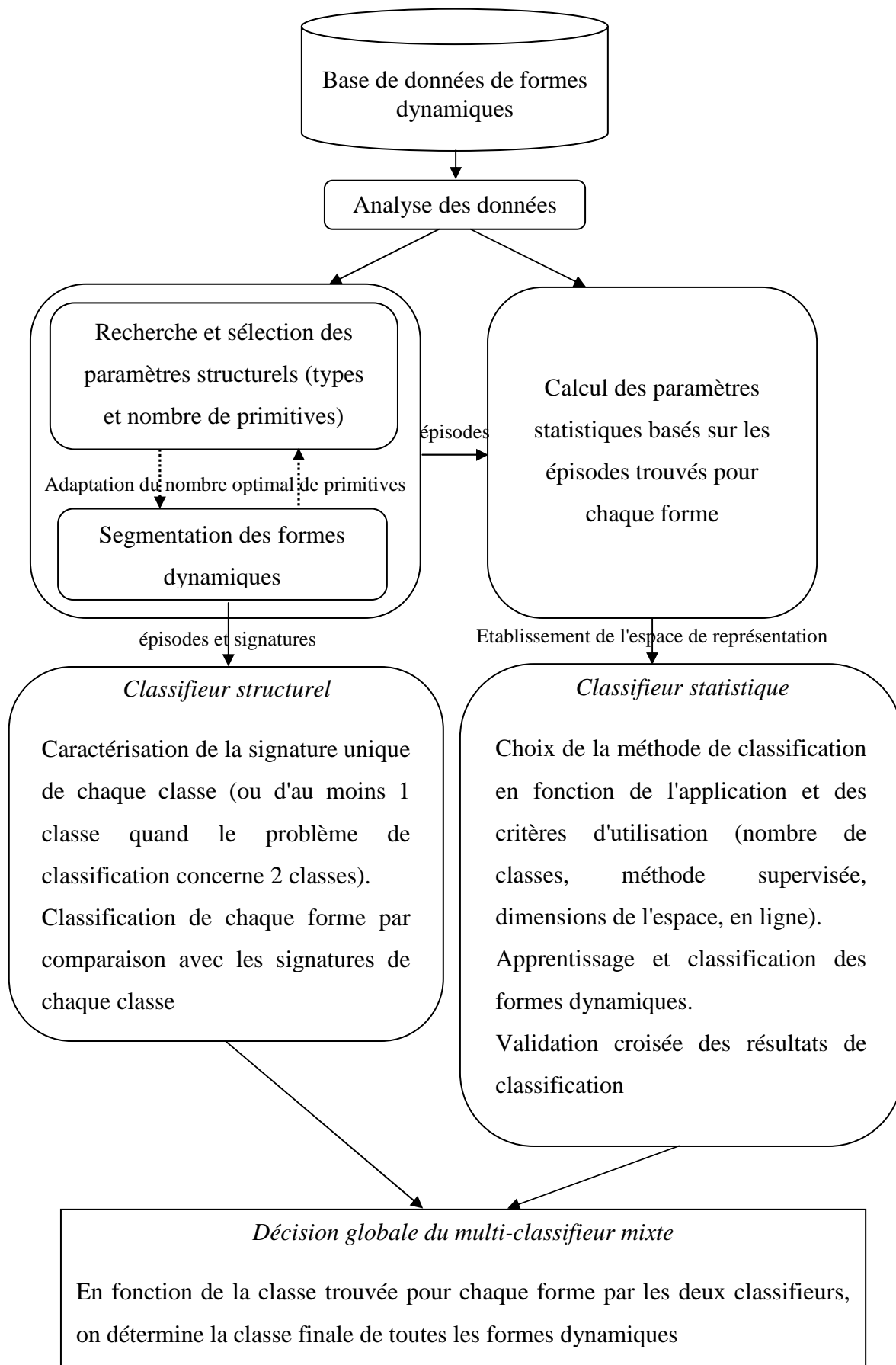


Figure 3.14. L'approche multi-classifieur mixte proposée.

3.4.2.a. Méthode de classification statistique

Dans le cadre de l'utilisation du multi-classifieur mixte proposé, il est nécessaire de choisir une méthode de classification statistique qui soit adaptée à l'application sur laquelle la méthode va être utilisée. Dans tous les cas, on doit d'abord analyser les formes issues de l'application pour obtenir les paramètres statistiques les plus informatifs permettant de discriminer au maximum chaque classe.

Détermination des paramètres statistiques :

L'une des particularités de l'approche proposée est d'obtenir une correspondance entre les paramètres statistiques et les paramètres structurels. C'est pourquoi nous avons jugé représentatif que les paramètres statistiques ne soient pas calculés sur n'importe quelle partie du signal (ou même sur le signal complet simplement). Nous avons choisi de baser les caractéristiques statistiques sur les épisodes obtenus par la méthode de reconnaissance structurelle des formes dynamiques. Par exemple, si la segmentation d'une forme dynamique mène à l'obtention de 4 épisodes, cela signifie que les caractéristiques structurelles informatives de cette forme sont obtenues sur chacun de ces 4 épisodes. De la même façon, les caractéristiques statistiques de chacun de ces épisodes doivent être informatives. De plus, l'utilisation de paramètres statistiques et structurels qui concernent les mêmes parties des formes dynamiques permet une meilleure comparaison des résultats des méthodes.

Parmi les paramètres statistiques que l'on peut calculer on peut citer : la moyenne d'amplitude, le nombre de pics, le moment où l'amplitude est minimale, la longueur d'un épisode, la valeur du Kurtosis, la valeur RMS, le temps de croissance d'un épisode, etc. Dans la mesure du possible, il est préférable que ces paramètres soit interprétable. Une fois les paramètres calculés, il est nécessaire de sélectionner ou extraire les paramètres les plus informatifs avec une méthode telle que l'Analyse en Composantes Principales (ACP), la mesure discriminante de Fisher ou d'autres techniques. Le paramètre statistique sélectionné peut être différent pour chaque épisode, mais dans ce cas l'interprétation du résultat peut être plus difficile à réaliser. Puis les paramètres obtenus permettent d'établir l'espace de représentation de la méthode de classification statistique. Chaque forme dynamique est alors représentée par un seul point dans l'espace de représentation.

Choix d'une méthode de classification statistique :

Le choix d'une méthode de classification statistique doit se faire en fonction de différents paramètres propres à l'application sur laquelle la méthode va être appliquée et à différentes contraintes définies par les utilisateurs de l'application. Parmi les caractéristiques qui doivent influencer l'utilisateur sur le choix de la méthode, nous pouvons citer :

- la taille de l'ensemble d'apprentissage,
- la complexité et le nombre de classes,
- la non-linéarité des données,
- le fait de travailler en ligne ou hors ligne,
- la disponibilité d'un expert, impliquant l'utilisation d'une méthode supervisée, non supervisée ou semi-supervisée,
- le retard de détection maximal autorisé, qui est en relation avec le temps de classification de chaque méthode par rapport à chaque application,
- la quantité de données à traiter,
- la nature du système (système évolutif, etc.),
- etc.

En fonction de l'ensemble de ces critères, l'utilisateur pourra alors être amené à choisir une méthode de classification statistique parmi la multitude de méthodes qui existent, avec entre autres :

- Fuzzy Pattern Matching (Cayrol, 1982) et ses extensions,
- K-Plus Proches Voisins (Cover, 1967) et ses extensions,
- Séparateurs à Vaste Marge (Vapnik, 1995) et ses extensions (Dufrenois, 2009),
- Fuzzy C-Means (Bezdek, 1981) et ses extensions,
- Auto-Adaptive & Dynamical Clustering (Lurette, 2003),
- evolving Takagi-Sugeno (Lughofer, 2009),
- Learn++ (Byorick, 2003),
- Concept-adapting Very Fast Decision Tree (Hulten, 2001),
- Self-Adaptive Kernel Machine (Amadou Boubacar, 2005a),
- evolving Neural Network (Angelov 2004b),
- SIMilarity MAPping (Medina, 2009),
- General Fuzzy Min-Max Neural Network (Gabrys, 2000),
- OLIN (Last, 2002),

- AddC (Guedalia, 1999),
- Unsupervised Optimal Fuzzy Clustering (Gath, 1989),
- etc.

De plus, la plupart de ces méthodes ont été développées pour tenir compte de la mise à jour incrémentale des classifieurs, pour diminuer le temps de classification, pour diminuer les contraintes de la méthode, etc. Il est également possible de tester les méthodes afin de sélectionner celle donnant les meilleurs résultats.

Validation croisée des résultats obtenus par une méthode statistique sélectionnée :

Dans le but de confirmer le taux de classification obtenu par la méthode statistique de classification sélectionnée, il est important d'utiliser une technique de validation croisée telle que "Leave one Out" (L1O). Cette technique est présentée sur la figure 3.15.

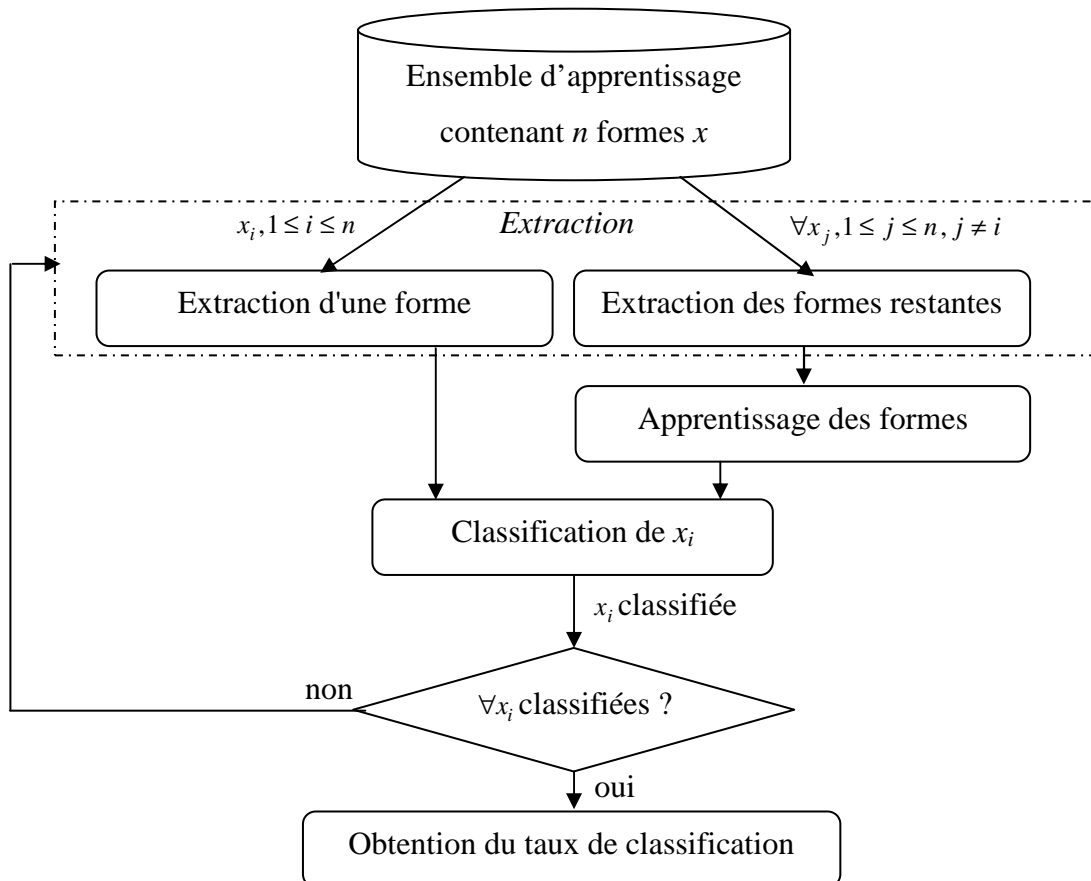


Figure 3.15. Technique de validation croisée « Leave One Out ».

Avec l'utilisation de la méthode structurale en parallèle de la méthode statistique, les résultats de classification obtenus par chaque méthode peuvent être fusionnés.

3.4.2.b. Fusion des résultats par le multi-classifieur

La méthode statistique de classification et la méthode structurale de classification ont été utilisées pour réaliser l'apprentissage et la classification d'une base de données. On obtient pour la méthode structurale un ensemble de signatures qui ont permis de caractériser et de classifier chaque forme. La méthode statistique fournit une décision de classification sur l'appartenance ou non de chaque forme par rapport à chaque classe.

Chaque forme a été classée par rapport à un certain nombre de paramètres pour chaque méthode de classification :

- les primitives pour la méthode structurale, dont le nombre est égal au nombre optimal d'épisodes trouvé par la technique de segmentation de l'approche structurale,
- les paramètres statistiques pour la méthode statistique, dont les valeurs ont été calculées par rapport aux épisodes trouvés par la technique de segmentation.

Règle de décision globale :

Il ne reste plus qu'à confronter les résultats de classification et à prendre une décision globale. Pour ce faire, le multi-classifieur utilise une règle de fusion pour classer chaque forme x_i en fonction du résultat trouvé par les deux classifieurs. La règle est définie comme suit :

$$\begin{aligned} & \text{si } C_{\text{classifieur1}}(x_i) = C_{\text{classifieur2}}(x_i) = C_j \text{ alors } x_i \in C_j, j \in \{1, \dots, c\}, i \in \{1, \dots, n\} \\ & \text{sinon } \begin{cases} \text{si } \text{poids}_{\text{classifieur1}} > \text{poids}_{\text{classifieur2}} \Rightarrow x_i \in C_{\text{classifieur1}}(x_i) \\ \text{si } \text{poids}_{\text{classifieur1}} < \text{poids}_{\text{classifieur2}} \Rightarrow x_i \in C_{\text{classifieur2}}(x_i) \\ \text{si } \text{poids}_{\text{classifieur1}} = \text{poids}_{\text{classifieur2}} \Rightarrow \text{décision différée} \end{cases} \end{aligned} \quad (3.2)$$

où $C_{\text{classifieur1}}(x_i)$ représente la classe trouvée par le classifieur 1, n le nombre de formes classées, $\text{poids}_{\text{classifieur1}}$ correspond au poids ($[0;1]$) du classifieur 1 dans la décision finale. La somme des poids des classifieurs doit être égale à 1. Le poids donné à un classifieur correspond à la fiabilité (ou la confiance) que l'on donne à un classifieur dans la prise de décision finale. Ce poids peut être déterminé en fonction des erreurs de classification que réalise chaque classifieur durant une phase de test. Chaque classifieur qui réussit la classification d'une forme, voit son poids augmenté. A l'inverse chaque erreur de classification diminue le poids donné à un classifieur.

Ainsi, si la classe d'une forme est la même pour les deux méthodes alors la forme est affectée à cette classe. En effet, si les deux classifieurs sont d'accord sur le résultat, il n'y a pas de doute concernant la classification de la forme. Par ailleurs, si la forme est classée par un

classifieur dans une classe C_i , et par l'autre classifieur dans une classe C_j , $i \neq j$, alors on peut avoir un doute sur la classe correspondant à cette forme. Dans ce cas, si l'on considère que le résultat d'un classifieur est plus fiable que celui de l'autre classifieur, on peut choisir de suivre la décision prise par le classifieur le plus fiable sinon on peut différer la décision et demander à l'expert, on parle alors d'apprentissage actif (Lughofer, 2010).

Comme nous l'avons vu dans la section 3.3.3, la signature obtenue pour chaque forme dynamique permet de la caractériser. Toutefois, les caractéristiques de cette signature peuvent ne pas être suffisamment discriminantes pour correctement classifier la forme représentée par cette signature. Il en va de même pour l'interprétation d'une signature. Les primitives d'une signature nous informent de la structure d'une forme dynamique mais elles ne donnent pas certaines informations statistiques pouvant être capitales. Prenons l'exemple simple d'une droite $y=5$. Du point de vue structurel on obtiendra avec une méthode structurelle une primitive simple représentant l'information "droite", qui est exacte. Par contre, on ne pourra pas dire que c'est une droite supérieure à 0. Or cette information peut être très importante. On peut citer un autre exemple qui est celui d'une route avec un virage à 30 degrés. Il n'est pas seulement important de trouver l'information qu'il y a un virage sur une route, mais quelle est l'ampleur de ce virage. C'est pourquoi dans la partie suivante, nous nous intéressons à la définition d'une signature mixte interprétable qui se veut complémentaire aux résultats de classification qu'obtient la méthode mixte proposée. A partir de cette signature mixte, une mesure de similarité mixte est calculée.

3.4.2.c. Signature mixte interprétable et mesure de similarité

L'approche structurelle développée précédemment (section 3.3) permet d'obtenir une signature caractéristique de chaque forme dont les primitives nous permettent de définir une interprétation. De la même façon, les paramètres statistiques obtenus sur chaque épisode permettent de compléter la caractérisation de chaque forme pour réaliser leur classification. La signature mixte interprétable que nous proposons repose donc sur l'utilisation de ces paramètres statistiques et structurels. Du point de vue structurel, on sélectionne la primitive p_i , $i=1 \dots n$ de chaque épisode e_i , avec n le nombre d'épisodes d'une signature. Du point de vue statistique, on sélectionne pour chaque épisode e_i d'une forme, la valeur s_i qui représente la quantification du paramètre statistique correspondant. Si plusieurs paramètres statistiques d'une forme ont été calculés pour un même épisode, on peut choisir d'agréger ces valeurs avec un opérateur tel que : min, max, produit, etc. Chaque épisode e_i est alors caractérisé par une

représentation mixte avec p_i et s_i . Un exemple de signature mixte est présenté dans le tableau 3.3.

Tableau 3.3. Exemple de signature mixte d'une forme dynamique.

Signature	A (P_1)	C (P_2)	B (P_3)	C (P_4)	A (P_5)
Valeurs des paramètres statistiques	0,4 (s_1)	8,2 (s_2)	233 (s_3)	45,5 (s_4)	-6,7 (s_5)
Signature mixte	0,4 A	8,2 C	233 B	45,5 C	-6,7 A

Dans le tableau 3.3, la signature structurelle est formée de 5 primitives « A C B C A ». Pour chacun des épisodes de cette signature, un paramètre statistique a été calculé. 5 valeurs représentant la quantification des paramètres statistiques sont donc obtenues : « 0,4 8,2 233 45,5 -6,7 », il s'agit du vecteur forme. La signature mixte proposée repose sur la combinaison de ces valeurs de paramètres statistiques et des primitives obtenues.

A partir de la signature mixte d'une forme, il est possible de calculer une nouvelle mesure de similarité basée sur celle de l'équation (3.1). Cette mesure est uniquement définie pour mesurer la similarité mixte d'une forme dynamique à une classe. La signature mixte représentative d'une classe est composée de la signature structurelle de cette classe et de la valeur statistique moyenne de chacun des épisodes obtenus pour les formes dynamiques de cette classe. Cette mesure de similarité mixte utilise à la fois la mesure de similarité structurelle (3.1) et la mesure de similarité statistique sim_{stat} définie entre deux signatures mixtes $Sigm1$ (d'une forme dynamique) et $SigmC$ (d'une classe) comme :

$$sim_{stat}^i(Sigm1, SigmC) = \begin{cases} 1 & si \ s_i^{Sigm1} = s_i^{SigmC}, \\ 0 & si \ |s_i^{SigmC} - s_i^{Sigm1}| > \sigma(C_i), \\ \frac{|s_i^{SigmC} - s_i^{Sigm1}|}{\sigma(C_i)} & si \ |s_i^{SigmC} - s_i^{Sigm1}| \leq \sigma(C_i), \end{cases} \quad \text{où } i = 1..n,$$

$$sim_{stat}(Sigm1, SigmC) = \frac{\sum_{i=1}^n sim_{stat}^i(Sigm1, SigmC)}{n} \quad (3.3)$$

avec s_i^{Sigm1} définie comme la valeur du paramètre statistique de l'épisode i de la signature mixte $Sigm1$ et $\sigma(C_i)$ est l'écart type de la classe par rapport au paramètre i . Puis on définit la valeur de similarité mixte par :

$$\begin{aligned} sim_{mixte} &= \alpha sim_{struct} + \beta sim_{stat}, \\ \alpha + \beta &= 1 \end{aligned} \quad (3.4)$$

où sim_{struct} est la mesure de similarité structurelle définie par (3.1), α représente le poids donné à la mesure de similarité structurelle dans le calcul de la mesure mixte, et β est le poids donné à la mesure de similarité statistique sim_{stat} dans le calcul de la mesure de similarité mixte. Si les deux mesures de similarité ont le même poids, elles ont chacune un poids de 0,5. A l'inverse, on peut choisir d'augmenter le poids d'une mesure de similarité, dans ce cas, il est nécessaire de descendre le poids de l'autre mesure. Cela peut être judicieux de donner un poids différent aux deux mesures, si l'on considère que les résultats sont plus fiables avec une mesure qu'avec l'autre, ou si la précision d'une mesure est plus importante que l'autre. Dans ce dernier cas, cela dépend surtout du nombre d'épisodes utilisés dans l'établissement des signatures.

3.5. Conclusion

Deux approches de classification des formes dynamiques ont été proposées dans ce chapitre. Pour rappel une forme dynamique est définie par une trajectoire représentant son développement temporel durant une certaine période.

Une première approche de classification structurelle repose sur une technique de segmentation qui ne nécessite pas la définition de seuils d'erreur d'approximation mais qui utilise un nombre fixe de primitives. Cette approche cherche récursivement le nombre optimal de primitives le plus adapté à chaque forme. Elle augmente la précision de la segmentation avec le nombre de primitives tant que ces dernières permettent de mieux caractériser la classe de la forme.

La seconde approche concerne un multi-classifieur mixte utilisant l'approche structurelle développée et une méthode de classification statistique adaptée à l'application sur laquelle on choisit d'utiliser le multi-classifieur. Il est d'ailleurs possible d'utiliser plus d'une méthode statistique de classification si c'est nécessaire. Cette approche tient compte des caractéristiques structurelles trouvées sur les formes dynamiques, pour définir les paramètres statistiques complémentaires pouvant aider à mieux discriminer les classes. Les paramètres statistiques sont issus des épisodes trouvés par la méthode structurelle, puis la classification

de chaque forme à l'aide des deux méthodes permet d'obtenir deux résultats de classification pour chaque forme. Une règle de décision globale est alors utilisée en tenant compte des résultats des deux classifieurs pour définir la classe de chaque forme.

A l'aide des résultats obtenus par le multi-classifieur, une signature mixte, i.e. composée de la signature structurelle et du vecteur forme statistique, proposée permet d'interpréter les caractéristiques de chaque forme.

Une mesure de similarité mixte, basée à la fois sur la similarité entre caractéristiques structurelles et statistiques, est ensuite calculée à partir de cette signature mixte pour mesurer l'appartenance d'une forme à une classe. Cette mesure permet également de quantifier l'évolution que peut réaliser une forme suite à un changement de caractéristiques d'un système. En effet, si l'on possède des formes dynamiques pré et post changement des caractéristiques d'un système, il est possible de mesurer la similarité de chacune de ces formes à la classe représentant le mode de fonctionnement du système. Si l'écart entre ces deux valeurs de similarité augmente post changement cela signifie que la forme fait davantage partie de sa classe (évolution) sinon qu'elle appartient moins à sa classe (régression).

Ces approches de reconnaissance des formes permettent d'automatiser la classification des données et ont pour but d'améliorer le taux de classification une fois le paramétrage de la méthode réalisé. Elles peuvent aider à la caractérisation des formes dynamiques et à l'interprétabilité des données par les cliniciens à l'aide de la signature mixte proposée. Toutefois, les méthodes proposées ont également leurs contraintes. Le nombre maximal de primitives à utiliser dans la méthode de segmentation doit être adapté à chaque application. La mesure de similarité structurelle est floue mais sa précision est limitée au nombre de primitives utilisé pour représenter une signature. Concernant la méthode mixte de classification, il est nécessaire d'obtenir une seule valeur de paramètre(s) statistique(s) pour chaque épisode de la signature d'une forme. Pour ce faire, il est nécessaire de choisir l'opérateur d'agrégation le plus adapté à chaque application quand plusieurs paramètres statistiques sont définis pour un épisode. Cet opérateur d'agrégation peut être choisi par expérimentation ou il peut être défini en fonction des paramètres statistiques utilisés. Il aura une influence sur l'interprétation des résultats puisqu'il modifie la signature mixte d'une forme.

Dans le chapitre suivant, chacune des méthodes de classification proposées dans ce manuscrit (Fuzzy Pattern Matching Dynamique, K Plus Proches Voisins Flous Dynamique-

Supervisée, K Plus Proches Voisins Flous Dynamique-Semi-Supervisée, méthodes structurelle et mixte de classification des formes dynamiques) seront évaluées en utilisant des applications simulées et réelles.

Chapitre 4. Application et validation des approches proposées

Les méthodes de reconnaissance dynamique des formes statiques et les méthodes de reconnaissance des formes dynamiques respectivement développées dans le chapitre 2 et 3 vont être appliquées sur plusieurs exemples de simulations et réels représentant des systèmes évolutifs. Ces systèmes touchent aux domaines de l'industrie (machines de soudures, générateur de vapeur des Réacteurs à Neutrons Rapides (RNR) Prototype Fast Reactor (PFR)) ou au domaine médical (la marche chez les patients hémiparétiques). Ces systèmes présentent des changements de leurs caractéristiques propres qui modifient dans le temps leur comportement. Cette modification de comportement conduit à un passage entre plusieurs modes de fonctionnement normal et/ou défaillant d'une façon progressive ou brutale. Compte tenu de l'aspect évolutif de ces systèmes, il était nécessaire d'utiliser les méthodes de reconnaissance dynamique des formes afin de mettre à jour la structure et les paramètres des classifieurs utilisés permettant ainsi de conserver leurs performances au cours du temps.

4.1. Application et validation des méthodes proposées de reconnaissance dynamique des formes

Selon le type d'application sur lequel on utilise une méthode de reconnaissance des formes, il est nécessaire de tenir compte de l'ensemble des informations disponibles (ensemble d'apprentissage, le type de données recueillies (formes dynamiques ou non), etc.) et de l'importance des critères d'évaluation (retard de détection, temps de classification, taux d'erreur de classification, etc.). C'est pourquoi dans les sections qui suivent, la méthode la plus adaptée à chaque application a été sélectionnée puis appliquée. Dans les sections suivantes, les applications et leur base de données sont présentées ainsi que les résultats de classification obtenus. De plus, ces exemples d'applications sont complémentaires pour tester avec des critères différents les méthodes développées.

4.1.1. Benchmark d'un système hydraulique à deux réservoirs

4.1.1.a. Présentation du benchmark

La méthode FPMD (Hartert, 2010b) a été appliquée au suivi d'évolution d'une fuite de faible ampleur sur le benchmark d'un système de deux réservoirs (Figure 4.1). Le fonctionnement normal du système est modélisé par l'automate de la Figure 4.2. Pour la configuration du benchmark, nous avons sélectionné une pompe P (P_m pompe en marche, P_a pompe arrêtée), deux conduites Co_2 et Co_3 toujours ouvertes reliant les réservoirs R_1 et R_2 , une vanne V_4 permettant l'écoulement du liquide et une vanne V_1 permettant de simuler les fuites. Chaque réservoir est équipé d'un capteur mesurant le niveau de liquide, l_1 pour R_1 et l_2 pour R_2 .

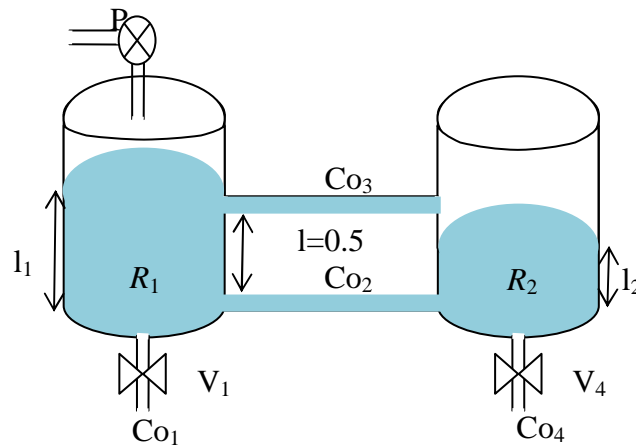


Figure 4.1. Benchmark de deux réservoirs.

Quatre modes de fonctionnement normaux peuvent survenir en fonction du niveau l_1 . Quand ce niveau l_1 est inférieur à un niveau prédéfini l , la pompe sera active tandis que si l_1 est supérieur à l , la pompe sera arrêtée. La valve V_4 sera ouverte ($V_{4\text{-ouverte}}$) ou fermée ($V_{4\text{-fermée}}$) respectivement selon les deux événements commandables e_1 et e_2 générés par le contrôleur en fonction de la valeur de l_2 .

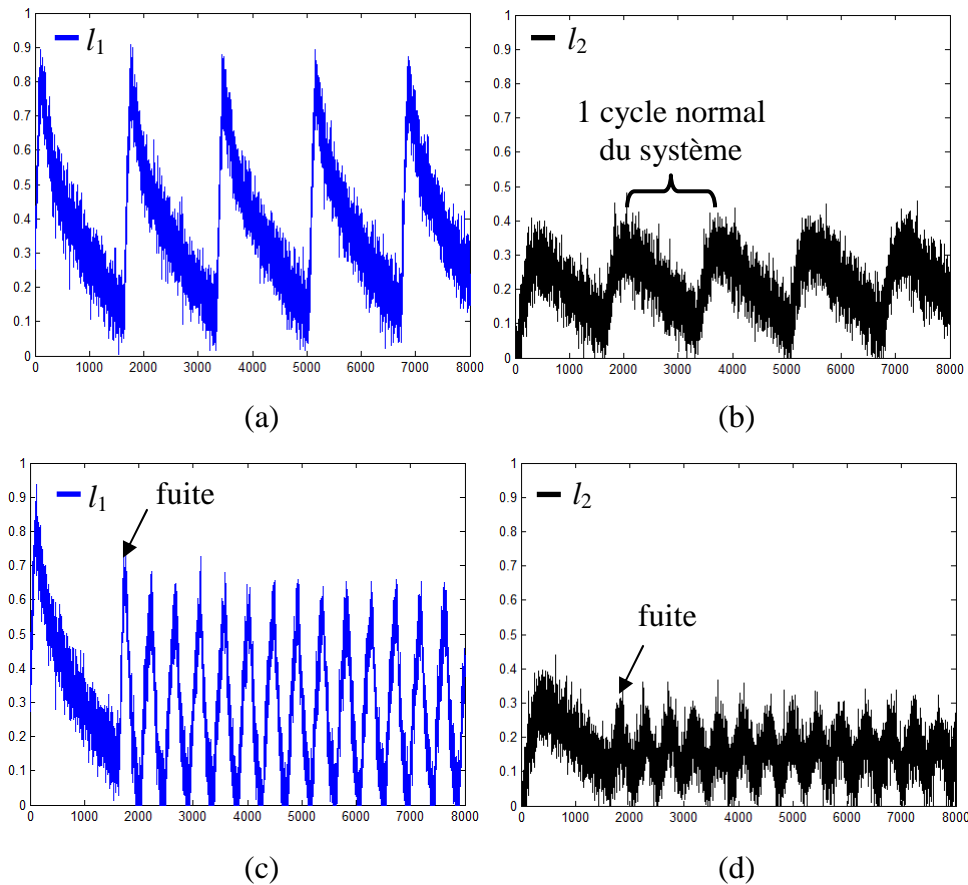


Figure 4.4. (a) (b) représentent respectivement le niveau du réservoir R_1 et R_2 dans un fonctionnement normal tandis que (c) (d) représentent ces niveaux dans le cas d'une faible fuite ayant eu lieu durant le mode 2 de fonctionnement normal.

4.1.1.b. Protocole expérimental

Les premières étapes réalisées sur ce système sont le prétraitement et l'analyse des données. Trois paramètres statistiques discriminatifs ont été extraits de ces données par rapport à chaque signal ; il s'agit du temps entre deux pics maximum (« temps max-max »), de la pente entre deux maxima (« pente ») et de la valeur de chaque maximum (« valeur max ») (Figure 4.5). Le premier paramètre permet de savoir si les temps de remplissage et de vidange des réservoirs sont plus ou moins importants qu'ils ne devraient être. Les deux autres paramètres donnent des informations sur l'importance de la fuite. L'information issue de ces trois paramètres peut être également obtenue, en agrégeant par l'opérateur « max » les deux signaux de niveaux des réservoirs. D'autres opérateurs d'agrégation (min, moyenne, etc.) ont également été testés, mais l'opérateur « max » (Figure 4.6) est l'opérateur le plus simple permettant de conserver les informations des deux signaux. L'intérêt de cette agrégation est de

ne traiter qu'un signal à la place de deux sans perdre d'informations. Le cycle du système (Figure 4.4.b) où la fuite apparaît correspond à une période transitoire spécifique dans laquelle l'ampleur du défaut ne peut pas être détectée. Le retard de détection correspondant à la durée de cette période est pris en considération dans les résultats de l'application de notre méthode.

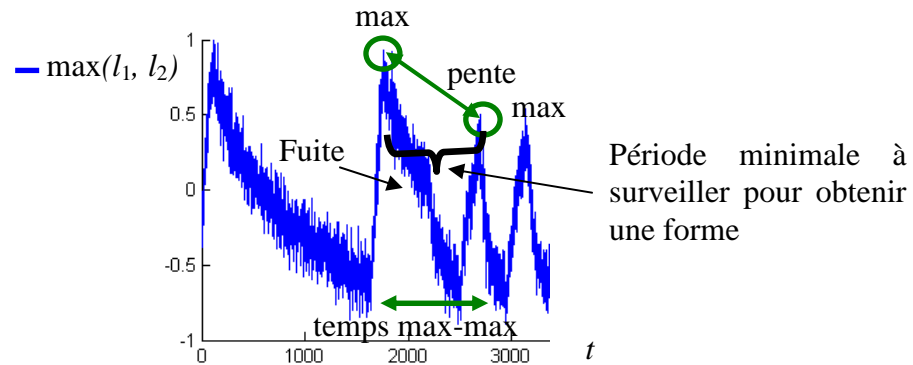


Figure 4.5. Représentation des trois paramètres informatifs obtenus sur une partie de signal.

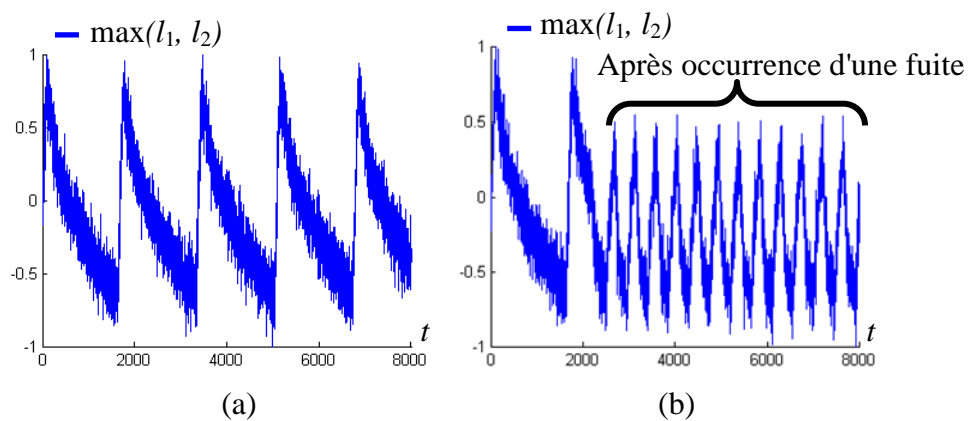


Figure 4.6. Opérateur maximum utilisé pour agréger les signaux correspondant au niveau des deux réservoirs dans le cadre d'un fonctionnement normal (a) et défaillant (b).

L'ampleur des fuites simulées varie en fonction du niveau du réservoir. C'est la vanne V_1 qui est utilisée pour simuler des fuites de différentes amplitudes dans le réservoir R_1 . L'importance des fuites simulées varie selon la taille de la section utilisée (2×10^{-4} , $2,5 \times 10^{-4}$ ou $3 \times 10^{-4} \text{ m}^2$) sur le réservoir R_1 . En utilisant ces sections, nous avons obtenu les classes de la figure 4.7.

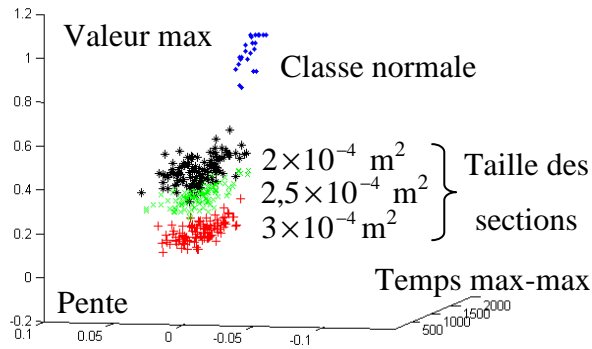


Figure 4.7. Classes trouvées pour le mode de fonctionnement normal et pour les fuites de trois ampleurs différentes.

Les classes de la figure 4.7 doivent permettre de confirmer l'ampleur de la fuite et de suivre son évolution (retour vers le mode normal ou amplification de la fuite). Dans notre cas, nous avons cherché à suivre l'évolution de la fuite dans l'espace de représentation afin d'évaluer le futur niveau de fuite (Figure 4.8). Les classes des figures 4.7 et 4.8 sont composées des différentes mesures réalisées sur plusieurs signaux bruités lors d'un fonctionnement normal ou après occurrence d'une fuite. 24 formes ont été obtenues pour le fonctionnement normal et 110 formes pour chaque classe correspondant à un fonctionnement défaillant.

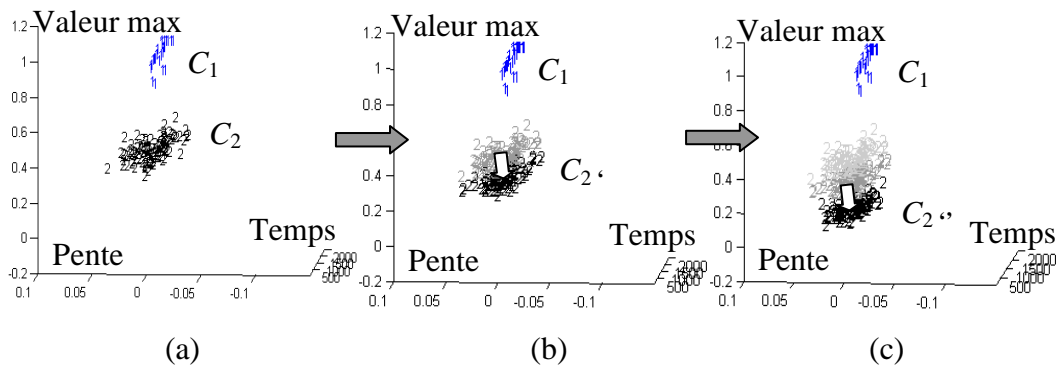


Figure 4.8. Ensemble d'apprentissage utilisé représentant la classe du fonctionnement normal 'C₁' et la classe de fuite de section $2 \times 10^{-4} \text{ m}^2$ (a). Evolution de la taille de la section jusqu'à $2,5 \times 10^{-4} \text{ m}^2$ (b) et $3 \times 10^{-4} \text{ m}^2$ (c).

4.1.1.c. Résultats de classification

La phase d'apprentissage a été réalisée en utilisant les 110 formes représentant une classe C₂ de fuite obtenue avec une section de $2 \times 10^{-4} \text{ m}^2$ et les 24 formes de la classe C₁ de fonctionnement normal. Ces formes ont permis à FPMD de calculer les fonctions

d'appartenance des deux classes. Ensuite, la classification des formes a été réalisée en utilisant deux ensembles de points de test :

- 110 formes correspondant à la première évolution de l'ampleur de fuite d'une section de 2×10^{-4} à $2,5 \times 10^{-4} \text{ m}^2$,
- 110 formes représentant la fuite qui continue d'évoluer jusqu'à la section de $3 \times 10^{-4} \text{ m}^2$.

Le résidu obtenu par FPMD après classification de l'ensemble des nouvelles formes est présenté sur la figure 4.9.

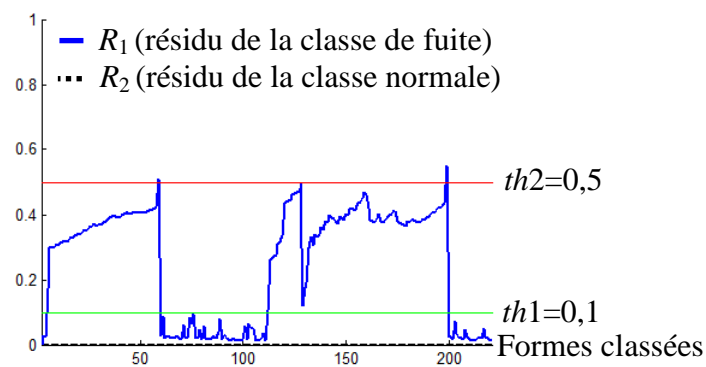


Figure 4.9. Résidu obtenu par FPMD pour l'ensemble des formes évolutives classées.

La figure 4.9 montre qu'une évolution de la classe de fuite de section 2×10^{-4} à $2,5 \times 10^{-4} \text{ m}^2$ est détectée après la classification de 4 formes. Puis les nouvelles formes sont classées dans la classe de fuite, et les formes classées sont jugées représentatives d'une évolution comme le montre le résidu. En effet, un changement de caractéristiques de la classe de fuite apparaît avec ces nouvelles formes qui se sont progressivement déplacées dans l'espace de représentation. Ces nouvelles formes qui sont représentatives d'un changement sont celles qui sont sélectionnées par FPMD.

Après la classification de la 59^{ème} forme de la fuite de section égale à $2,5 \times 10^{-4} \text{ m}^2$, $th2$ est atteint et la classe est adaptée. Afin de trouver les caractéristiques de la classe évoluée, c'est-à-dire son nouveau centre de gravité, un retard de détection correspondant au nombre de formes classées pour réaliser l'adaptation de la classe est nécessaire. Après cette première adaptation, les formes non représentatives sont oubliées, la fonction d'appartenance de la classe est mise à jour et les seules formes de la classe de fuite sont celles classées entre la 4^{ème} et la 59^{ème} forme.

Les formes suivantes sont classées dans la classe jusqu'au début de la seconde évolution détectée après la classification de la 111^{ème} forme. Le résidu atteint *th2* après la 128^{ème} forme et après la 199^{ème} forme. La classe est donc adaptée deux fois. La classe contient finalement 93 formes représentatives du mode de fonctionnement anormal final.

Les étapes réalisées par FPMD sur la classe de fuite sont synthétisées sur le schéma de la figure 4.10.

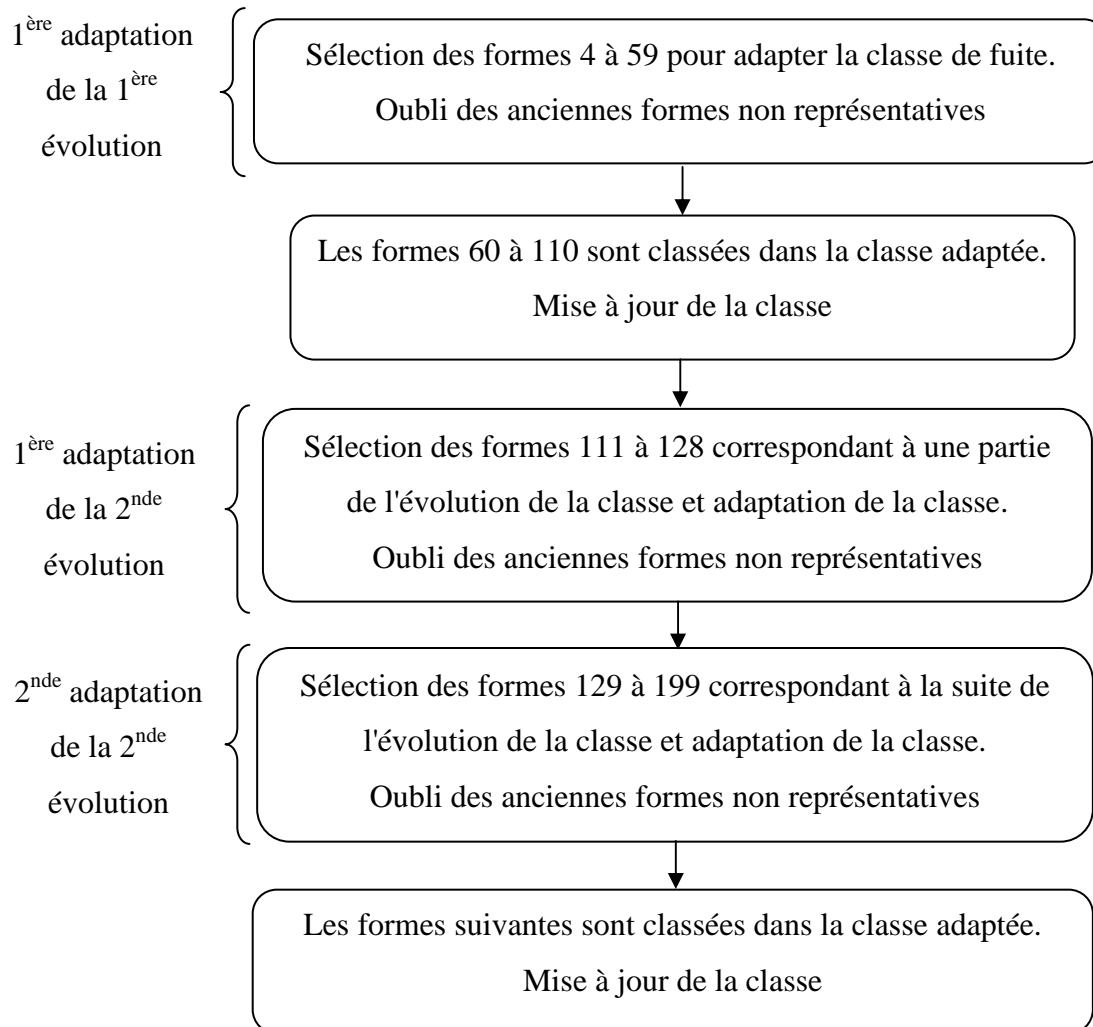


Figure 4.10. Résidu obtenu par FPMD pour l'ensemble des formes évolutives classées.

Les résultats de classification obtenus par la méthode FPMD et par les méthodes incrémentales pour ce cas sont présentés dans le tableau 4.1.

Tableau 4.1. Résultats de classification obtenus par FPMD ($h=5$, $th1=0,1$ et $th2=0,5$), FPMI ($h=5$), SVMi (noyau Gaussien avec variance=10 et constante de régularisation=5) et KPPVI ($k=5$).

	FPMD supervisée	FPMI	SVMi	KPPVI
ind_U pour la 1 ^{ère} évolution	100 %	50 %	50 %	50 %
ind_U pour la 2 ^{ème} évolution	100 %	33,3 %	33,3 %	33,3 %
t_{retard} de la 1 ^{ère} évolution	$t_{retard}=4$ formes	Pas de détection des évolutions	Pas de détection des évolutions	Pas de détection des évolutions
t_{retard} de la 2 ^{ème} évolution	$t_{retard}=1$ forme	Pas de détection des évolutions	Pas de détection des évolutions	Pas de détection des évolutions

4.1.2. Système de commutation

4.1.2.a. Le système

Ce cas décrit un système évolutif réalisant une commutation entre plusieurs modes de fonctionnement. Le système est initialement dans le mode 1, puis il évolue vers le mode 3 en passant une brève période dans un mode de fonctionnement transitoire (mode 2) chevauchant les deux autres modes (Figure 4.11). Le classifieur dynamique doit détecter une évolution du système et doit estimer les paramètres du nouveau mode de fonctionnement. Cela pourrait ainsi permettre de bien classifier les nouvelles formes en fonction des conditions d'évolution de l'environnement du système.

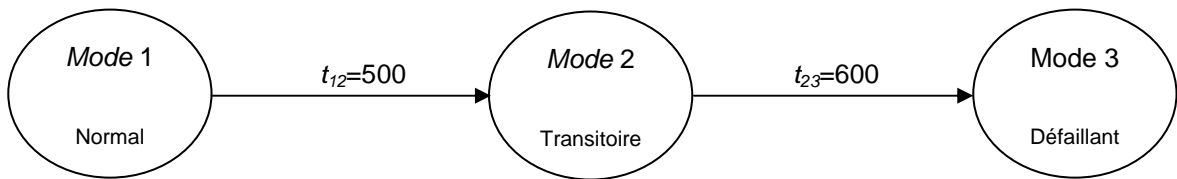


Figure 4.11. Système évolutif commutant à travers trois modes de fonctionnement.

Les trois modes de ce système sont décrits par les fonctions de transferts discrètes suivantes:

$$F_1(z) = \frac{z+0.9}{z^4 - 0.8z^3 + 0.13z^2 - 0.16z + 0.45}, 0 < t \leq 500 \quad (4.1)$$

$$F_2(z) = \frac{z+0.8}{z^4 - 0.7z^3 + 0.2z^2 + 0.3z + 0.15}, 500 < t \leq 600 \quad (4.2)$$

$$F_3(z) = \frac{z + 0.8}{z^4 - 0.5z^3 + 0.3z^2 + 0.4z + 0.05}, 600 < t \leq 1000 \quad (4.3)$$

La période d'échantillonnage du signal de sortie obtenu est de 0,1 unité de temps (Figure 4.12). La sortie contient 10000 points et le signal est généré en utilisant une séquence binaire pseudo aléatoire comme entrée.

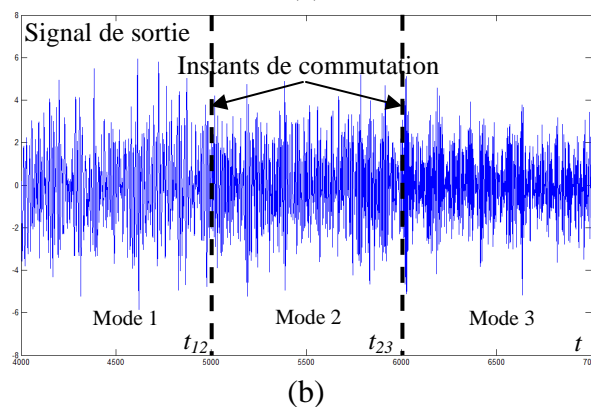
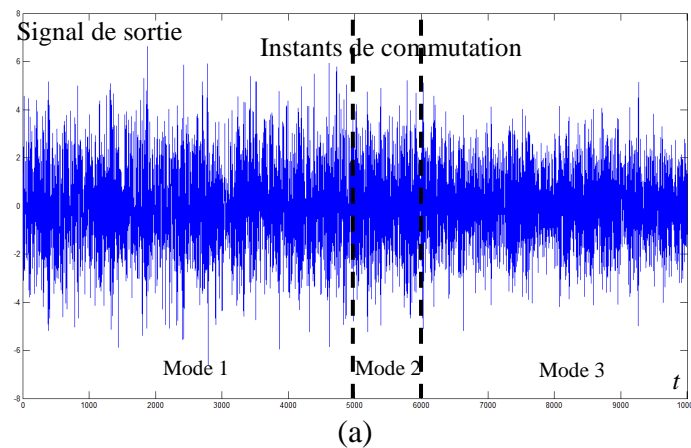


Figure 4.12. Signal de sortie (a) et zoom sur les parties représentatives du changement entre les modes de fonctionnement (b).

4.1.2.b. Recherche des paramètres discriminatifs

Dans le but de construire un classifieur dynamique efficace, les paramètres caractéristiques des modes de fonctionnement doivent être extraits et sélectionnés. Le but est de détecter aussi rapidement que possible un changement dans les conditions des modes de fonctionnement du système et d'estimer correctement ses caractéristiques (paramètres de la fonction de transfert calculables avec Matlab). De nombreux paramètres statistiques (spectraux et temporels) ont été calculés. Ces paramètres sont liés à l'énergie du système, l'entropie, le spectre de puissance, l'amplitude du signal, etc. Dans la phase d'extraction des paramètres, nous avons cherché l'ensemble minimal de paramètres non corrélés en utilisant la méthode d'Analyse en

Composante Principale (ACP) (Jolliffe, 2002). Ensuite, nous avons sélectionné à partir de cet ensemble minimal, les paramètres qui permettent d'obtenir des classes séparées et compactes. Dans le cadre de cette application, les paramètres spectraux n'ont pas permis d'obtenir un espace de représentation discriminant, par contre trois paramètres statistiques ont été conservés :

- la valeur minimale en amplitude, caractéristique de l'évolution du signal,
- la moyenne quadratique (Root Mean Square (RMS)), mesure statistique caractéristique du changement de l'énergie du signal,
- le nombre de passages par zéro (Zero Crossing Rate ou ZCR) utilisé de façon fréquente pour discriminer les données dans des signaux acoustiques. Ce paramètre est souvent utilisé dans le cadre de la reconnaissance de la parole.

Les propriétés statistiques du signal de sortie de ce système évolutif changent avec le temps. Afin de détecter ce changement, les valeurs des paramètres sélectionnés sont calculées pendant une fenêtre glissante. Cependant, cette fenêtre doit inclure suffisamment de points pour estimer correctement les paramètres de chaque mode. Nous avons testé plusieurs tailles de fenêtre. La valeur retenue pour la largeur d'une fenêtre est égale à 250 points ; c'est celle qui maximise le pouvoir de discrimination entre les différents modes de l'espace de représentation. Le pas de la fenêtre est choisi comme égal à la moitié de la taille de la fenêtre. Ce chevauchement entre deux fenêtres consécutives permet de considérer le lien entre elles. Ce lien est fort quand les données appartiennent au même mode de fonctionnement tandis qu'il devient faible quand elles appartiennent à deux modes différents. Toutefois, en chevauchant les fenêtres, la moitié des données d'une fenêtre est utilisée une deuxième fois par la fenêtre suivante. Le chevauchement des fenêtres peut donc être utilisé particulièrement sur les applications pour lesquelles le changement de caractéristiques peut être brutal et surtout puisqu'il implique le traitement et la classification de plus de données, il doit laisser suffisamment de temps à la méthode pour classifier chaque nouvelle forme avant que d'autres formes n'apparaissent. La figure 4.13 montre les formes obtenues sur le signal de sortie de la figure 4.12 dans l'espace de représentation formé par les trois attributs conservés.

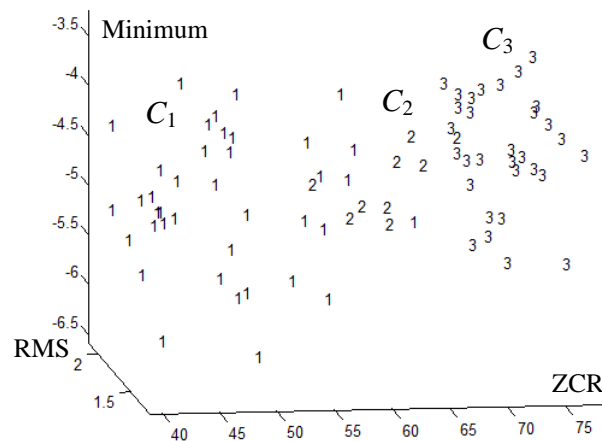


Figure 4.13. Formes appartenant aux trois modes de fonctionnement du système.

Les 79 formes de la figure 4.13 correspondent aux modes de fonctionnement du système en fonction de chaque fenêtre de données. Leur classe correspondante est :

- C_1 pour les formes allant de la 1^{ère} jusqu'à la 39^{ème} fenêtre (39 formes),
- C_2 pour les formes allant de la 40^{ème} jusqu'à la 48^{ème} fenêtre (9 formes),
- C_3 pour les formes allant de la 49^{ème} jusqu'à la 79^{ème} fenêtre (31 formes).

4.1.2.c. Résultats de classification

Les 10 premières formes du signal de la figure 4.12 ont été utilisées comme ensemble d'apprentissage par la méthode KPPVFD Supervisée. Puisque toutes ces formes correspondent à la classe C_1 ; seule une classe a été apprise initialement. Les autres formes constituent l'ensemble de test. Seul le mode de fonctionnement normal correspondant à la classe C_1 est connu *a priori*. L'erreur d'estimation de la fonction de transfert est calculée à partir des paramètres de la fonction de transfert calculés par Matlab. Pour KPPVFD-S, une fois l'évolution confirmée, on connaît le moment t_{adapt} où le signal a changé de mode de fonctionnement. Dans ce cas, avec la fonction 'arx' de Matlab on peut calculer les paramètres de la fonction de transfert sur tout le signal jusqu'à t_{adapt} . L'erreur d'estimation de la fonction de transfert est ensuite calculée comme la moyenne des valeurs absolues des écarts entre les paramètres estimés pour les résultats d'une méthode et les paramètres attendus et connus pour chaque mode de fonctionnement.

Les paramètres utilisés par la méthode KPPVFD-S sont : $k=5$, $th1=5$, $NbMin=7$, $th_{Fusion}=0,1$, $n_1=10$ et $n_2=20$. Les paramètres k , et $NbMin$ ont été définis en fonction du faible nombre de formes de l'espace de représentation. $th1$ est défini pour détecter les évolutions progressives de la classe et les valeurs des paramètres n_1 et n_2 sont celles définies par défaut.

Les résultats de classification obtenus par KPPVFD Supervisée sont présentés sur la figure 4.14.

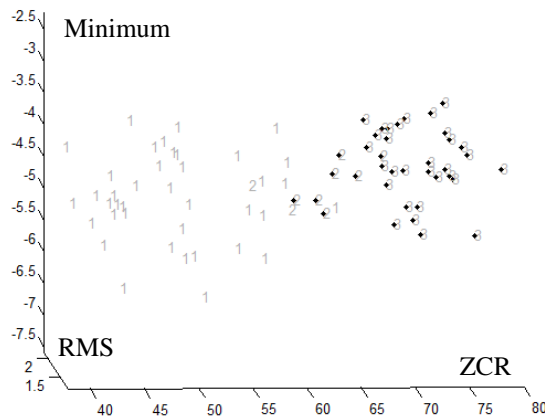


Figure 4.14. Classe finale obtenue par KPPVFD Supervisée. Les formes grisées ont été supprimées après confirmation de l'évolution de la classe.

Une adaptation de la classe a eu lieu avec KPPVFD Supervisée à $t=36$, c'est-à-dire après la classification de la 46^{ème} forme de la figure 4.13 puisque 10 formes ont été utilisées comme ensemble d'apprentissage. Cette évolution validée à $t=36$, signifie que le début de l'évolution de la classe a été détecté à $t=29$ puis qu'il a fallu un nombre $NbMin$ de formes pour valider l'évolution ($t_{Détection}=t_{Validation}-NbMin$). L'évolution de la classe est donc détectée exactement après la classification de toutes les formes de C_1 (39 formes dont les 10 de l'ensemble d'apprentissage).

La classe adaptée à $t=36$ contenait k formes du mode de fonctionnement transitoire qui possède des caractéristiques très proches de celles du mode 3. C'est pourquoi, les formes venant après $t=36$ ont été classées dans la classe adaptée sans détecter de nouvelle évolution de la classe. La classe finale obtenue par la méthode contient une partie des formes de la classe représentant le mode de fonctionnement transitoire et l'ensemble des formes de la classe représentant le fonctionnement anormal (les formes de C_3). La validation de l'évolution a été réalisée avec un retard minimal correspondant au nombre de formes utilisées par $NbMin$ pour confirmer l'évolution de la classe. Les résultats de classification obtenus par KPPVFD-S sont présentés dans le tableau 4.2. Le mode 2 n'est pas représenté dans le tableau puisqu'il représente un mode de fonctionnement transitoire.

Tableau 4.2. Résultats de classification obtenus par KPPVFD Supervisée ($k=5$, $th1=5$, $NbMin=7$, $th_{Fusion}=0,1$, $n_1=10$ et $n_2=20$), FPMI ($h=5$), SVM (noyau Gaussien avec $variance=10$ et constante de régularisation=5) et KPPVI ($k=5$).

	<i>KPPVFD Supervisée</i>	<i>FPMI</i>	<i>SVM</i>	<i>KPPVI</i>
<i>ind_U pour le mode 1</i>	100 %	49,4%	49,4%	49,4%
<i>ind_U pour le mode 3</i>	81,6 %	39,2%	39,2%	39,2%
<i>t_{retard}</i>	Pas de retard de détection de l'évolution	Pas de détection de l'évolution	Pas de détection de l'évolution	Pas de détection de l'évolution

La classe obtenue par la méthode KPPVFD Supervisée correspond deux fois plus à la classe attendue pour le mode 3 que celle obtenue par les méthodes incrémentales. Ce résultat est dû à l'utilisation des seules formes représentatives (ind_U) dans la mise à jour du classifieur. De plus, la méthode permet de détecter et valider l'évolution réalisée par la classe. Ainsi, à partir d'une seule classe connue, la méthode peut prévenir un opérateur qu'un changement dans les caractéristiques de fonctionnement du système a eu lieu. La méthode a mis à jour la classe en fonction des changements temporels des caractéristiques du système. Cependant, une fois les caractéristiques de la classe calculées, c'est à un opérateur que revient la décision de faire correspondre cette classe à un nouveau mode de fonctionnement (mode 3 et non plus mode 1).

Pour confirmer ces résultats de classification, un nouveau signal est généré avec une évolution différente des modes de fonctionnement (Figure 4.15). Ce signal est utilisé pour tester les caractéristiques des classes déjà apprises en utilisant le signal de la figure 4.12. Le nouveau signal commence dans le mode 3 (39 formes), puis il se situe dans le mode 2 (9 formes) et finalement il atteint le mode 1 (31 formes).

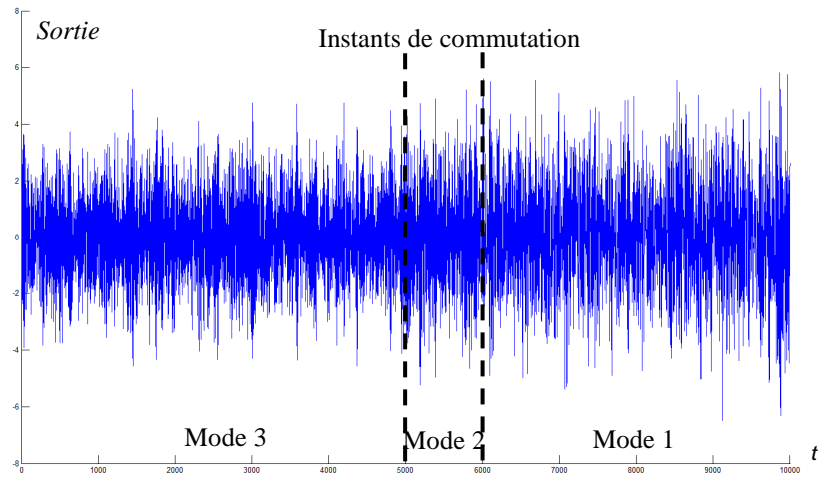


Figure 4.15. Signal de test généré dans un ordre différent de celui utilisé pour l'ensemble d'apprentissage.

La classification des nouvelles formes obtenues sur ce signal est réalisée avec les paramètres de KPPVFD Supervisée définis précédemment (Tableau 4.2). Les résultats de classification obtenus par KPPVFD-S pour ce signal sont présentés dans le tableau 4.3.

Tableau 4.3. Résultats de classification obtenus par KPPVFD-S pour le signal de la figure 4.15 en utilisant les classes trouvées précédemment (Figure 4.14) (Tableau 4.2).

	Temps réel	Résultat obtenu par KPPVFD-S
Temps de détection du mode 3	$t=1$ à 39	$t=1$ à 43
Temps de détection du mode 2	$t=40$ à 48	-
Temps de détection du mode 1	$t=49$ à 79	$t=44$ à 79

Les résultats obtenus pour ce signal dépendent des classes trouvées sur la figure 4.14. Puisque la classe de transition (la classe 2) n'a pas été détectée précédemment, elle n'est pas détectée non plus pour ce nouveau signal. Le temps de détection de chaque mode correspond aux résultats obtenus pour les formes de la figure 4.14. Seules les formes de la classe transitoire sont mal classées. Quatre d'entre elles sont classifiées dans le mode 1 et les cinq autres dans le mode 3. Aucune nouvelle évolution de la classe n'a été détectée comme prévue durant la classification de ces formes.

4.1.3. Suivi de qualité des soudures réalisées par un robot industriel

4.1.3.a. Système industriel de soudure

Cette application concerne le problème de suivi de qualité des soudures d'un système industriel de la société Turquais située à Raucourt-et-Flaba dans les Ardennes (08, France). Cette société réalise la mise en forme de fils et de tubes depuis 1840. Les pièces réalisées sont soumises à des contrôles de qualité très précis pour être utilisées dans le milieu automobile, dans l'équipement électroménager, dans les mobiliers de bureau, dans le domaine de la maroquinerie, etc. Le système de la société Turquais auquel nous nous sommes intéressés est capable de réaliser la soudure de différents types de métaux en quelques secondes. A chaque cycle de soudure, c'est-à-dire après la mise en place de nouvelles pièces par un opérateur humain, le système réalise la soudure de 4 à 6 ensembles de pièces (Figure 4.16).

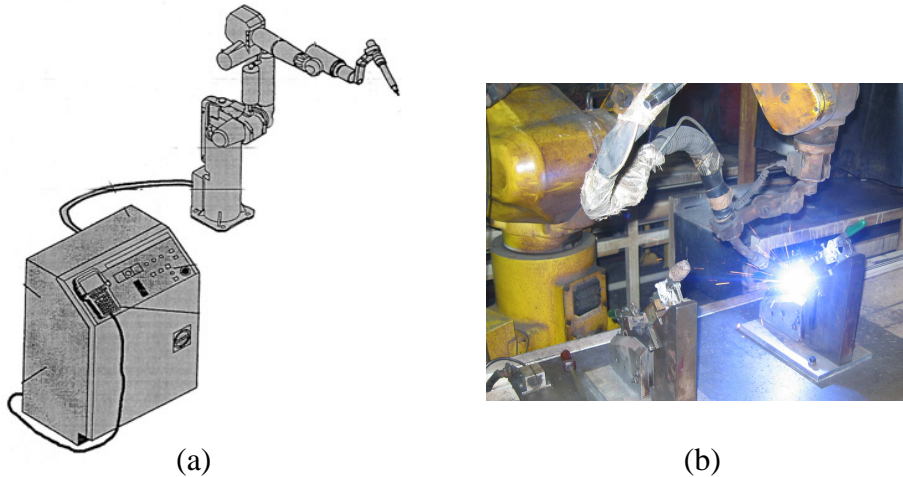


Figure 4.16. (a) le système de soudure surveillé avec son système de contrôle. (b) Soudure de deux pièces métalliques.

En fonction du type de pièces soudées, le nombre de soudures, leurs longueurs et d'autres caractéristiques peuvent changer (Figure 4.17).

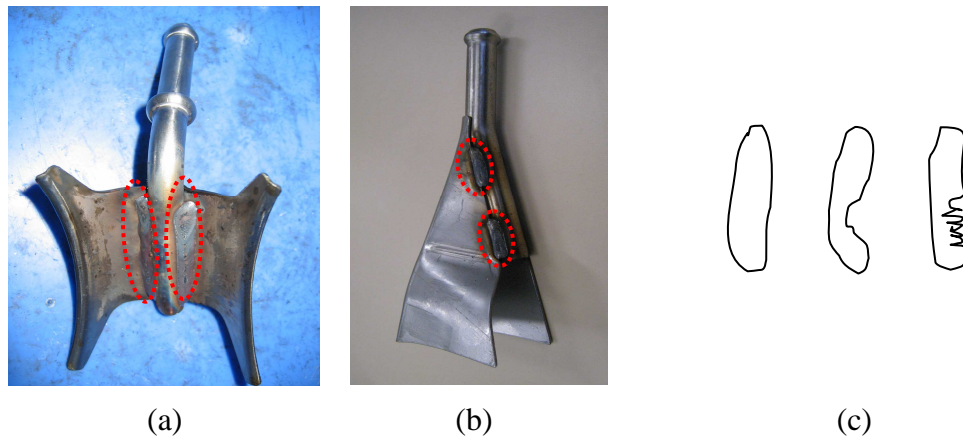


Figure 4.17. Exemple de deux types de pièces soudées (Type 1 – (a), et Type 2 – (b)). Ces deux pièces sont de bonne qualité. (c) Trois représentations schématiques de soudures. De gauche à droite, ces soudures sont respectivement bonne, trop fine et pour la dernière fine et discontinue.

Sur la figure 4.17.c, deux représentations schématiques de deux différentes soudures de mauvaise qualité sont présentées. En effet, quand une soudure est trop fine ou discontinue, elle mène à la création de pièces défectueuses, ce qui n'est pas tolérable. Ainsi, l'intérêt de surveiller ce système est de détecter en ligne toutes les mauvaises soudures afin de corriger les paramètres du système ou de changer une pièce de la machine dès qu'un problème est détecté. L'approche proposée est basée sur l'analyse, l'interprétation et la classification des signaux acoustiques issus des soudures de deux pièces métalliques. De plus, il peut être particulièrement intéressant de suivre l'évolution des soudures, afin de détecter au plus tôt une défaillance, et de voir si le défaut se confirme ou si la soudure redevient rapidement de bonne qualité. Pour cela l'approche proposée doit permettre de détecter l'évolution de la qualité des pièces soudées et un opérateur humain doit être averti si une pièce est considérée comme étant de mauvaise qualité. Actuellement, les opérateurs humains experts de cette machine sont supposés détecter un changement dans la qualité de soudure à partir des sons qu'ils entendent. Pour être capable de réaliser le même traitement en utilisant un ordinateur il était nécessaire d'utiliser cette information sonore.

4.1.3.b. Protocole expérimental

Initialement, deux microphones ont été positionnés ; un sur le robot de soudure entre 40 et 50 cm de la soudure et un en dehors du périmètre du système de soudure pour acquérir tous les sons de l'environnement et pour les extraire du signal de soudure. Le second microphone ne s'est pas montré d'une grande utilité, un seul microphone posé sur le bras de soudure a donc

été retenu. Le microphone est placé sur le système de soudure, dans la direction de la soudure, pour obtenir un son de soudure plus précis et pour réduire significativement le bruit de l'environnement du système. Le système d'acquisition installé est présenté sur la figure 4.18.

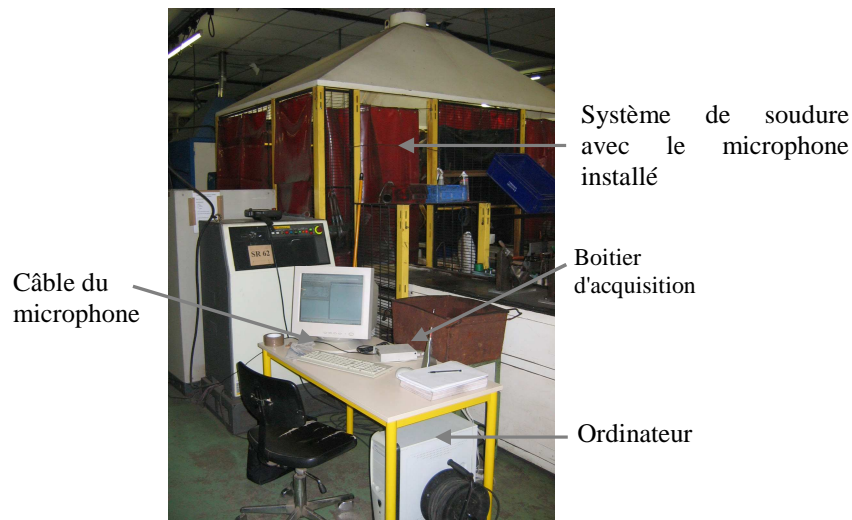


Figure 4.18. Système d'acquisition installé à proximité de la machine de soudure.

Un microphone EM135 (marque LEM) est utilisé pour acquérir les données. Ce microphone est connecté au boîtier d'acquisition SCB-68 (marque National Instruments) en utilisant un câble blindé de 7 mètres résistant aux mouvements du bras de soudure et à certaines perturbations. Puis, le système d'acquisition est connecté à l'ordinateur. La fréquence d'échantillonnage a été fixée pour les premières mesures à 75 KHz (Hartert, 2009b) dans le but de contenir tous les sons qu'un humain peut entendre et pour respecter la loi de Shannon qui impose une fréquence d'échantillonnage au moins deux fois supérieure à la fréquence maximale de l'évènement à étudier. Cependant, cette fréquence d'échantillonnage s'est avérée trop élevée, ce qui se traduisait entre autres par du bruit sur les données recueillies. Par la suite, la fréquence d'échantillonnage a été ramenée à 15 KHz. En effet, l'étude des fréquences intéressantes à surveiller, présentée dans la prochaine section, a montré que ces fréquences ne sont pas très élevées. La base de données est composée de 70 soudures dont 9 de mauvaise qualité. Un exemple de son de soudure de bonne qualité est présenté sur la figure 4.19.

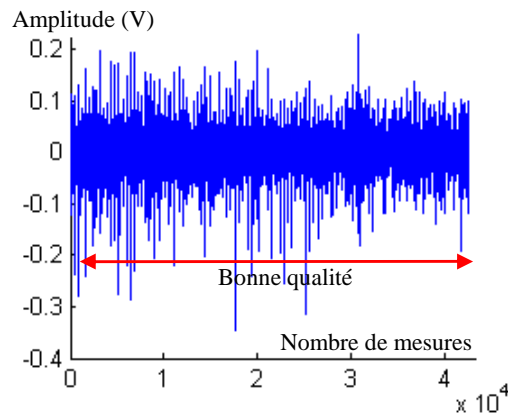


Figure 4.19. Signal acoustique obtenu pour une soudure de bonne qualité.

Il est intéressant de comparer ce signal avec celui d'une mauvaise qualité de soudure (Figure 4.20) afin de voir si des caractéristiques statistiques et/ou structurelles peuvent être discriminatoires.

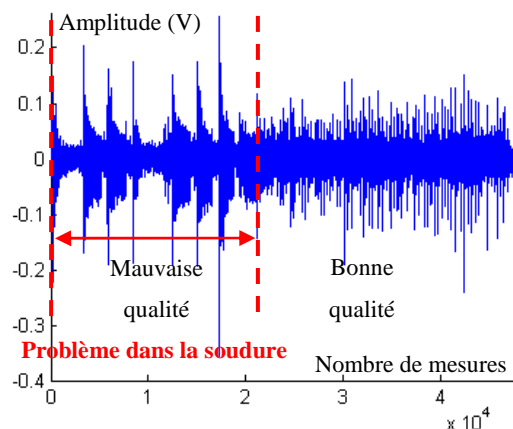


Figure 4.20. Signal acoustique pour une soudure de mauvaise qualité. La partie de soudure de mauvaise qualité apparaît seulement sur une période de quelques secondes, voire millisecondes, dans différentes parties des signaux.

Dans la figure 4.20 le signal semble avoir une partie similaire à celui de la figure 4.19, mais d'autres parties différentes sont également présentes. En effet, sur la figure 4.20 la partie notée 'Mauvaise qualité' concerne l'occurrence de plusieurs défauts de soudure. Ces défauts sont dus aux changements temporels des caractéristiques du système aussi bien qu'à l'usure de ses outils. Cela entraîne un changement dans son mode de fonctionnement qui évolue de normal à défaillant pour produire des pièces de mauvaise qualité. La qualité d'une soudure peut évoluer rapidement de telle façon que même si une partie d'un signal représente une soudure de bonne qualité, une autre peut représenter une soudure de mauvaise qualité. Il est également important de préciser que suite à une période de soudure de mauvaise qualité, la

soudure peut revenir dans un mode de bonne qualité comme le montre la figure 4.20. L'ensemble de ces observations montre la difficulté pour des opérateurs humains de détecter l'ensemble des soudures de mauvaise qualité et surtout l'impossibilité qu'ils ont de détecter l'évolution de la qualité des soudures au cours du temps.

La partie suivante présente la phase d'analyse qui a été réalisée sur ces signaux pour trouver les paramètres informatifs du système qui peuvent être utilisés pour discriminer les qualités de soudures.

4.1.3.c. Phase d'analyse des signaux acquis

Dans un premier temps, nous avons cherché quelle était l'étendue des fréquences utilisées lors de la réalisation d'une soudure. Pour cela, nous avons calculé la densité spectrale de chaque signal (Figure 4.21).

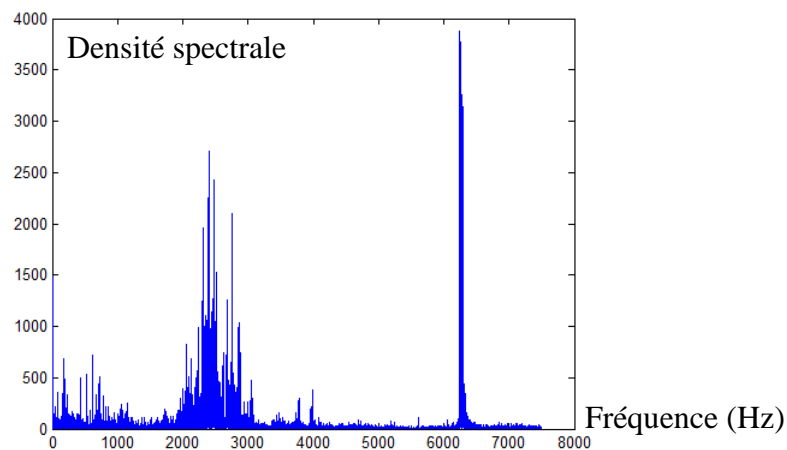


Figure 4.21. Densité spectrale d'un signal acoustique de soudure.

Sur la figure 4.21, on peut voir que certaines bandes de fréquences sont particulièrement présentes dans les signaux, par exemple les bandes de fréquence de 2000 à 4000 Hz et de 6000 à 7000 Hz. D'une manière globale, on peut dire que l'ensemble des fréquences informatives semble être situé en dessous de 7000 Hz, c'est d'ailleurs l'une des raisons qui ont permis de baisser la fréquence d'échantillonnage à 15 KHz. Afin de suivre l'évolution de chaque soudure au cours du temps, nous avons utilisé une fenêtre glissante. Il est important que cette fenêtre contienne suffisamment de formes pour que les informations soient représentatives. Pour cela, nous avons étudié par expérimentation différentes tailles de fenêtres, contenant de 20 à 500 points. Une fenêtre trop petite ne permettait pas de caractériser les modes de fonctionnement du système puisqu'elle ne contenait pas suffisamment de données informatives, tandis qu'une fenêtre trop grande créait un retard de détection des

évolutions et diminuait le taux de classification des données. C'est une fenêtre de 200 points qui a été sélectionnée. Une fenêtre de cette taille s'est avérée suffisamment informative sans engendrer un retard dans le calcul des paramètres de l'espace de représentation. Le pas d'une fenêtre est de 200 points, c'est-à-dire que les fenêtres ne se chevauchent pas mais se suivent. Le chevauchement des fenêtres n'a pas été réalisé puisque suffisamment de formes étaient obtenues par ces fenêtres pour suivre le fonctionnement du système et que compte tenu du temps de classification nécessaire à la méthode, il ne fallait pas que ce chevauchement crée un retard dans la détection des soudures de mauvaise qualité. Pour chacune de ces fenêtres, nous calculons la densité spectrale et plusieurs paramètres statistiques (moyenne, maximum, RMS, Kurtosis, coefficient de dissymétrie, variance, etc.). En ce qui concerne la densité spectrale, il s'est avéré particulièrement intéressant d'étudier les valeurs obtenues par différents paramètres statistiques dans la bande de fréquence 6000-7000 Hz. A partir de l'ensemble de ces paramètres, nous avons sélectionné ceux qui permettaient de discriminer les classes de soudures de bonne et de mauvaise qualité. Deux paramètres ont été retenus pour établir l'espace de représentation :

- la valeur du coefficient de dissymétrie (skewness), notée p_1 , calculée pour la dérivée première de chaque fenêtre de soudure,
- la valeur RMS de la densité spectrale pour la bande de fréquence 6000-7000 Hz, notée p_2 , calculée pour chaque fenêtre de soudure.

A chacune des fenêtres de données correspond une forme dans l'espace de représentation. Sur la figure 4.22, nous représentons un signal de soudure de bonne qualité et les formes qui lui correspondent dans l'espace de représentation.

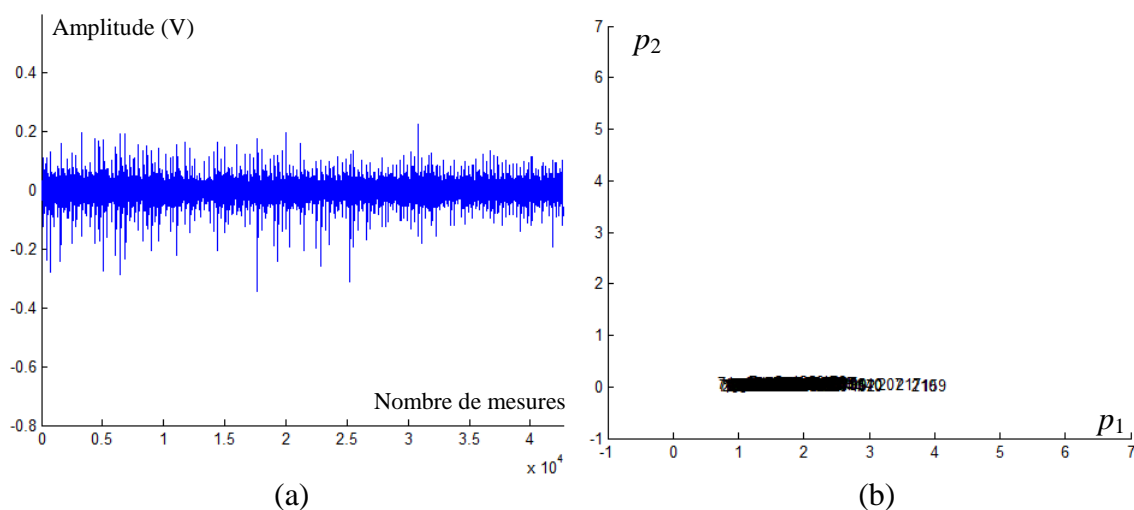
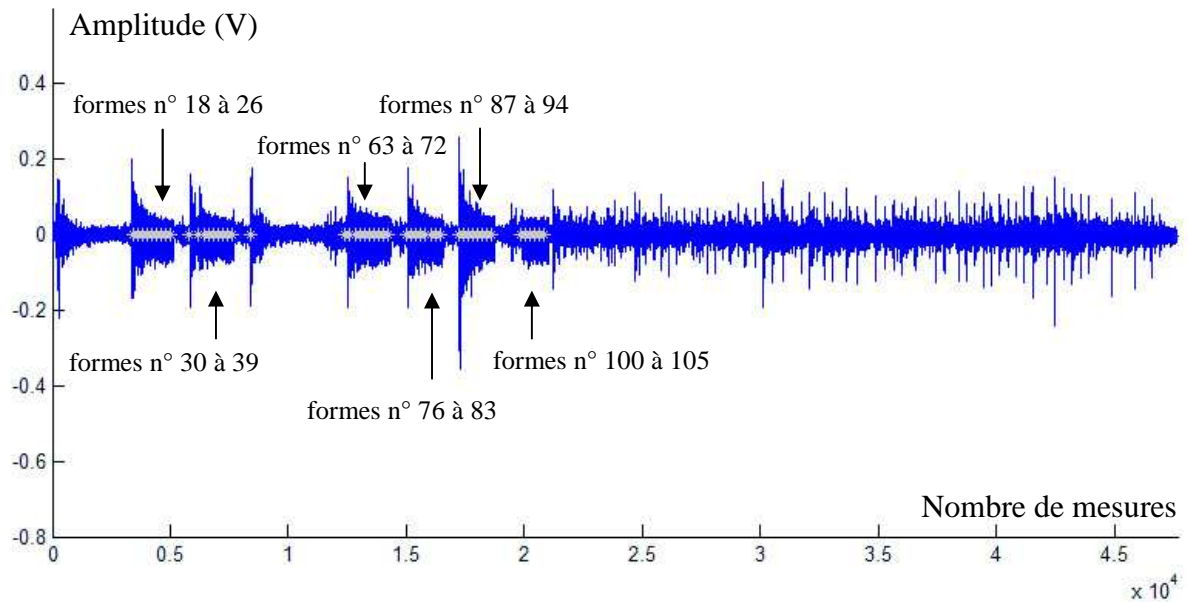
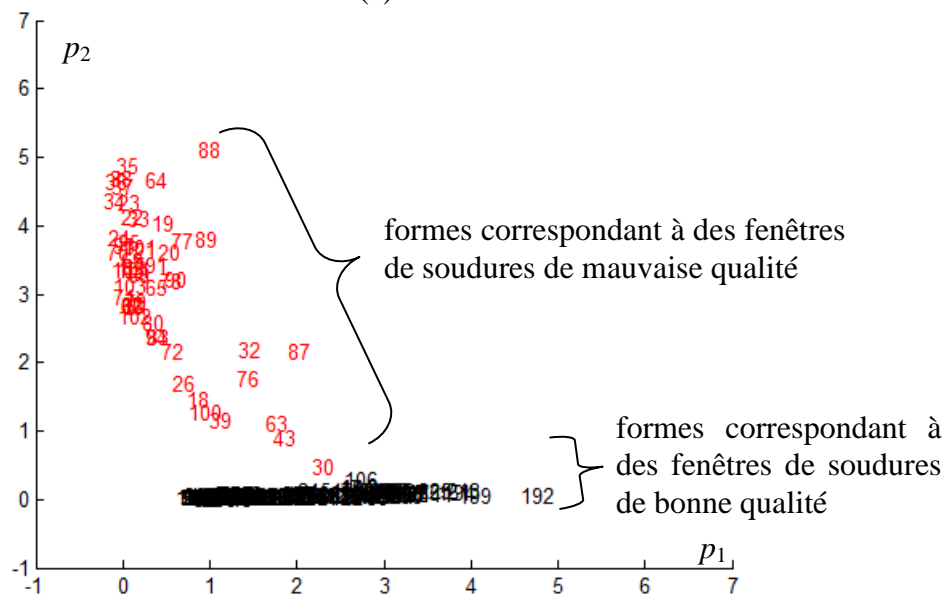


Figure 4.22. Signal de soudure de bonne qualité (a) et les formes qui lui correspondent dans l'espace de représentation (b). Chaque forme est représentée par son numéro de fenêtre.

Sur la figure 4.23.a nous représentons un signal de soudure de mauvaise qualité. Sur cette figure, le début de la fenêtre de chaque forme correspondant à une mauvaise qualité de soudure est représenté par '*'. Les formes correspondant à ce signal sont présentées sur la figure 4.23.b.



(a)



(b)

Figure 4.23. Signal de soudure de mauvaise qualité (a) et les formes lui correspondant dans l'espace de représentation (b).

Sur la figure 4.23, on peut voir les deux classes dont les fonctions d'appartenance devraient être estimées par le classifieur. On peut également constater que certaines formes font la transition entre les fenêtres de soudure de bonne qualité et celles de mauvaise qualité. Par

exemple, la forme de la fenêtre 18 mène le système vers la classe défailiante, tandis que la forme 26 ramène le système vers la classe de fonctionnement normal. Ainsi, lorsqu'une mauvaise qualité de soudure apparaît, plusieurs allers et retours ont lieu entre les deux classes. Le mode de fonctionnement évolue donc en fonction des caractéristiques temporelles et fréquentielles du système. Afin de montrer avec plus de précision l'aller retour des formes, un zoom est réalisé sur une partie du signal et ses formes correspondantes sont présentées sur l'espace de représentation (Figure 4.24).

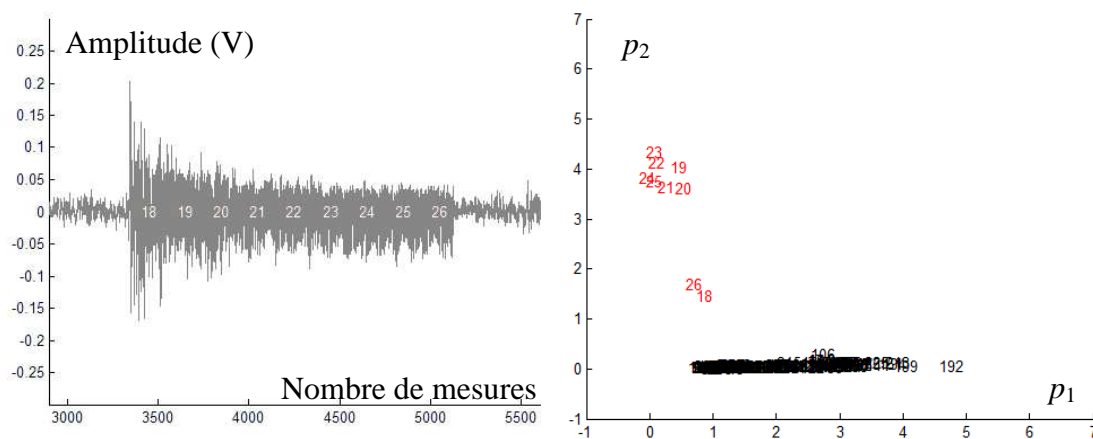


Figure 4.24. Zoom sur les formes correspondant à un problème de soudure et formes correspondantes dans l'espace de représentation. Les formes de la classe de bonne qualité sont également affichées pour donner des repères quand à l'évolution réalisée.

4.1.3.d. Résultats de classification

Dans le but de se mettre dans les mêmes conditions que l'opérateur humain pour l'utilisation de notre méthode, nous ne considérons qu'une seule classe connue ; la classe C_1 qui contient les formes correspondant aux fenêtres de soudures de bonne qualité. Pour la classification de l'ensemble des soudures acquises nous avons utilisé un ensemble d'apprentissage tel que celui de la figure 4.25.

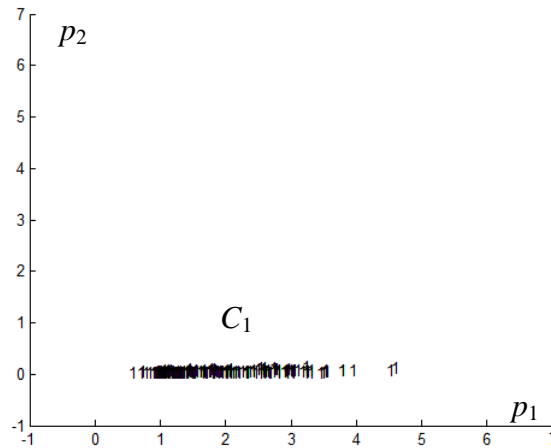


Figure 4.25. Exemple d'ensemble d'apprentissage utilisé pour la classification de nouvelles soudures. 175 formes font partie de C_1 .

A partir de cet ensemble d'apprentissage, nous avons réalisé la classification de chaque soudure, c'est-à-dire la classification de chaque signal acquis sur le système, les unes après les autres. Suite à la classification d'une première soudure de mauvaise qualité, la méthode KPPVFD Semi-Supervisée ($k=5$; $thI=5$; $NbMin=5$; $th_{Fusion}=0,1$; $n_1=10$ et $n_2=400$) nous permet d'obtenir les classes de la figure 4.26.

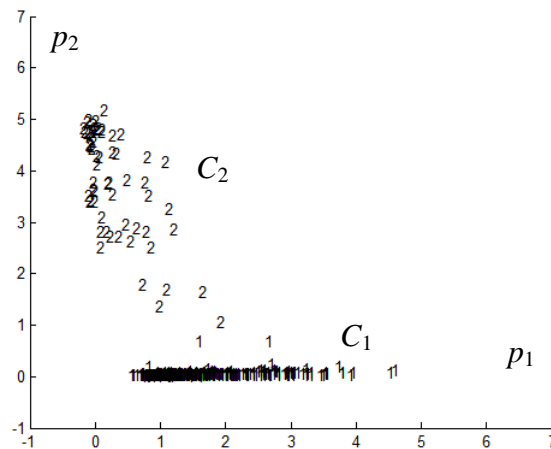


Figure 4.26. Classification de la première soudure de mauvaise qualité.

Suite à la classification de cette première soudure, deux classes ont été identifiées par la méthode. L'évolution de la classe a été validée à $t=299$ alors que l'évolution a réellement débuté à $t=295$. Un retard de détection de $NbMin=5$ fenêtres a eu lieu. Ce retard correspond aux formes qui peuvent être classées dans une classe de transition. Il permet de confirmer l'évolution de la classe C_1 avec un retard faible tout en évitant certaines fausses alarmes qui peuvent avoir lieu avec le bruit présent sur le système. Puis les autres soudures de mauvaise

qualité, provenant des autres signaux acquis, ont également été classifiées. Le résultat de classification de l'ensemble de ces soudures est présenté sur la figure 4.27.

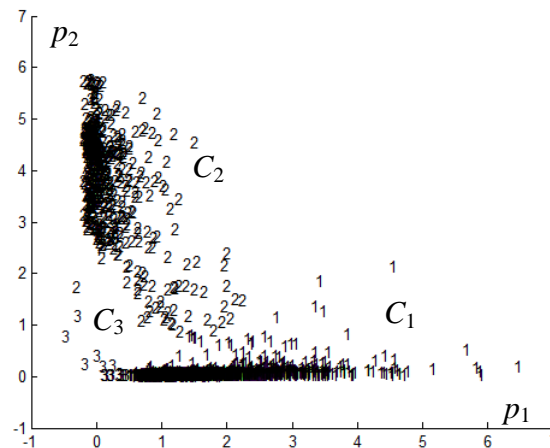


Figure 4.27. Classification de l'ensemble des formes des soudures de mauvaise qualité.

Une nouvelle classe C_3 a été créée, elle correspond à une zone transitoire de qualité de soudure moyenne comprise entre les soudures de bonne qualité et celles de mauvaise qualité. Seules les formes ayant subi un changement de caractéristiques suffisant sont donc classifiées dans la classe C_2 , ce qui permet de ne pas avoir de fausses alarmes. L'ensemble des autres soudures de bonne qualité a ensuite été classé. Toutes les soudures de bonne qualité sont entièrement classifiées dans C_1 (Figure 4.28).

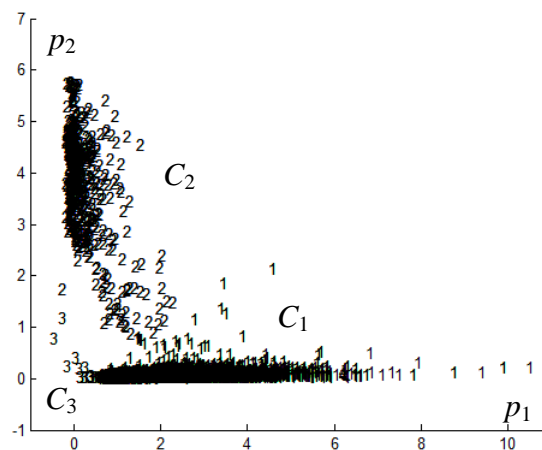


Figure 4.28. Résultat de classification de l'ensemble des soudures de la base de données.

Suite à la classification de l'ensemble de ces soudures, certaines conclusions peuvent être données :

- 100 % des soudures de bonne qualité sont classifiées dans la classe C_1 ,
- 100 % des soudures de mauvaise qualité sont détectées,

- les formes mal classifiées (0,2 % de l'ensemble des formes) correspondent à des formes transitoires. Elles influencent seulement le retard de détection de certaines soudures de mauvaise qualité,
- très peu d'informations (minimum 2 points de C_1) sont nécessaires à la mise en place de la méthode KPPVFD Semi-Supervisée,
- le retard de détection d'une mauvaise soudure est faible (8 millisecondes),
- certaines classes temporaires créées par la méthode ont été supprimées automatiquement durant la classification de l'ensemble des mesures. Ces classes ont été jugées non représentatives puisqu'elles ne contenaient que peu de points et qu'elles n'en recevaient plus suite à leur création. Cette suppression de classes non représentatives permet de maintenir à jour le classifieur en n'utilisant que les classes informatives du comportement du système.

L'ensemble des résultats de classification obtenu par KPPVFD Semi-Supervisée pour la classification des soudures de bonne et mauvaise qualités est résumé dans le tableau 4.4. Les résultats de classification des méthodes incrémentales sont également présentés en utilisant les mêmes conditions expérimentales (ensemble d'apprentissage et formes à classifier).

Tableau 4.4. Résultat de classification de KPPVFD Semi-Supervisée et des méthodes incrémentales FPMI ($h=5$), SVMi (noyau Gaussien avec variance=10 et constante de régularisation=5) et KPPVI ($k=5$).

	<i>KPPVFD Semi- Supervisée</i>	<i>FPMI</i>	<i>SVMi</i>	<i>KPPVI</i>
<i>Détection des mauvaises soudures</i>	100 %	0 %	0 %	0 %
<i>Détection des bonnes soudures</i>	100 %	100 %	100 %	100 %
<i>t_{retard} moyen de détection d'une mauvaise soudure</i>	5 formes	Aucune détection	Aucune détection	Aucune détection
<i>t_{retard} moyen de détection d'une bonne soudure</i>	Pas de retard	Pas de retard	Pas de retard	Pas de retard

L'ensemble des soudures de mauvaise qualité a eu de nombreuses formes classées dans C_2 et C_3 . Seules quelques formes des soudures de mauvaise qualité ont été classées par erreur dans C_1 . Il s'agit des formes ayant engendrées un retard de détection de l'évolution. Ces formes ne sont donc pas mal classées consécutivement, elles correspondent à quelques

premières formes de soudures de mauvaise qualité. A l'inverse aucune soudure de bonne qualité n'a de forme classée dans C_2 et C_3 . Les méthodes incrémentales ont toutes leurs formes classées dans C_1 , elles ne détectent pas l'évolution de la classe.

Ainsi, le résultat de classification obtenu par KPPVFD-SS permet de parfaitement distinguer les soudures de bonne qualité de celles de mauvaise qualité. De plus, seul un faible retard de détection de la qualité d'une pièce a lieu ce qui permet d'appliquer en ligne cette approche à ce système. Un système d'alarme sonore ou visible pourra être mis en place pour prévenir les opérateurs d'un défaut de soudure.

4.1.4. Suivi du fonctionnement d'un générateur de vapeur du Réacteur à Neutrons Rapides (RNR) Prototype Fast Reactor (PFR)

4.1.4.a. Présentation du système

Cette application concerne le développement d'un outil de détection acoustique passive des fuites d'eau du Générateur de Vapeur (GV) du Réacteur à Neutrons Rapides (RNR) Prototype Fast Reactor (PFR). L'objectif de cette approche consiste à détecter les fuites d'eau de plusieurs ampleurs et à des endroits différents avec un temps de retard très réduit. En effet, la réaction produite suite à un contact entre l'eau et le sodium utilisé pour refroidir le corps du réacteur est la cause d'une explosion.

La base de données utilisée est composée d'enregistrements acoustiques de l'injection d'Argon dans le sodium du GV. Ces données sont issues d'expériences réalisées en coopération entre le Commissariat à l'Energie Atomique (CEA) et son homologue britannique l'Atomic Energy Authority (AEA) sur un GV du RNR PFR. Ces expériences ont consisté à simuler des fuites, en injectant, dans le sodium du GV, certaines quantités d'Argon, dans des conditions de pression et de débit variables. Le but de l'utilisation de l'Argon est de calibrer le système de détection acoustique de fuite d'eau. Les injections d'Argon ont été réalisées par l'AEA dans le GV du RNR PFR, à l'occasion de la fin de vie du réacteur, après son arrêt, en juin 1994. L'objectif de cette approche est de caractériser le bruit de la réaction sodium et Argon en fonction de l'importance de la fuite. Cette caractérisation permettrait de définir un outil de détection supplémentaire (à côté de celui déjà utilisé et basé sur la détection de formation des bulles d'hydrogène) afin d'augmenter la fiabilité de détection et de réduire le taux de fausses alarmes.

4.1.4.b. Analyse des données

Une première étape d'analyse fréquentielle et statistique des données a été réalisée pour déterminer les paramètres informatifs et discriminants nécessaires pour définir l'espace de représentation. Puis, la méthode KPPVFD Semi-Supervisée a été appliquée à partir des seules données connues d'une classe de fonctionnement normal pour détecter les nouvelles classes et leurs évolutions. Chaque signal a une fréquence d'échantillonnage égale à 2048 Hz. La figure 4.29 présente un signal acoustique résultant d'une commande d'injection d'Argon dans le GV.

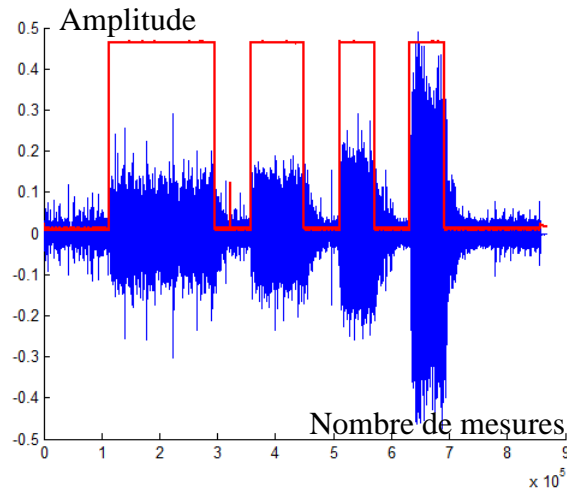


Figure 4.29. Signal acoustique utilisé composé de modes non injection et injection, avec sa commande d'injection représentée par '- '.

Nous avons étudié un nombre important de paramètres temporels, statistiques et fréquentiels. Parmi ceux-ci, nous pouvons citer : la valeur RMS, le Kurtosis, le coefficient de dissymétrie, le facteur crête, la valeur médiane, les coefficients a_i du modèle dynamique AutoRegressif (AR), etc. Ces paramètres ont été calculés dans une fenêtre glissante. La taille de cette fenêtre glissante a été calculée par apprentissage. Cette taille devait être relativement faible pour contenir suffisamment de données sans engendrer un retard de détection important. L'une des méthodes actuelles utilisée sur le générateur de vapeur détecte l'apparition d'un problème en 6s. L'un de nos objectifs était donc de mettre moins de 6s pour détecter l'apparition d'Argon dans le sodium. La fenêtre que nous avons trouvée la plus adaptée à cette application contient 8192 données et se déplace avec un pas de 2048 données. Ce pas de 2048 données correspond à un déplacement d'1s dans le signal, ce qui permet d'obtenir un retard de surveillance faible. L'utilisation de cette fenêtre glissante permet donc de suivre en ligne l'évolution du fonctionnement du GV sur un ensemble de données.

Au niveau des paramètres temporels, nous avons cherché à estimer les coefficients a_i du modèle autorégressif. Pour cela, il nous fallait déterminer l'ordre maximal p du modèle. Nous avons commencé par chercher cet ordre pour une partie des données statiques à l'aide du critère d'information d'Akaike (AIC) (Akaike, 1974) et de l'utilisation de la méthode ACP. Ce critère nous a permis de fixer l'ordre p du modèle réduit à 13. Il nous a ensuite été possible d'estimer ces 13 coefficients pour le modèle. Cependant, notre objectif n'était pas d'estimer le modèle exact du système, nous avons donc choisi les coefficients les plus informatifs parmi les 13 estimés. Pour cela, la méthode ACP nous a montré que deux paramètres du modèle AR, le coefficient a_3 et le coefficient a_5 permettaient d'obtenir une inertie de 94,3%. Ces deux paramètres du modèle AR permettaient d'obtenir l'erreur de classification la plus faible pour les deux classes (injection et non injection). Cette erreur de classification a été calculée en utilisant différentes combinaisons de paramètres avec les méthodes FPM supervisée et SVM. Nous avons également sélectionné un autre paramètre, la moyenne normalisée, qui permettait d'améliorer le résultat de classification.

Comme le montre la figure 4.29, les modes de fonctionnement du système sont amenés à évoluer dès lors que plusieurs fuites se succèdent. L'accumulation des fuites mène alors à une sévérité plus importante du problème. L'utilisation de la méthode KPPVFD Semi-Supervisée doit permettre de suivre ces évolutions et de caractériser chacune des classes évoluées. De plus, très peu d'informations sont disponibles concernant ce système et la méthode KPPVFD Semi-Supervisée n'a besoin de connaître que les informations d'une classe pour être initialisée. En effet, les classes de défaut ne sont pas toutes connues *a priori*.

4.1.4.c. Résultats de classification

La méthode KPPVFD Semi-Supervisée a utilisé comme ensemble d'apprentissage 53 points issus du fonctionnement normal (C_1) du GV. Puis, les formes d'un signal typique (Figure 4.29) ont été classifiées. Les formes et classes correspondant au résultat de classification attendu sont présentées sur la figure 4.30. Le résultat de classification obtenu est présenté sur la figure 4.31.

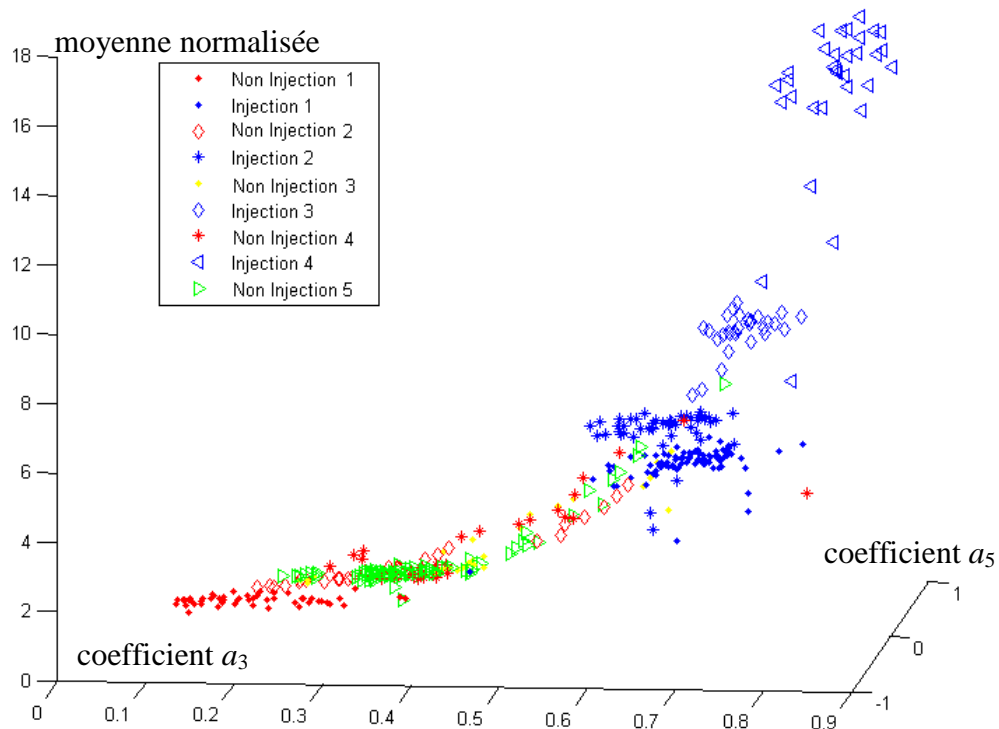


Figure 4.30. Formes correspondants à chaque injection et non injection du signal 4.29.

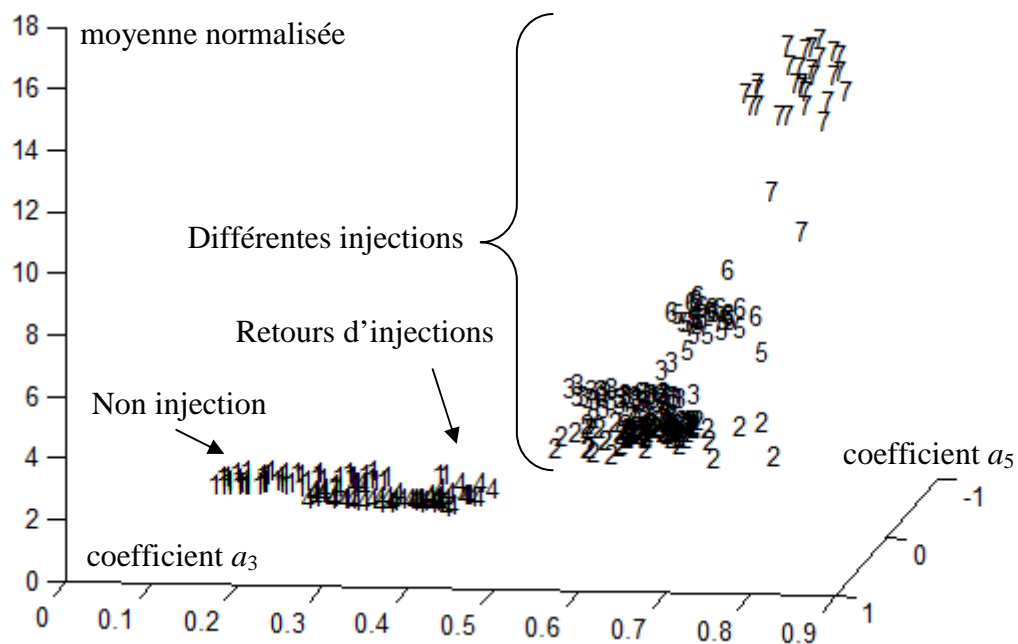


Figure 4.31. Résultat de classification obtenu par KPPVFD Semi-Supervisée après classification de toutes les formes du signal 4.29.

Lors de la classification des formes du signal de la figure 4.26, plusieurs évolutions de classes ont eu lieu. Plusieurs classes ont été créées et les classes obtenues peuvent aider les experts du système à caractériser les différents modes de fonctionnement du système.

Certaines classes de transition, notamment entre C_2 et C_4 (Figure 4.31), ont été jugées non représentatives après un certain temps ; elles ont donc été supprimées automatiquement. Ces classes ne font donc pas partie des classes qui permettent de classifier les formes des autres signaux. Dans le cadre de cette application, les classes trouvées ont permis d'identifier le mode non injection, différents niveaux d'injection (différents niveaux de fuite) et un mode transitoire correspondant au retour au mode non injection suite à une injection. La classe initiale C_1 a conservé ses caractéristiques et les classes obtenues correspondent à celles estimées à l'aide de la commande d'injection (tableau 4.5).

Tableau 4.5. Correspondance des classes trouvées avec les injections et les non injections réalisées. Le retard de détection de l'évolution de chaque classe par la méthode KPPVFD Semi-Supervisée est indiqué.

<i>Numéro d'injection ou de non injection</i>	<i>Classe correspondante trouvée</i>	<i>Retard de détection de l'évolution (en nombre de fenêtres)</i>
Non injection	C_1	0
1 ^{ère} injection	C_2	1
Retour d'injection	C_4	2
2 ^{ème} injection	C_3	1
3 ^{ème} injection	C_5 - C_6	2
4 ^{ème} injection	C_7	2

On peut voir dans le tableau 1 que la détection de l'évolution de chaque classe s'est faite avec un léger retard (maximum 2s). Il est intéressant de voir sur la figure 4.30 que chaque injection est suivie de quelques points transitoires qui se rapprochent progressivement de la classe de retour d'injection. On peut donc dire qu'une fois la classification des points réalisée dans une classe d'injection, les points qui suivent devront être classés dans la classe de retour d'injection.

4.1.4.d. Définition d'un automate pour améliorer le taux de classification

Pour la classification des points contenus dans les zones transitoires de retour d'injection (entre une injection et une non injection), nous définissons un automate en complément de la méthode KPPVFD Semi-Supervisée. L'automate nous permet de suivre le cycle d'évolutions et de commutations des modes de fonctionnement. Il peut être mis à jour en fonction des nouveaux modes de fonctionnement détectés. L'automate utilisé est présenté sur la figure 4.32.

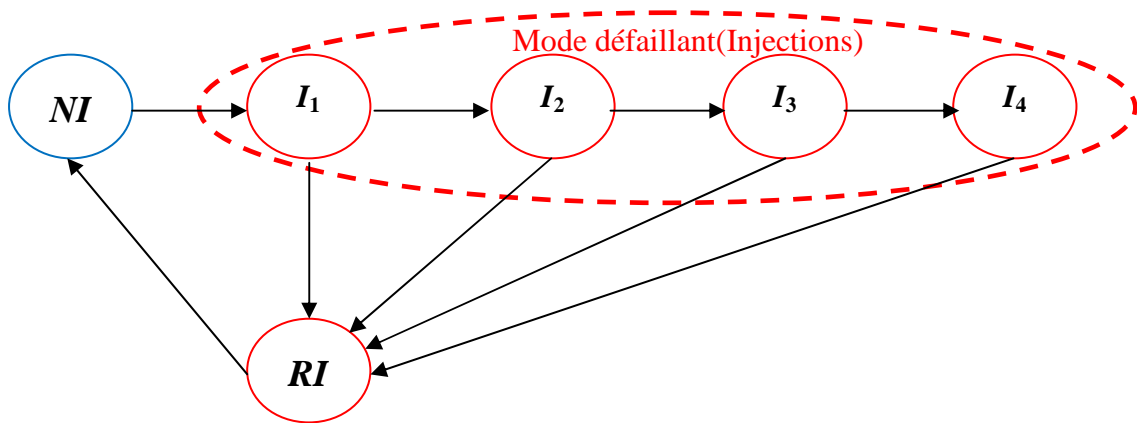


Figure 4.32. Automate utilisé avec KPPVFD Semi-Supervisée pour le suivi des changements de modes de fonctionnement. NI correspond à Non Injection, I_k correspond à la $k^{\text{ème}}$ injection, et RI correspond au retour d'injection.

L'automate défini modélise l'ordre des différents passages que l'on peut avoir entre un mode de fonctionnement normal (Non Injection) et un mode de fonctionnement défaillant (Injection) au cours du fonctionnement du générateur de vapeur. Il modélise aussi les degrés de défaillance que l'on peut avoir quand une fuite d'Argon se présente dans le générateur de vapeur du réacteur PFR. Il faut noter que l'ordre de l'intensité des injections suit l'évolution de cet automate, c'est-à-dire que l'injection I_2 est plus importante que l'injection I_1 , I_3 est supérieure à I_2 et I_4 est supérieure à toutes les injections précédentes. Si une fuite d'Argon se produit dans l'enceinte du GV et que cette dernière est importante (I_4), le système évolue du mode de fonctionnement normal (NI) vers le mode défectueux (I_4), en passant d'abord par les modes (I_1, I_2, I_3).

En tenant compte à la fois des résultats de classification de KPPVFD Semi-Supervisée et de l'automate précédemment défini (Figure 4.32), nous obtenons les résultats de classification du tableau 4.6. A titre de comparaison, les résultats de classification obtenus par les méthodes incrémentales FPMI, SVMi et KPPVI sont également présentés dans le tableau 4.6.

Tableau 4.6. Résultats de classification obtenus par KPPVFD Semi-Supervisée ($k=5$; $th1=5$; $NbMin=6$; $th_{Fusion}=0,2$; $n_1=10$ et $n_2=20$) en coordination avec un automate, par FPMI ($h=5$), par SVM (noyau Gaussien avec $variance=10$ et constante de régularisation=5) et par KPPVI ($k=5$).

	<i>KPPVFD Semi-Supervisée</i>	<i>FPMI</i>	<i>SVM</i>	<i>KPPVI</i>
<i>err_{classif} entre la 1^{ère} non-injection et la 1^{ère} injection</i>	1,1 %	100 %	100 %	100 %
<i>err_{classif} globale</i>	3,2 %	55,9 %	55,9 %	55,9 %

Seules quelques formes sont mal classées avec KPPVFD-SS suite au retard de détection des évolutions des classes (tableau 4.5). Le résultat le plus important est que la première injection est parfaitement détectée. Les formes de cette injection sont bien classées à 98,9 %, seul un retard de détection d'une forme a lieu. A partir de ce résultat il est possible de détecter le début d'un problème sur le générateur de vapeur. On peut voir également dans le tableau 4.6 que le reste des formes des injections et des non injections sont bien classées à 96,8 %, ce qui permet d'utiliser cette méthode en ligne pour le suivi des modes de fonctionnement du GV.

4.2. Application de la méthode mixte proposée pour la reconnaissance des formes dynamiques

Dans cette partie, nous nous intéressons à la classification des formes dynamiques de la coordination inter-segmentaire de la marche chez les patients hémiparétiques. Pour cette application, nous avons choisi d'utiliser une méthode combinant les informations structurelles et statistiques issues des formes dynamiques. La base de données utilisée est tout d'abord présentée puis la méthode mixte de reconnaissance des formes proposée dans le chapitre 3 est appliquée sur cette base de données. Enfin, les résultats de classification sont présentés ainsi que les conclusions tirées de cette démarche.

4.2.1. Caractérisation et interprétation de la coordination inter-segmentaire de la marche des patients hémiparétiques

4.2.1.a. Présentation de l'application

La base de données utilisée concerne les patients hémiparétiques de l'hôpital de Garches (près de Paris). Les patients hémiparétiques présentent des troubles de la marche résultants de

lésions dans leur système nerveux central (Olney, 1996). Les membres du Laboratoire de l'Analyse de la Marche de l'hôpital de Garches utilisent L'Analyse Quantifiée de la Marche (AQM) pour étudier la marche des hémiparétiques. L'AQM clinique fournit des données cinématiques, i.e. des mesures du mouvement, et kinétiques, i.e. des mesures des forces, de chaque articulation des membres inférieurs en 4 dimensions (3 spatiales + 1 temporelle). L'analyse de ces données permet de quantifier et de déterminer les causes principales des troubles de la marche, et par conséquent d'aider le clinicien durant les soins thérapeutiques (Olney, 1996). Cependant, l'interprétation des données quantifiées issues de l'analyse de la marche clinique reste complexe et nécessite beaucoup de temps. En effet, il est difficile d'analyser les relations entre les nombreux paramètres de l'AQM pour caractériser le patron de la marche de sujets humains. C'est pourquoi, la base de données utilisée est limitée aux données de la Phase Relative Continue (CRP) de l'AQM qui est calculée pour chaque patient. La CRP quantifie la coordination entre deux segments (cuisse-jambe, jambe-pied) durant la marche, en corrélant la position de l'angle des segments et la vitesse de chaque segment (Hamill, 1999) (Kelso, 1986). La mesure de la CRP présente deux intérêts principaux : la réduction de la quantité de données et la quantification de la coordination de la marche. La section suivante présente la façon dont la base de données clinique a été établie.

4.2.1.b. Etablissement de la base de données clinique

La base de données clinique concerne la coordination inter-segmentaire (CRP) des patients hémiparétiques durant leur marche. Ces données ont été acquises et calculées à l'hôpital de Garches. Les CRP cuisse-jambe ($CRP_{\text{Cuisse-Jambe}}$) et jambe-pied ($CRP_{\text{Jambe-Pied}}$) ont été mesurées pendant chaque phase de la marche pour 66 patients hémiparétiques n'ayant pas eu de traitement thérapeutique ainsi que pour 27 sujets sains et 14 patients hémiparétiques pré et post traitement.

Lors de l'enregistrement, chaque sujet marche pieds nus à la vitesse qu'il préfère. La marche est analysée en utilisant un système de capture de mouvement (Motion Analysis Corporation, fréquence d'échantillonnage de 100 Hz) et deux capteurs de force (marque AMTI, fréquence d'échantillonnage de 1000 Hz). 29 marqueurs placés sur diverses parties anatomiques ont permis d'obtenir des paramètres spatio-temporels durant la marche. Ces données de marqueurs ont été collectées et filtrées en utilisant un filtre passe bas de Butterworth d'ordre 4, avec une fréquence de coupure de 6 Hz. Pour chaque sujet au moins 10 cycles de la marche ont été moyennés pour obtenir chaque CRP. Le cycle de la marche a été

divisé en quatre phases :

- phase 1 : Double Appui Initial (DAI),
- phase 2 : Simple Appui (SA),
- phase 3 : Pré-Phase Oscillante (PPO),
- phase 4 : Phase Oscillante (PO).

Les angles de chaque segment (cuisse, jambe et pied) ont été calculés dans le plan sagittal avec respect de l'axe horizontal. La vitesse angulaire a été obtenue à partir de la dérivée première de la position de l'angle. Les positions d'angle et les vitesses angulaires normalisées ont ensuite été rééchantillonnées sur 10 % du cycle de la marche pour la première phase, 40 % pour la deuxième phase, 10 % pour la troisième phase et 40 % pour la dernière phase (Figure 4.33).

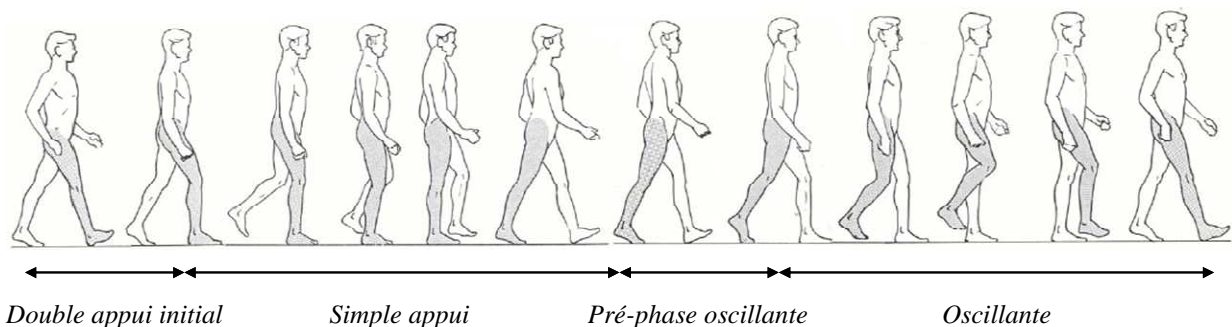


Figure 4.33. Les quatre phases du cycle de la marche humaine.

L'utilisation de la vitesse angulaire et du déplacement angulaire permet d'obtenir l'angle de phase de chaque segment à chaque instant du cycle de la marche (Burgess-Limerick, 1993). Finalement, la CRP est définie comme la différence entre les angles de phase de deux segments à n'importe quel moment du cycle de la marche. L'unité d'une CRP est le degré ($^{\circ}$). Une $CRP_{\text{Cuisse-Jambe}}$ de 0° indique que les segments sont en phase (cas d'une jambe raide sans aucun mouvement du genou). A l'inverse, une $CRP_{\text{Cuisse-Jambe}}$ positive indique que la cuisse 'mène' la jambe.

Les données de chaque CRP ont été normalisées en fonction des valeurs minimales et maximales obtenues pour chaque patient. Cependant, puisque le signe de la CRP représente une information importante du déphasage entre deux segments durant la marche, il a été conservé. Les données ont été normalisées entre -1 et 1. La figure 4.34 présente les $CRP_{\text{Cuisse-Jambe}}$ d'un sujet sain et d'un patient hémiparétique.

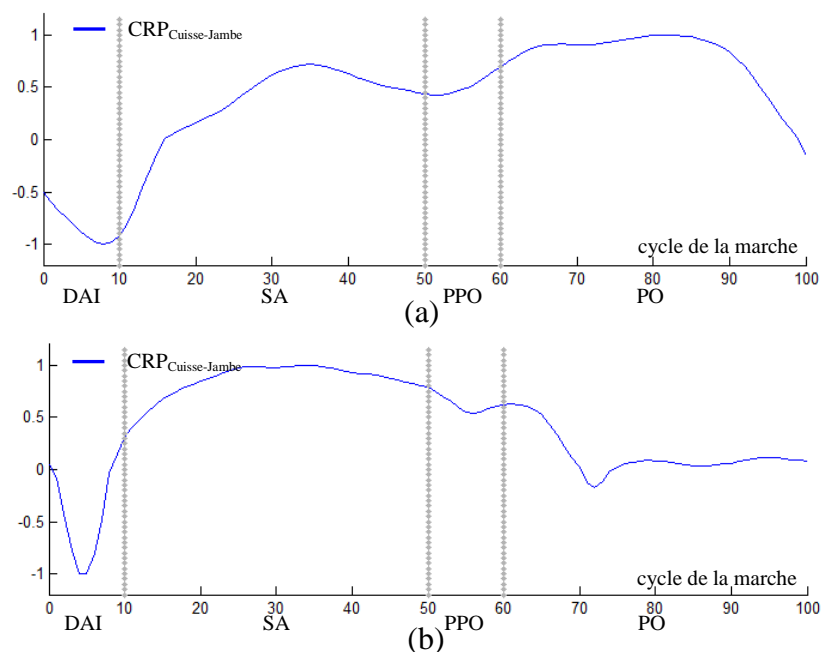


Figure 4.34. $CRP_{Cuisse-Jambe}$ d'un sujet sain (a) et d'un patient hémiparétique (b).

La figure 4.35 présente la deuxième CRP ($CRP_{Jambe-Pied}$) mesurée pour le même sujet sain et pour le même patient hémiparétique que ceux de la figure 4.34.

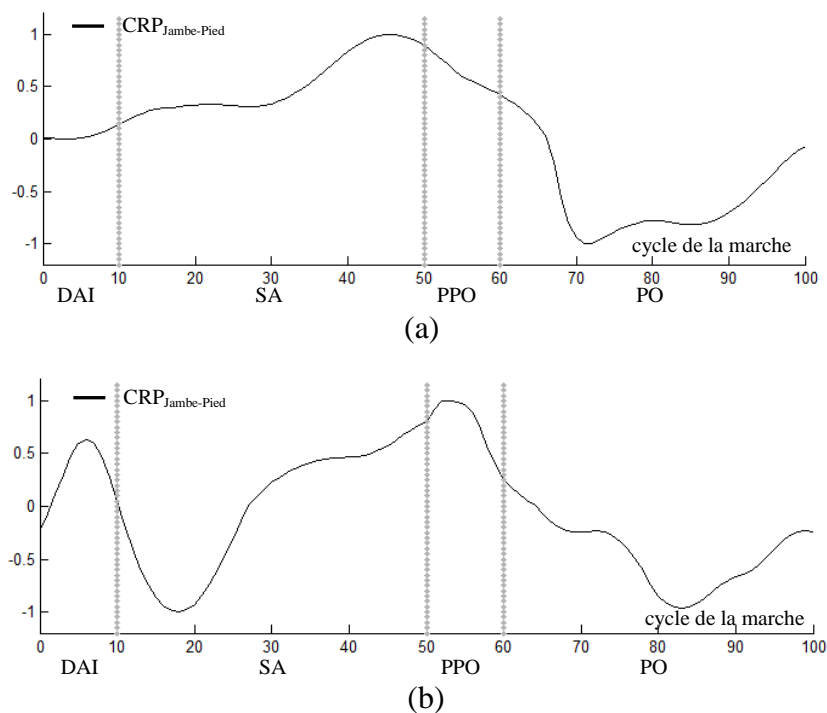


Figure 4.35. $CRP_{Jambe-Pied}$ d'un sujet sain (a) et d'un patient hémiparétique (b).

Les CRP présentées sur les figures 4.34 et 4.35 ne sont pas caractéristiques de tous les patients puisque parmi les 66 patients hémiparétiques de la base de données, plusieurs

souffrent de troubles de la marche différents. Pour conserver les informations interprétables, chaque CRP est utilisée séparément par le multi-classifieur dans la prochaine section.

4.2.1.c. Approches existantes

Dans la littérature, plusieurs méthodes de reconnaissance des formes ont été utilisées pour discriminer les données médicales. Dans (Barton, 1997), un réseau de neurones à rétropropagation à quatre couches est utilisé pour distinguer les patrons de marche en se basant sur des données provenant de marqueurs situés à la hanche et au genou des patients. Trois types différents de réseaux de neurones (rétropropagation standard, gradient conjugué, rétropropagation avec régularisation bayésienne) sont également utilisés dans (Begg, 2006) pour réaliser l'identification des patrons de marche en fonction de l'âge des sujets. La méthode des Séparateurs à Vaste Marge (SVM) est utilisée dans (Lau, 2009) (Mao, 2008) (Kamruzzaman, 2006) pour classifier différents types de marche respectivement en fonction de l'hémiplégie du pied, de la simulation d'un problème de la marche ou de données acquises sur la paralysie cérébrale des patients. Dans (Xu, 2007), c'est l'analyse marginale de Fisher qui est utilisée pour réaliser la classification de la marche humaine en se basant sur des images de marches. Plusieurs autres techniques de classification sont employées sur des données de marche (240 séquences d'images de 20 personnes qui marchent), en utilisant des propriétés statistiques (extraction du contour de chaque personne, etc.) (Rodda, 2001) (Wang, 2003a) ou par extraction de caractéristiques avec l'Analyse en Composante Principale à Noyau (KPCA) (Wu, 2007).

Pour l'application médicale de la caractérisation de la coordination inter-segmentaire de la marche des patients hémiplégiques, nous devons fournir des résultats interprétables aux cliniciens. Il n'est donc pas possible d'utiliser une des méthodes décrites ci-dessus. Ces méthodes réalisent la classification des données médicales, mais elles ne permettent pas d'obtenir un résultat directement interprétable pour les cliniciens.

Pour être capable de réaliser l'interprétation et la classification des formes dynamiques de notre application médicale, nous proposons donc d'utiliser l'approche mixte de reconnaissance des formes dynamiques développée au chapitre 3. Cette approche consiste à utiliser les caractéristiques structurelles et statistiques issues des données de la marche clinique pour discriminer les patrons de coordination de la marche des sujets sains et des patients hémiplégiques. Cette discrimination permettra de caractériser les patients afin

d'identifier les parties les plus caractéristiques de la coordination inter-segmentaire lors de la marche. Puis l'approche proposée est appliquée au suivi d'évolution de la coordination de la marche des patients hémiparétiques ayant suivi un traitement thérapeutique.

4.2.1.d. Résultats de classification

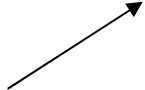

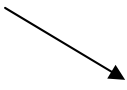
Nous allons commencer par présenter la phase de caractérisation des sujets sains et des patients hémiparétiques ainsi que la phase de classification. Pour cela, le multi-classifieur mixte proposé utilise les informations liées à la structure et aux propriétés statistiques issues des CRP de chaque patient. Chaque CRP est analysée en détails dans chacune de ces quatre phases. En effet, l'interprétabilité des résultats devant se faire par rapport aux 4 phases de la marche, il était important de conserver cette décomposition.

Résultats du classifieur structurel :

La méthode structurelle de classification utilise un nombre de primitives défini pour chaque phase du cycle de la marche. Ce nombre est fixé en fonction de la longueur de chaque phase. Le nombre de primitives maximum N_{max} défini pour les phases 1 et 3 de chaque CRP est de trois, tandis que les phases 2 et 4 peuvent avoir jusqu'à cinq primitives compte tenu de la structure des formes dans ces phases et de leur longueur qui est plus importante. Ce nombre N_{max} a été défini par expérimentation. C'est ensuite l'utilisation de la méthode de segmentation qui nous permet de déterminer le nombre optimal de primitives à utiliser pour chaque phase. Le nombre de primitives retenu est celui qui permet de représenter chaque phase avec une signature unique pour l'ensemble des sujets sains, tout en utilisant un maximum de primitives pour chaque phase. Pour cette application nous avons pour objectif d'obtenir une signature unique de classe afin de pouvoir caractériser l'ensemble des sujets faisant partie de la même classe. Si aucun nombre de primitives ne permet de caractériser la coordination inter-segmentaire de la marche des sujets sains dans une phase, alors la phase est considérée comme non discriminante et elle n'est pas utilisée. Les primitives utilisées sont au nombre de trois (Tableau 4.7). Elles sont du 1^{er} ordre puisque les données ne sont pas bruitées et qu'il ne fallait pas une précision importante afin d'obtenir un résultat de classification interprétable. Les primitives *A* et *C* correspondent respectivement à un épisode croissant et décroissant. Elles permettent de distinguer le segment de jambe qui mène de plus en plus un autre segment, i.e. le mouvement est accentué. La primitive *B* montre que l'angle de CRP entre deux segments de jambe reste le même au cours du temps. Cela signifie qu'il n'y a pas de mouvement entre les deux segments de jambe. Compte tenu de l'interprétation qu'il est

possible de faire concernant la coordination de la marche d'un patient, il n'était pas nécessaire d'utiliser d'autres types de primitives.

Tableau 4.7. Les trois primitives sélectionnées.

<i>Episode</i>			
<i>Primitive</i>	A	B	C

Dans le cadre de l'application clinique, deux classes seulement nous intéressent, celle de la coordination inter-segmentaire de la marche des sujets sains et celle des patients hémiparétiques. Ainsi, il nous faut caractériser la classe des sujets sains avec un maximum de précision pour réussir à discriminer les patients hémiparétiques. Nous avons choisi de caractériser la classe des sujets sains puisque ce sont les sujets dont le comportement reste le plus similaire entre eux. L'utilisation des paramètres et primitives obtenus pour la méthode structurelle a permis d'obtenir une unique signature pour les sujets sains (Tableau 4.8) et plusieurs signatures pour les patients hémiparétiques. Ce résultat est normal puisque chaque patient présente des troubles de la marche différents de ceux des autres patients. Le nombre de primitives utilisées pour chaque phase est :

- 1 primitive pour la phase de simple appui, pour la pré-phase oscillante et pour la phase oscillante de la CRP_{Cuisse-Jambe},
- 1 primitive pour la phase de simple appui de la CRP_{Jambe-Pied},
- 2 primitives pour la pré-phase oscillante et pour la phase oscillante de la CRP_{Jambe-Pied}.

La première phase de chaque CRP a été considérée comme non représentative de la coordination inter-segmentaire de la marche des sujets sains puisqu'elle ne permettait pas de les discriminer. Une fois la signature obtenue pour chacun des deux segments de jambe, nous avons regroupé les deux signatures obtenues. Dans le tableau 4.8, on peut voir que plusieurs primitives identiques se succèdent (exemple : A A ou C C). Cela signifie que plusieurs épisodes successifs ont la même tendance. On ne peut pas fusionner (ou regrouper) ces primitives en une seule, sinon on perd la précision que nous a fourni la signature. En effet, les signatures obtenues pour chaque patient contiennent des primitives qui sont propres aux phases du cycle de la marche. Par exemple, on peut avoir deux signatures A A C et A C C qui regroupées seront identiques (A C) alors que non regroupées elles montrent que les patients n'ont pas la même signature et donc que leur primitives ne sont pas les mêmes par rapport à

chaque phase du cycle de la marche. On peut d'ailleurs rappeler que le principe des méthodes structurelles de classification est de tenir compte du type des primitives mais également de leur ordre. La place de chaque primitive est donc aussi capitale que la primitive elle-même.

Tableau 4.8. Signature caractéristique obtenue pour chaque CRP des sujets sains et présentation de la signature globale obtenue.

<i>Signature de la CRP_{Cuisse-Jambe}</i>	A A C
<i>Signature de la CRP_{Jambe-Pied}</i>	A C C C A
<i>Signature globale de chaque sujet sain</i>	A A C A C C C A

La segmentation des deux CRP d'un sujet sain est présentée sur la figure 4.35 en utilisant la méthode structurelle. La signature correspondant à chacune de ses CRP est présentée.

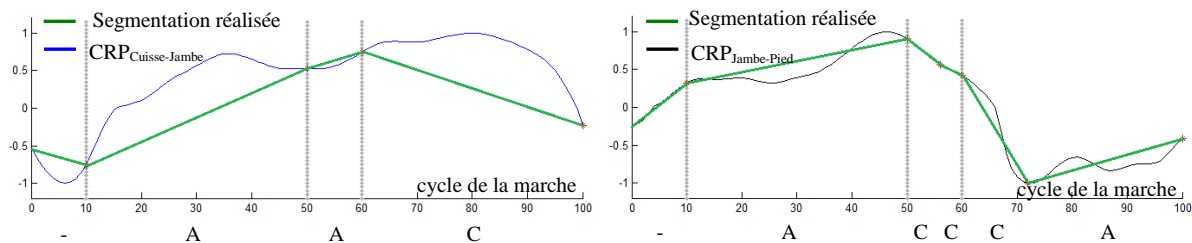


Figure 4.36. Exemple de segmentation et de signature obtenue pour chaque CRP d'un sujet sain. La première phase de chaque segmentation n'est pas retenue dans l'établissement de la signature de chaque sujet.

La signature globale obtenue pour chaque patient permet de caractériser sa coordination inter-segmentaire pour l'ensemble des segments de la jambe. L'utilisation d'une séquence de 8 primitives permet de caractériser la signature globale unique de l'ensemble des sujets sains. A l'inverse, il n'y a pas de signature unique permettant de caractériser les patients hémiparétiques puisqu'ils ont des degrés de troubles de la marche qui leurs sont propres.

En utilisant la règle de classification de la méthode structurelle, les patients sont classifiés en fonction de la similarité établie entre leur signature globale et celle caractérisant les sujets sains. Chaque primitive p_i , $i=1\dots 8$, composant la signature d'un patient est comparée à la primitive p_i lui correspondant chez les sujets sains. L'ordre des primitives est ainsi conservé. Nous avons considéré les primitives A, B et C utilisées comme complètement différentes, i.e. elles n'ont aucune similarité entre elles. Ainsi, dès qu'une primitive ne correspond pas à celle d'un sujet sain, cela montre une différence importante de la coordination inter-segmentaire de la marche du patient. La mesure de similarité définie à l'équation (3.1) est alors utilisée pour classier chaque patient. Afin d'obtenir un résultat de classification structurelle le plus fiable

possible, nous avons considéré que seuls les patients avec une similarité égale à 1 envers la classe des sujets sains sont classés dans cette classe.

90,3 % des patients sont bien classifiés en utilisant seulement la méthode structurelle. Ce pourcentage de classification s'explique par le fait que certains patients hémiparétiques ont des CRP avec une tendance globalement similaire à celle des sujets sains. Ces patients présentent des troubles de la marche moins importants.

Résultats du classifieur statistique :

Pour l'application clinique sur les CRP de patients hémiparétiques, plus de 20 paramètres statistiques ont été calculés : la moyenne d'amplitude, le nombre de pics, le moment où l'amplitude est minimale, la longueur d'un épisode, la valeur du Kurtosis, la valeur RMS, le temps de croissance d'un épisode, etc. Ces paramètres statistiques ont été calculés sur des fenêtres temporelles provenant des épisodes trouvés par la méthode de segmentation. Il est important de préciser que les paramètres statistiques, qui ont été calculés sur chaque épisode d'une forme, apportaient chacun une information interprétable de la coordination de la marche. Il pouvait s'agir d'un déphasage, d'un nombre élevé de tremblements, etc. Seulement, tous ces paramètres ne permettaient pas de discriminer les classes. Pour sélectionner les paramètres les plus informatifs, le critère discriminant de Fisher (Fisher, 1936) a été calculé par rapport aux paramètres de chaque phase. Ce critère calcule le pourcentage d'informations donné à chaque paramètre dans la discrimination des classes. Les paramètres sélectionnés ont permis d'établir l'espace de représentation de la méthode et ainsi de représenter chaque forme dynamique par un seul point.

C'est la valeur moyenne d'amplitude entre chaque couple de points de rupture, i.e. pour chaque épisode, qui est le paramètre le plus discriminant de chaque épisode. L'espace de représentation est donc composé de 8 paramètres statistiques, correspondant à la valeur moyenne d'amplitude des 8 épisodes trouvés par la méthode de segmentation. Chaque patient est alors représenté par un point dans cet espace lors de sa classification.

Nous avons choisi d'utiliser la méthode K-Plus Proches Voisins (KPPV). Cette méthode est simple et efficace. Parmi les inconvénients de la méthode on peut citer la difficulté que l'on peut avoir à définir son paramètre k ou le fait que KPPV devient inefficace si la taille de l'ensemble des données devient trop importante. Cependant dans le cadre de notre application médicale, le faible nombre de formes et de dimensions de l'espace de représentation rend cette méthode parfaitement adaptée. De nombreux articles traitent la méthode KPPV, toutefois

nous l'expliquons rapidement dans les lignes qui suivent pour permettre au lecteur de comprendre la globalité de l'approche mixte proposée en utilisant cette méthode statistique.

KPPV estime la densité de probabilité conditionnelle de chaque forme par rapport à ses voisins de la même classe. La méthode est supervisée, elle nécessite un ensemble d'apprentissage de données connues, réparties en c classes. Pour classer une forme x , KPPV estime la densité de probabilité conditionnelle p de cette forme par rapport à chaque classe C_i , $i=1, \dots, c$:

$$p\langle x|C_i \rangle = \frac{k_i}{N_i} \quad (4.4)$$

où N_i est le nombre de formes comprises dans la classe C_i et k_i représente le nombre de plus proches voisins de x qui appartiennent à cette classe C_i . Les k plus proches voisins de la forme x font partie d'un volume $V(x)$ centré sur x . Ces voisins sont ceux qui ont la plus petite distance par rapport à x . Selon la mesure de distance choisie, la forme du volume $V(x)$ est modifiée. Avec la distance Euclidienne, le volume est une hypersphère, avec la distance de Manhattan le volume $V(x)$ est un hypercube et avec la distance de Mahalanobis $V(x)$ est une hyper-ellipse. La forme x est affectée à la classe la plus représentée parmi ses k voisins. Dans cette version de KPPV, la valeur d'appartenance d'une forme à une classe est égale à 1 ou à 0, mais plusieurs autres versions des KPPV existent et permettent de calculer différemment la valeur d'appartenance d'une forme par rapport à chaque classe. La méthode KPPV Floue (KPPVF) est une version très populaire de KPPV permettant d'obtenir une valeur d'appartenance floue.

Les performances des méthodes KPPV dépendent donc du paramètre k . Son choix est déterminé par expérimentation, mais il reste un paramètre difficile à fixer. Un critère souvent utilisé pour sa définition est $k = \sqrt{N}$ défini par Dubuisson (Dubuisson, 1990).

En utilisant la méthode de validation croisée "Leave One Out" sur chacune formes obtenues pour la base de données des CRP, le résultat de classification de la méthode statistique est de 98,9 %. En plus de valider les résultats de classification, la méthode "Leave One Out" permet de profiter pleinement de chacune des formes de l'ensemble d'apprentissage. En effet durant la phase de classification d'une forme, c'est l'ensemble des autres formes connues qui sont utilisées. L'utilisation de l'ensemble des données connues pour la classification de chaque forme permet ainsi d'obtenir un résultat de classification plus fiable.

Résultats de classification obtenus par le multi-classifieur :

En se basant sur les résultats obtenus par le classifieur structurel et le classifieur statistique, la règle de décision globale est appliquée. Nous avons fixé le même poids de fiabilité aux deux classifieurs utilisés par le multi-classifieur, c'est-à-dire que nous faisons autant confiance au classifieur structurel qu'au classifieur statistique. Ainsi, par rapport à la règle de décision définie à l'équation 3.2, il nous fallait définir ou faire définir la décision à prendre pour les cas litigieux. Les sujets sains ont des caractéristiques similaires entre eux. S'il y a un doute au niveau de la classification d'un patient cela signifie donc qu'il ne fait pas pleinement partie de la classe des sujets sains. Par exemple, un patient peut avoir des caractéristiques statistiques proches de celles d'un sujet sain mais avoir des caractéristiques structurelles différentes de celles d'un sujet sain. Cette différence peut être plus faible pour certains patients que pour d'autres, mais nous avons considéré qu'en cas de doute, un patient fait vraisemblablement partie de la classe de coordination inter-segmentaire de la marche des patients hémiparétiques.

En utilisant cette règle, 100 % des patients sont alors bien classifiés. La classification de chaque forme de la base de données clinique est alors réalisée en environ 10^{-1} s. Les résultats de classification des CRP sont regroupés dans le tableau 4.9.

Tableau 4.9. Résultats de classification obtenus par le multi-classifieur mixte.

	<i>Résultats de classification</i>
<i>Classifieur structurel</i>	90,3 %
<i>Classifieur statistique</i>	98,9 %
<i>Multi-classifieur mixte</i>	100 %

L'utilisation conjointe des informations structurelle et statistique obtenues sur les CRP permet de discriminer l'ensemble des patients hémiparétiques des sujets sains.

4.2.1.e. Interprétation des résultats

L'interprétation des résultats est nécessaire aux cliniciens. Elle permet de caractériser les patients et leurs maladies tout en permettant de sélectionner le traitement le plus adapté au trouble identifié. En plus des résultats de classification obtenus, il est nécessaire d'obtenir d'autres informations pour pouvoir correctement interpréter une signature. Dans le cadre de l'application clinique de cette thèse, on ne peut pas simplement se baser sur les primitives définies pour désigner la coordination d'un patient. Une primitive seule indique pour une CRP

une progression ou une régression, d'un segment de jambe par rapport à l'autre, mais on ne peut pas savoir quel est le segment qui mène, pourtant c'est une information très importante. Les primitives de la signature d'une CRP permettent de caractériser la forme dynamique de la CRP, mais l'utilisation conjointe de paramètres statistiques permet de rendre complètement interprétable une signature.

Comme nous avons pu caractériser les sujets sains avec une signature structurée, nous avons cherché à caractériser ces mêmes sujets de façon statistique pour obtenir une signature mixte pour chaque patient. Cette signature mixte doit permettre une interprétation complète et simple de chaque patient. La signature mixte de chaque patient est établie simplement par rapport à chaque épisode, à l'aide de sa primitive et de sa valeur d'amplitude moyenne. Cette signature mixte est représentée pour un sujet sain dans le tableau 4.10.

Tableau 4.10. Signature mixte d'un sujet sain, basée sur sa signature structurée et sur la valeur statistique de chacun de ses épisodes de signature.

<i>Signature mixte caractéristique des sujets sains</i>	0,6 A	0,6 A	0,7 C	0,5 A	0,6 C	0,3 C	-0,6 C	-0,6 A
---	-------	-------	-------	-------	-------	-------	--------	--------

La signature globale obtenue pour chaque patient peut alors être complétée en utilisant les valeurs statistiques obtenues pour chacun de leur épisode. Nous avons considéré que la signature mixte représentative de la classe des sujets sains est composée de leur signature structurée unique et de la valeur statistique moyenne de chacun des épisodes obtenus pour tous les sujets sains de l'ensemble d'apprentissage.

Le signe obtenu pour la valeur statistique de chaque épisode amène une information très importante. En fonction du signe et de la primitive de chaque épisode de la signature mixte d'un patient, on peut déduire une interprétation de sa CRP. Il est d'ailleurs très important de signaler que l'ensemble des sujets sains ont le même signe pour chaque épisode. On peut donc dire qu'une signature unique mixte permet de caractériser l'ensemble des sujets sains (Tableau 4.11).

Tableau 4.11. Signature mixte de la classe des sujets sains, basée sur la primitive et le signe de chaque épisode.

<i>Signature mixte caractéristique des sujets sains</i>	+ A	+ A	+ C	+ A	+ C	+ C	- C	- A
---	-----	-----	-----	-----	-----	-----	-----	-----

En se basant sur cette signature mixte du tableau 4.11, chaque épisode auquel un sujet sain correspond peut alors être interprété dans l'ordre suivant par :

- +A : la 2^{ème} phase de la CRP_{Cuisse-Jambe} a une augmentation globale de sa tendance qui est supérieure à zéro. Cela signifie que dans la 2^{ème} phase de la CRP_{Cuisse-Jambe}, la cuisse mène de plus en plus la jambe,
- +A : dans la 3^{ème} phase de la CRP_{Cuisse-Jambe}, la cuisse mène de plus en plus la jambe,
- +C : dans la 4^{ème} phase de la CRP_{Cuisse-Jambe}, la cuisse mène progressivement moins la jambe,
- +A : dans la 2^{ème} phase de la CRP_{Jambe-Pied}, la jambe mène progressivement plus le pied,
- +C : dans la première partie de la 3^{ème} phase de la CRP_{Jambe-Pied}, la jambe mène progressivement moins le pied,
- +C : dans la seconde partie de la 3^{ème} phase de la CRP_{Jambe-Pied}, la jambe mène progressivement moins le pied,
- -C : dans la première partie de la 4^{ème} phase de la CRP_{Jambe-Pied}, le pied mène progressivement moins la jambe,
- -A : dans la seconde partie de la 4^{ème} phase de la CRP_{Jambe-Pied}, le pied mène progressivement plus la jambe.

100 % des sujets sains et des patients hémiparétiques ont été bien classifiés. Leurs phases caractéristiques sont analysées et leur interprétation peut être réalisée à l'aide de leur signature mixte. Nous pouvons alors nous intéresser à la mesure des valeurs d'appartenance de chaque patient à la classe des sujets sains, ainsi qu'à la quantification de l'évolution de la coordination des patients hémiparétiques avant et après un traitement thérapeutique, à l'aide de la mesure de similarité mixte proposée.

4.2.1.f. Utilisation de la mesure de similarité mixte

Dans la base de données que nous utilisons, certains patients ont passé une analyse de la marche pré et post traitement à la toxine botulique. Ce suivi de la marche des patients est intéressant pour quantifier leur évolution et distinguer les traitements faisant le plus effet sur leur hémiparésie. Ainsi, les cliniciens pourront valider un traitement pour certains patients ou le changer pour d'autres si cela s'avère nécessaire.

Une mesure de similarité mixte (3.4) est utilisée pour quantifier l'évolution des patients post traitement. La mesure de la similarité mixte est réalisée entre la signature mixte de

chaque patient et celle de la classe des sujets sains. En effet, on considère qu'un patient a une évolution positive de sa coordination s'il se rapproche de la classe des sujets sains. Cette mesure utilise la mesure de similarité structurelle (3.2) qui est basée sur le nombre de primitives communes entre signatures. Effectivement, puisque toutes les primitives utilisées (*A*, *B* et *C*) sont complètement différentes entre elles et que la longueur de chaque signature obtenue est identique aux autres signatures, la mesure de similarité est réalisée sur le nombre de primitives qui sont communes à celles des sujets sains en prenant en compte leur ordre, c'est-à-dire leur position dans une signature. La mesure de similarité mixte repose aussi sur la mesure de similarité statistique (3.3) qui est basée sur la valeur d'amplitude moyenne de la CRP par rapport à chaque épisode. Chaque valeur de similarité floue peut aller de 0 à 1. Une valeur de similarité de 1 signifie que le patient a exactement toutes les caractéristiques mixtes d'un sujet sain. Un patient avec une similarité de 0,4 ressemble peu aux sujets sains; il appartient peu à la classe des sujets sains.

Un exemple de mesure de similarité mixte calculée pour un sujet sain, pour un patient hémiparétique et pour un patient pré et post traitement est donné dans le tableau 4.12.

Tableau 4.12. Signatures mixtes et mesures de similarité de plusieurs patients.

	<i>Signature mixte</i>	<i>Mesure de similarité par rapport aux sujets sains</i>
<i>Sujet sain</i>	0,4A 0,7A 0,8C 0,6A 0,7C 0,4C -0,5C -0,6A (+A +A +C +A +C +C -C -A)	0,8
<i>Patient hémiparétique</i>	0,7A 0,7C 0,4C 0,2A 0,9A 0,8C -0,5C -0,8A (+A +C +C +A +A +C -C -A)	0,4
<i>Patient hémiparétique pré traitement</i>	0,7A 0,4C 0C 0,3A 0,9A 0,6C -0,3C -0,9A (+A +C +C +A +C +C -C -A)	0,4
<i>Patient hémiparétique post traitement</i>	0,7A 0,6A 0,3C 0,4A 0,9A 0,8C -0,2C -0,7A (+A +A +C +A +A +C -C -A)	0,5

Les résultats de classification obtenus par le multi-classifieur pour les sujets sains montrent qu'ils ont une valeur moyenne de similarité mixte égale à 0,7. Pour les patients hémiparétiques cette valeur décroît en moyenne jusque 0,4. Les patients hémiparétiques appartiennent donc beaucoup moins que les sujets sains à la classe des sujets sains d'après cette mesure. Pour les patients hémiparétiques pré et post traitement, cette mesure moyenne de similarité mixte passe de 0,4 à 0,5. Il y a donc eu amélioration globale de la coordination des patients post traitement.

4.3. Conclusion

L'ensemble des méthodes développées dans les chapitres 2 et 3 de ce manuscrit ont été appliquées sur divers exemples de simulations et réels. Les méthodes Fuzzy Pattern Matching Dynamique (FPMD), K-Plus Proches Voisins Flous Dynamiques Supervisée (KPPVFD-S) et Semi-Supervisée (KPPVFD-SS), proposées et détaillées dans le chapitre 2, ont été comparées à des méthodes d'apprentissage incrémental (FPMI, SVMI, KPPVI) afin de montrer l'apport des méthodes de classification dynamique pour les systèmes évolutifs.

Les méthodes supervisées de classification dynamique proposées (FPMD et KPPVFD-S) sont adaptées aux systèmes dont tous les modes de fonctionnement sont connus *a priori*. A l'inverse, l'utilisation de la méthode semi-supervisée (KPPVFD-SS) est plus adaptée aux cas des systèmes dont les modes de fonctionnement anormaux ne peuvent pas être connus à l'avance. Ces trois méthodes permettent de traiter l'évolution progressive de chaque classe et l'oubli des classes non représentatives du fonctionnement d'un système.

Le multi-classifieur mixte proposé au chapitre 3 est adapté à la classification de formes dynamiques. Il a permis de classer des formes dynamiques en tenant compte à la fois de leurs propriétés statistiques et structurelles. Cette combinaison de propriétés statistiques et structurelles a permis d'établir une signature mixte interprétable par les cliniciens pour chaque patient. De plus, une mesure de similarité mixte proposée permet de quantifier l'évolution que peut réaliser un patient avec un traitement.

Toutefois, ces méthodes nécessitent le réglage de plusieurs paramètres comme les seuils $th1$ et $th2$ pour la méthode FPMD et th_{Fusion} , $NbMin$, n_1 et n_2 pour les méthodes KPPVFD-S et KPPVFD-SS. C'est pourquoi, le développement d'un mécanisme permettant la détermination dynamique et adaptative de ces paramètres, en fonction de l'application et de l'évolution de son environnement, constitue une des perspectives de cette thèse.

En ce qui concerne les deux applications majeures de ce chapitre, le suivi de qualité des soudures et la caractérisation de la coordination inter-segmentaire de la marche des patients hémiparétiques, certaines conclusions peuvent être tirées. Pour le système de soudure, la classification de la qualité (bonne/mauvaise) de chaque soudure a bien été réalisée. Le retard de détection des problèmes de qualité est minime et la détection et le suivi des classes peuvent être réalisés efficacement en ligne. De plus, le retard de détection est compensé par le fait de ne pas avoir de fausses alarmes tout en ayant seulement une faible marge d'écart de qualité

dans les soudures considérées comme bonnes. Pour l'application clinique concernant les patients hémiparétiques, le multi-classifieur mixte proposé a permis de classifier et d'interpréter les formes de coordination inter-segmentaire de la marche des patients. Cette méthode n'est pas utilisable en ligne. Toutefois, cela n'était pas nécessaire puisque l'enregistrement des informations n'est pas réalisé en continu et la méthode ne met que peu de temps pour réaliser la classification des Phases Relatives Continues (CRP) des patients. L'utilisation d'une méthode de reconnaissance des formes dynamiques pour cette application a permis une classification automatique des sujets dont l'analyse quantifiée de la marche est réalisée et peut aider les cliniciens dans l'analyse de leur coordination. La combinaison des informations structurelle et statistique a amélioré les résultats de classification et a permis d'obtenir des résultats interprétables par les cliniciens. De plus, deux phases du cycle de la marche ont montrées qu'elles n'étaient pas discriminatives de la coordination inter-segmentaire de la marche des sujets sains et des patients hémiparétiques. La difficulté de caractériser les patients hémiparétiques à l'aide de l'approche proposée s'explique par le fait que l'hémiparésie est différente chez chaque patient. Toutefois, dans l'avenir il pourrait être intéressant de trouver des sous-classes de coordination pour regrouper les hémiparétiques dont les pathologies sont les plus proches.

Conclusion générale et Perspectives

Dans ce manuscrit, nous nous sommes intéressés au diagnostic et au suivi des systèmes évolutifs à l'aide des méthodes de reconnaissance dynamique des formes et des méthodes de reconnaissance des formes dynamiques. En effet suite aux changements temporels qui surviennent lors de l'utilisation des systèmes évolutifs, les formes et les classes évoluent. Dans ce cas, pour pouvoir correctement estimer les modes de fonctionnement normaux ou défaillants d'un système, il est nécessaire d'adapter les classes en utilisant les données représentatives des caractéristiques actuelles du système. Afin de montrer cette nécessité de classification dynamique, plusieurs méthodes d'apprentissage incrémental ont été utilisées sur des applications dynamiques réelles présentées dans le chapitre 4. Ces méthodes permettent d'intégrer en ligne l'information apportée par une nouvelle forme pour améliorer l'estimation des modèles de classes. Cependant, il a été démontré qu'elles ne permettent pas de tenir compte de la représentativité des nouvelles formes au cours du temps et par conséquent qu'elles n'adaptent pas correctement les classes évoluées. Sans réaliser ce suivi progressif et en ligne des classes dynamiques, les méthodes incrémentales ne détectent pas les évolutions des modes de fonctionnement, ce qui peut mener à une dégradation croissante de leurs performances.

C'est pourquoi dans ce manuscrit nous avons commencé par présenter dans le chapitre 1, les différentes méthodes de reconnaissance dynamique des formes et de reconnaissance des formes dynamiques qui constituent une solution à ce problème. Elles permettent de suivre l'évolution des classes au cours du temps, et de tenir compte de l'historique et de la structure des formes.

Dans le chapitre 2, nous avons montré les limites des méthodes d'apprentissage incrémental puis nous avons présenté trois méthodes de classification dynamique que nous avons proposées. Ces méthodes permettent de prendre en compte les évolutions des classes en considérant les formes les plus représentatives du fonctionnement d'un système. Plusieurs exemples illustratifs ont été présentés.

Le chapitre 3 concernait les méthodes de classification des formes dynamiques. Le principe de plusieurs de ces méthodes a été détaillé. Une approche structurelle de classification des formes dynamiques a été proposée et son fonctionnement a été illustré. Puis, un multi-classifieur mixte reposant sur les caractéristiques statistiques et structurelles des

formes a été proposé. A partir des méthodes présentées et proposées, il a été vu que les objectifs de ces méthodes sont souvent propres à une application.

Dans le chapitre 4, les méthodes de reconnaissance dynamique des formes et les méthodes de reconnaissance des formes dynamiques proposées ont été appliquées à plusieurs systèmes simulés et réels. Ces méthodes ont vu leurs fonctionnements détaillés et leurs résultats présentés. Les résultats de classification obtenus montrent que les approches proposées ont réalisé efficacement le suivi et le diagnostic des systèmes étudiés.

La contribution de ce manuscrit au niveau des méthodes de reconnaissance des formes dans un environnement dynamique a été réalisée par :

- la méthode Fuzzy Pattern Matching Dynamique (FPMD). FPMD permet de suivre les évolutions d'un système en adaptant les caractéristiques d'une classe. Cette méthode basée sur la méthode Fuzzy Pattern Matching Incrémentale (FPMI) est supervisée. Elle comble les lacunes de FPMI concernant les classes dynamiques. Pour ce faire, FPMD utilise une mesure d'utilité des formes et un résidu pour détecter, suivre et confirmer les évolutions du système. Un mécanisme de suppression des classes devenues obsolètes fait partie de la méthode tout comme deux mécanismes de scission et de fusion afin d'adapter au mieux chaque classe à l'évolution lui correspondant (déplacement, fusion, scission, rotation, etc.). Les caractéristiques du classifieur sont alors mises à jour et la classification des nouvelles formes peut se faire en ligne. Toutefois, le fonctionnement de FPMD repose sur plusieurs paramètres comme les deux seuils $th1$ et $th2$ dont le choix des valeurs a une influence sur les performances de la méthode FPMD,
- deux méthodes basées sur les K-Plus Proches Voisins Flous (KPPVF) : KPPVF Dynamique Supervisée (KPPVFD-S) et KPPVF Dynamique Semi-Supervisée (KPPVFD-SS). Ces méthodes intègrent des mécanismes de détection et de confirmation des évolutions des classes basées sur deux mesures de variation d'écart type et de variation de la position du centre de gravité d'une classe. Les deux mesures définies tiennent compte du bruit des données dans la validation d'une évolution. Une sélection des formes les plus représentatives du fonctionnement actuel d'un système a alors lieu et la ou les classes sont adaptées en ligne. Un mécanisme de suppression des classes devenues obsolètes est utilisé de façon continue. Des mécanismes permettant de gérer les cas de scission et de fusion des classes ont également été intégrés à ces

méthodes. La version semi-supervisée a été proposée dans le cadre des systèmes pour lesquels l'ensemble des modes de fonctionnement ne peut pas être connu initialement. Cette méthode conserve l'ensemble des classes connues et intègre un mécanisme lui permettant de détecter l'apparition de nouvelles classes lorsque des évolutions de système ont lieu. Ainsi ces deux méthodes n'ont pas totalement la même vocation. La méthode supervisée met complètement à jour le classifieur en oubliant les informations obsolètes après adaptation d'une classe. La méthode semi-supervisée met à jour les classes après adaptation, et elle détecte l'apparition de nouvelles classes quand il y a évolution. Par contre lors d'une adaptation de classe, KPPVFD-SS ne supprime pas les anciennes formes de la classe. Elle les conserve et ne les supprimera qu'après validation de la non représentativité de la classe. Ainsi KPPVFD-SS conserve certaines données qui peuvent être importantes, telles que celles concernant les modes de fonctionnement transitoire d'un système. Les paramètres de ces méthodes doivent être définis de façon propre à chaque système, avec l'aide d'un expert lorsque cela est possible. Cependant, il est possible d'utiliser initialement les valeurs par défaut proposées pour chaque méthode. Parmi les paramètres dont les valeurs influent sur les résultats de classification et de suivi des évolutions des classes, on peut citer $th1$, $NbMin$, n_1 et n_2 pour les méthodes KPPVFD-S et KPPVFD-SS.

- deux méthodes de classification des formes dynamiques. La première méthode est basée sur les caractéristiques structurelles, i.e. les primitives, qui représentent l'évolution d'une forme dynamique au cours du temps. La méthode est composée d'une technique de segmentation ne nécessitant pas d'erreur d'approximation, reposant sur la définition d'une signature de taille fixe pour chaque forme dynamique. En complément, une méthode d'adaptation du nombre de primitives est proposée afin d'obtenir une signature caractéristique. Une fois la signature de chaque forme dynamique obtenue, sa classification est réalisée à l'aide d'une mesure de similarité. Cette méthode se veut adaptée aux systèmes dont les données dynamiques peuvent avoir une structure propre à un ou plusieurs modes de fonctionnement. Toutefois, l'approche proposée fonctionne hors ligne et doit être adaptée à chaque système. La deuxième méthode proposée pour les formes dynamiques utilise les caractéristiques structurelles (primitives, signatures, etc.) de chaque forme ainsi que les paramètres statistiques calculés dans des fenêtres de données de chaque forme dynamique. Ces fenêtres de données correspondent aux épisodes trouvés par la méthode de

classification structurelle. La complémentarité de ces informations permet en général d'augmenter le taux de classification. De plus, la méthode proposée permet d'obtenir une signature aux caractéristiques mixtes particulièrement représentative d'une classe. Cette signature a pour but d'être facilement interprétable en tenant compte de la liaison entre les données et de leurs caractéristiques. Une mesure de similarité mixte, basée à la fois sur la similarité entre caractéristiques structurelles et statistiques, a également été calculée. Elle peut être utilisée pour mesurer l'appartenance d'une forme à une classe et pour quantifier l'évolution que peut réaliser une forme suite à un changement de caractéristiques d'un système. La limite de ces méthodes vient de la dépendance du choix de leurs paramètres (types de primitives, N_{max} , etc.) à l'application utilisée.

Chacune des méthodes développées a montré ses capacités à travailler avec différents types d'applications simulées et réelles. Le suivi de l'évolution d'un défaut a été réalisé sur un benchmark de deux réservoirs. Le changement de mode de fonctionnement a été détecté sur un système de commutation dont les modes de fonctionnement étaient particulièrement proches. Une étude complète (acquisition, analyse, classification) a été réalisée sur un système industriel de soudures métalliques. Cette étude a permis la caractérisation des classes de fonctionnement normal et défaillant ainsi que leur suivi au cours du temps. Ainsi, suite à l'usure des outils de la machine de soudure ou suite à d'autres problèmes temporels, il est possible de détecter la qualité de soudure d'une pièce métallique. Une autre application concerne la détection de fuites sur un générateur de vapeur. La méthode utilisée a permis de caractériser les évolutions possibles sur ce système et de réaliser une classification pertinente des données. Enfin, les méthodes de reconnaissance des formes dynamiques ont été appliquées à la caractérisation de la coordination inter-segmentaire de la marche des patients hémiparétiques. Des résultats interprétables ont été obtenus afin d'aider les cliniciens dans leurs démarches de diagnostic et une mesure permettant de quantifier l'efficacité d'un traitement a été proposée.

L'ensemble des méthodes proposées ont pour but d'aider les experts humains dans la phase de classification des données. Outre le fait qu'elles permettent de suivre les modes de fonctionnement d'un système, elles se veulent complémentaires à ce qu'apporte un expert humain. Elles utilisent chacune des paramètres différents et ont des contraintes d'utilisation différentes (méthode supervisée, semi-supervisée, en ligne, hors ligne, etc.). Ainsi, il est

important de sélectionner la méthode correspondant le mieux au système sur lequel on souhaite l'appliquer.

Les résultats obtenus dans ce manuscrit démontrent l'apport des méthodes de reconnaissance des formes pour le diagnostic et le suivi des systèmes évolutifs. Toutefois, certaines améliorations pourront être faites en perspective de ces travaux. On peut citer entre autres :

- la simplification de la définition des paramètres des méthodes. Par exemple, pour FPMD, il serait intéressant de réaliser la détermination dynamique et adaptative de ses paramètres, en particulier $th1$ et $th2$, en fonction de chaque application et de l'évolution de son environnement,
- au niveau de KPPVFD-S et KPPVFD-SS, il serait intéressant d'intégrer un mécanisme d'adaptation dynamique de k ainsi qu'une mise à jour adaptative de certains paramètres propres à chaque classe. Par exemple, la valeur de $\sigma_i^{j_{min}}$ pourrait être mise à jour une fois les caractéristiques d'une classe validées dans le temps,
- pour la méthode KPPVFD-SS, il serait important au niveau du temps de classification, de limiter le nombre de formes conservées dans l'ensemble d'apprentissage. Pour ce faire, il faudrait supprimer davantage de formes obsolètes. Par exemple, si une classe conserve ses caractéristiques, i.e. elle ne réalise pas d'évolutions, il n'est pas utile de conserver chacune des formes classifiées, il suffit de garder un nombre de formes représentatives de cette classe,
- au niveau des méthodes de classification des formes dynamiques proposées, il serait intéressant de proposer une mesure permettant d'estimer le nombre de primitives le plus adapté à une forme dynamique en fonction de sa structure,
- qu'afin de regrouper les méthodes développées et de simplifier leur accès aux utilisateurs, il serait intéressant de développer une Interface Homme Machine (IHM). Elle permettrait aussi d'obtenir un résultat adapté aux modes de raisonnement des opérateurs. Cette IHM pourrait intégrer une aide à la définition des paramètres des méthodes en testant plusieurs paramètres possibles, et en proposant ou en sélectionnant par défaut la valeur la plus adaptée à un système. En effet, en plus du résultat de classification attendu pour une méthode de reconnaissance des formes, il est important de présenter des informations interprétables aux utilisateurs d'une méthode,

- que dans le cadre du développement des multi-classifieurs, il serait intéressant de sélectionner de façon dynamique les classifieurs les plus adaptés à une application. Ces travaux sont déjà commencés dans (Ayad, 2010), où les auteurs proposent de sélectionner la meilleure combinaison de classifieurs pour la déployer uniquement dans les zones de chevauchement ou d'ambiguïté qui ont lieu entre certaines classes,
- qu'il peut être intéressant de combiner des classifieurs qui réalisent le suivi et l'adaptation des classes dynamiques. On parle alors de multi-classifieur évolutif. On pourrait par exemple proposer une version de multi-classifieur évolutif combinant les avantages de FPMD, KPPVFD-S et KPPVFD-SS et réduisant le taux d'erreur de classification ou le retard de détection des évolutions,
- que puisque les méthodes de classification dynamique permettent de limiter le retard de détection des évolutions et de suivre le déplacement d'une classe, des techniques de prédiction pourraient leur être ajoutées afin de ne plus réaliser le suivi d'un système mais son pronostic.

Bibliographie

- (Aksela, 2003) Aksela M., Comparison of classifier selection methods for improving committee performance. *Proceedings of MCS2003*, pp. 84-93, 2003.
- (Al-Ani, 2002) Al-Ani A., Deriche M., A new technique for combining multiple classifiers using the dempster-shafer theory of evidence, *Journal of Artificial Intelligence Research*, v.17 n.1, pp. 333-361, 2002.
- (Amadou Boubacar, 2005) Amadou Boubacar H., Lecoeuche S., Maouche S., Self-Adaptive Kernel Machine: Online Clustering in RKHS. *IEEE IJCNN05*, Montreal, Canada, 2005.
- (Amadou Boubacar, 2005) Amadou Boubacar H., Lecoeuche S., Maouche S., AUDyC Neural Network using a new Gaussian Densities Merge Mechanism. *In the 7th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA05, ed.)*, pp. 155-158, Coimbra, Portugal, 2005.
- (Amadou Boubacar, 2006) Amadou Boubacar H., *Classification Dynamique de Données non-stationnaires : Apprentissage séquentiel des Classes évolutives*. Thèse de Doctorat de l'Université des Sciences & Technologies de Lille, France, 2006.
- (Angelov, 2004a) Angelov P.P., Filev D.P., An Approach to Online Identification of Takagi-Sugeno Fuzzy Models. *IEEE Transactions on Systems, Man and Cybernetics* 34 (1), pp. 484-498, 2004.
- (Angelov, 2004b) Angelov P.P., Filev D.P., Flexible models with evolving structure. *IEEE Symposium on Intelligent Systems IS'02*, Varna, Bulgaria, pp. 28-33, 2002.
- (Angelov, 2010) Angelov P.P., Filev D.P., Kasabov N.K., Evolving intelligent systems: methodology and applications. *IEEE Press Series in Computational Intelligence. John Wiley and Sons and IEEE Press*, New York. ISBN 978-0-470-28719-4, 2010.
- (Angstenberger, 2000) Angstenberger L., *Dynamic Fuzzy Pattern Recognition*. Dissertation, Fakultät für Wirtschaftswissenschaften der Rheinisch-Westfälischen Technischen Hochschule, Aachen, Germany, 2000.
- (Armand, 2005) Armand S., *Analyse quantifiée de la marche - extraction de connaissances à partir de données pour l'aide à l'interprétation clinique de la marche digitigrade*, Université de Valenciennes et du Hainaut Cambresis, France, 2005.
- (Auda, 2000) Auda G., Kamel M., CMNN: cooperative modular neural network. *IEEE Conference on Neural Networks*. pp. 226-231, 2000.
- (Ayad, 2010) Ayad O., Sayed Mouchaweh M., Billaudel P., Sélection dynamique des classifieurs pour l'amélioration du taux de classification dans les zones d'ambiguïté, *6^{ème} Conférence Internationale Francophone d'Automatique (CIFA 2010)*, Nancy, France, 2010.

- (Bahler, 2000) Bahler D., Navarro L., Methods for Combining Heterogeneous Sets of Classifiers, *Proc. 17th Nat'l Conf. Am. Assoc. for Artificial Intelligence*, 2000.
- (Bakshi, 1994) Bakshi B.R., Stephanopoulos G., Representation of process trends. IV: Induction of real-time patterns from operating data for diagnosis and supervisory control, *Computers and Chemical Engineering*, 18(4), pp. 267-302, 1994.
- (Ball, 1965) Ball G.H., Hall D.J., *ISODATA, A novel method of data analysis and pattern classification*, Stanford Research Institute, Menlo Park, CA, Tech. Rep. AD 699616, 1965.
- (Batchelor, 1973) Batchelor B.G., A comparison of the decision surfaces of the Nearest Neighbour and Potential Function Classifiers, *Information Sciences* 5, pp. 171-178, 1973.
- (Barton, 1997) Barton J.G., Lees A., An application of neural networks for distinguishing gait patterns on the basis of hip-knee joint angle diagrams, *Gait & Posture Vol. 5 (1)*, pp. 28-33, 1997.
- (Begg 2006) Begg R., Kamruzzaman J., Neural networks for detection and classification of walking pattern changes due to ageing. *Australas Phys Eng Sci Med* 29(2), pp. 188-95, 2006.
- (Ben-David, 2004) Ben-David S., Gehrke J., Kifer D., Detecting Change in Data Streams, *Proceedings of VLDB*, 2004.
- (Benzécri, 1973) Benzécri J.P., *L'Analyse des Données tome 1 : Taxinomie et tome 2 : L'Analyse des Correspondances*, Bordas, Paris, 1973.
- (Beringer, 2007) Beringer, J., Hüllermeier, E., Efficient Instance-Based Learning On Data Streams. *Intelligent Data Analysis* 11 (6), pp. 627-650, 2007.
- (Bezdek, 1981) Bezdek J.C., *Pattern Recognition with fuzzy objective function algorithms*. Plenum Press, 1981.
- (Biederman, 1987) Biederman, I., Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review* 94, pp. 115-147, 1987.
- (Bifet, 2009) Bifet A., Gavaldà R., Adaptive learning from evolving data streams. *Proceedings of the 8th International Symposium on Intelligent Data Analysis: Advances in Intelligent Data Analysis VIII*, Lyon, France, 2009.
- (Borer, 2003) Borer S., *New Support Vector Algorithms for Multi-categorical Data: Applied to Real-Time Object Recognition*. Ph. D. Thesis, EPFL, Lausanne Swiss, 2003.

- (Boser, 1992) Boser B. E., Guyon I.M., Vapnik V.N., A training algorithm for optimal margin classifiers. *5th Annual Workshop on Computational Learning Theory*, pp. 144-152, Pittsburgh, USA, 1992.
- (Boukharouba, 2008) Boukharouba K., Lecoeuche S., Online Clustering of Non-stationary Data Using Incremental and Decremental SVM, *ICANN 08*, pp. 336-345, 2008.
- (Box, 1970) Box G., Jenkins G., *Time series analysis: Forecasting and control*. San Francisco: Holden-Day, 1970.
- (Brown, 1975) Brown R.L., Durbin J., Evans J.M., Techniques for Testing the Constancy of Regression Relationships over Time. *Journal of the Royal Statistical Society (37) 2*, pp. 149-192, 1975.
- (Brown, 1989) Brown D.C., Chandrasekaran B., Design problem solving. Knowledge structure and control strategy. *Research notes on Artificial Intelligence*. San Francisco: Morgan Kaufmann, 1989.
- (Bunke, 1990) Bunke H., Sanfeliu A., *Syntactic and Structural Pattern Recognition: Theory and Applications*, World Scientific, Second edition, 1990.
- (Burgess-Limerick, 1993) Burgess-Limerick R., Abernethy B., Neal R.J., Relative phase quantifies interjoint coordination. *J Biomechanics 26 (1)*, pp. 91-94, 1993.
- (Byington, 2002) Byington C., Roemer M., Kacprzyński G., Galie T., Prognostic Enhancements to diagnostic Systems for Improved Condition-based maintenance. *IEEE Aerospace Conference*, Big Sky, Montana, 2002.
- (Byorick, 2003) Byorick J.T., *Assessing classification confidence using a weighted exponential based technique with the Learn++ incremental learning algorithm*. Thesis, Rowan University, New Jersey, USA, 2003.
- (Calvelo, 1999) Calvelo D., Chambrin M.C., Pomorski D., Ravoux P., Towards symbolisation using data; driven extraction of local trends for ICU monitoring, *Artificial Intelligence in Medicine*, vol. 1(2), pp: 203-223, 2000.
- (Cauwenberghs, 2001) Cauwenberghs G., Poggio T., Incremental and Decremental Support Vector Machine Learning. *Adv. Neural Information Processing Systems*, Cambridge MA: MIT Press, vol. 13, 2001.
- (Cayrol, 1982) Cayrol M., Farreny H., Prade H., Fuzzy Pattern Matching. *Kybernetes*, 11, pp. 103-116, 1982.

- (Charbonnier, 2002) Charbonnier S., Becq G., Biot L., Carry P., Perdrix J.P., On line segmentation algorithm for ICU continuously monitored clinical data, *15th World IFAC congress, Barcelona, Spain, 2002.*
- (Charbonnier, 2004) Charbonnier S., Garcia-Beltran C., Cadet C., Gentil S., Trends extraction and analysis for complex system monitoring and decision support, *Engineering Applications of Artificial Intelligence*, vol. 18/1, pp. 21-36, 2004.
- (Chen, 1997) Chen K., Wang L., Chi H., Methods of combining multiple classifiers with different features and their applications to text-independent speaker identification. *Int. J. Pattern Recognit. Artificial Intell.* 11 (3), pp. 417-445, 1997.
- (Cheung, 1990) Cheung J.T.Y., Stephanopoulos G., Representation of process trends- Part I: A formal representation framework, *Computers chem. Engng*, vol. 14 (4/5), pp. 495-510, 1990.
- (Chiementin, 2008) Chiementin X., Bolaers F., Cousinard O., Rasolofondraibe L., Early detection of rolling bearing defect by demodulation of vibration signal using adapted wavelet. *Journal of Vibration and Control*, Vol. 14, No. 11, pp 1675-1690, 2008.
- (Ciarapica, 2006) Ciarapica F.E., Giacchetta G., Managing the condition-based maintenance of a combined-cycle power plant: an approach using soft computing techniques. *Journal of Loss Prevention in the Process Industries*, 19, pp. 316-325, 2006.
- (Cohen, 1995) Cohen W., Fast effective rule induction. *Int. Conf. on Machine Learning (ICML)*, pp. 115–123, 1995.
- (Cohen, 2004) Cohen L., Avrahami G., Last M., Incremental Info-Fuzzy Algorithm for Real Time Data Mining of Non-Stationary Data Streams. *TDM Workshop*, Brighton UK, 2004.
- (Colomer, 2002) Colomer J., Melendez J., Gamero F.I., Pattern recognition based on episodes and DTW. Application to diagnosis of a level control system, *16th International Workshop on Qualitative Reasoning QR02*, pp. 37-43, Sitges, Espagne, 2002.
- (Cremer, 2001) Cremer F., Schutte K., Schavemaker J.G.M., Den Breejen E., A comparison of decision-level sensor-fusion methods for anti-personnel landmine detection, *Information Fusion* 2, pp. 187-208, 2001.
- (Cover, 1967) Cover T.M., Hart P.E., Nearest neighbour pattern classification. *IEEE Transactions on Information Theory*, vol. 13, pp: 21-27, 1967.
- (Cozman, 2003) Cozman F., Cohen I., Cirelo M., Semi-supervised learning of mixture models. *20th International Conference on Machine Learning (ICML)*, Washington DC, USA, 2003.

- (Das, 1997) Das G., Gunopulos D., Mannila H., Finding Similar Time Series. In: Komorowski, J., Zytkow, J. (Eds.) *Principles of Data Mining and Knowledge Discovery. Proceedings of the First European Symposium PKDD'97*, Trondheim, Norway, June 1997, Springer, pp. 88-100, 1997.
- (Dasarathy, 1991) Dasarathy B.V., Nearest Neighbor (NN) Norms. *NN Pattern Classification Techniques*, 1991.
- (Dash, 2001) Dash S., Rengaswamy R., Venkatasubramanian V., A novel interval-halving algorithm for process trend identification. *4th IFAC Workshop on On-line Fault Detection and Supervision in the Chemical Process Industries*, Jeju Island, Korea, pp. 173-178, 2001.
- (Dash, 2003) Dash S., Rengaswamy R., Venkatasubramanian V., Fuzzy logic based trend classification for fault diagnosis of chemical processes. *Computers and Chemical Engineering* (27), pp. 347-362, 2003.
- (Daw, 2003) Daw C.S., Finney C.E.A., Tracy E.R., A review of symbolic analysis of experimental data, *Scientific Instruments*, vol. 74, no. 2, pp. 915-930, 2003.
- (Deng, 2003) Deng D., Kasabov N., On-line pattern analysis by evolving self organizing maps. *Neurocomputing*, vol. 51, pp: 87-103, 2003.
- (Denoeux, 2001) Denoeux T., Zouhal L.M., Handling possibilistic labels in pattern classification using evidential reasoning. *Fuzzy Sets and Systems*, 1(22), pp.409-424, 2001.
- (Devillez, 1998) Devillez A., Billaudel P., Villerman-Lecolier G., Une méthode possibiliste de classification pour le tri des bouteilles plastiques. *6^{ème} rencontres de la société francophone de classification SFC'98*, pp. 73-76, Montpellier, France, 1998.
- (Dijkstra, 1959) Dijkstra E.W., A note on two problems in connexion with graphs *Numerische Mathematik* (1), pp. 269-271, 1959.
- (Dimitru, 1986) Dimitru V., Luban F., On some optimisation problems under uncertainty. *Fuzzy Sets and Systems* 18, pp. 257-272, 1986.
- (Dombi, 1990) Dombi J., Membership function as an evaluation, *Fuzzy Sets and Systems* 35 (1), pp. 1-21, 1990.
- (Domingos, 2000) Domingos P., Hulten G., Mining high-speed data streams. *KDD 00*, pp. 71-80, 2000.
- (Dormido-Canto, 2008) Dormido-Canto S., Fadas G., Donnido R., Vega J., Sanchez J., Duro N., Vargas H., Ratta G., Pereira A., Portas A., Structural pattern recognition methods

- based on string comparison for fusion databases, *IAEA TM on Control, Data Acquisition, and Remote Participation for Fusion Research N°6*, Inuyama, Japan, 2008.
- (Dragomir, 2008) Dragomir O., Gouriveau R., Zerhouni N., Adaptive neuro-fuzzy inference system for mid term prognostic error stabilization. *International Journal of Computers Communications & Control*, 3, pp. 271-276, 2008.
- (Dubois, 1988) Dubois D., Prade H., *Possibility Theory*, New York: Plenum Press, 1988.
- (Dubuisson, 1990) Dubuisson B., *Diagnostic et reconnaissance des formes*. Traité des Nouvelles Technologies, série Diagnostic et Maintenance, HERMES, 1990.
- (Dubuisson, 2001) Dubuisson B., *Automatique et statistiques pour le diagnostic*. Traité IC2, série Productique. (Hermès Sciences), 2001.
- (Duda, 1973) Duda R.O., Hart P.E., *Pattern Classification and Scene Analysis*. Wiley-Interscience, John Wiley and Sons, New York, 1973.
- (Duda, 2001) Duda R.O., Hart P.E., Stork D.G., *Pattern Classification 2nd edition*. Wiley-Interscience, 2001.
- (Dufrenois, 2009) Dufrenois F., Colliez J., Hamad D., Bounded Influence Support Vector Regression for Robust Single-Model Estimation. *IEEE Transactions on Neural Networks* 20 (11), pp. 1689-1706, 2009.
- (Elshenawy, 2008) Elshenawy L., Ding S., *An advanced PCA for process monitoring: a benchmark study*. Coventry, UK: IAR / ACD, 2008.
- (Eltoft, 1998) Eltoft T., A new Neural Network for Cluster-Detection-and-Labeling. *IEEE Trans. on Neural Networks*, vol. 9(5), pp: 1021-1035, 1998.
- (Fayyad, 1996) Fayyad U.M., Piatetsky-Shapiro G., Smyth P., From Data Mining to Knowledge Discovery in Databases. *AI Magazine* 17(3):, pp. 37-54, 1996.
- (Fisher, 1936) Fisher R.A., The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics* 7, pp. 179-188, 1936.
- (Fix, 1951) Fix E., Hodges J.L., *Discriminatory analysis: Non-parametric discrimination: Consistency properties*. USAF School of Aviation Medicine, Randolph Field, TX, pp. 261–279, 1951.
- (Freedman, 1989) Freedman L.S., Spiegelhalter D.J., Comparison of Bayesian with group sequential methods for monitoring clinical trials. *Controlled Clinical Trials*, 10, pp. 357–367, 1989.
- (Friedman, 1999) Friedman M., Kandel A., *Introduction to Pattern Recognition: Statistical, Structural, Neural, and Fuzzy Logic Approaches*. World Scientific, Singapore, 1999.

- (Frélicot, 2006) Frélicot C., Berthier M., Mascarilla L., A new cluster validity index for fuzzy clustering based on combination of dual triples. *IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE)*, Vancouver, Canada, 2006.
- (Frigui, 1996) Frigui H., Krishnapuram R., A robust algorithm for automatic extraction of an unknown number of clusters from noisy data. *Pattern Recognition Letters*, vol. 17, pp 1223-1232, 1996.
- (Frigui, 1997) Frigui H., Krishnapuram R., Clustering by competitive agglomeration. *Pattern Recognition*, vol. 307, pp. 1109-1119, 1997.
- (Fu, 1982) Fu, K.S., *Syntactic Pattern Recognition and Applications*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- (Fukuyama, 1989) Fukuyama Y., Sugeno M., A new method for choosing the number of clusters for the fuzzy c-means method. *Proc. 5th Fuzzy Systems Symposium*, pp. 247-250, 1989.
- (Gabrys, 2000) Gabrys B., Bargiela A., General Fuzzy Min-Max Neural Network for Clustering and Classification. *IEEE Trans. Neural Networks*, vol. 11(3), pp: 769-783, 2000.
- (Gama, 2006) Gama J., Castillo G., Learning with local drift detection. *In Advanced Data Mining and Applications (ADMA)*, pp. 42-55, 2006.
- (Garcia, 2008) Garcia V., *Suivi d'objets d'intérêt dans une séquence d'images : des points saillants aux mesures statistiques*. Thèse de Doctorat de l'Université de Nice Sophia-Antipolis, 2009.
- (Gath, 1989) Gath I., Geva A.B., Unsupervised optimal Fuzzy Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, pp. 773-781, 1989.
- (Geetha, 2010) Geetha S., Ishwarya N., Kamaraj N., Evolving decision tree rule based system for audio stego anomalies detection based on Hausdorff distance statistics. *Information Sciences*, 2010.
- (Giacinto, 1998) Giacinto G., *Design of Multiple Classifier Systems*, PhD Thesis, Italy, 1998.
- (Giacinto, 2000) Giacinto G., Roli F., Dynamic Classifier Selection, *Proc. of the First Int. Workshop on Multiple Classifier Systems*, Cagliari, Italy, 2000.
- (Giani, 1999) Giani A., *Parallel Cooperative Classifier Systems*, Ph.D. Thesis, Pisa, Italy, 1999.
- (Gibb, 1994) Gibb W.J., Auslander D.M., Griffin J.C., Adaptive classification of myocardial electrogram waveforms. *IEEE Transactions on Biomedical Engineering*, 41, pp. 804-808, 1994.

- (Gil-Garcia, 2010) Gil-Garcia R., Pons-Porrata A., Dynamic hierarchical algorithms for document clustering. *Pattern Recognition Letters* 31 (6), pp.469-477, 2010.
- (Grossberg, 1988) Grossberg S., Nonlinear neural networks: principles, mechanism, and architectures. *Neural Networks* 1, pp. 17-61,1988.
- (Guedalia, 1999) Guedalia I.D., London M., Werman M., An On-line Agglomerative Clustering Method for Non-Stationary Data. *Neural Computation*, 11 (2), pp 521-540, 1999.
- (Guermeur, 2002) Guermeur Y., Combining Discriminant Models with New Multi-Class SVMs. *Pattern Analysis and Applications*, vol. 5(2), pp: 168-179, 2002.
- (Guigue, 2005) Guigue V., *Méthodes à noyaux pour la représentation et la discrimination de signaux non-stationnaires*. Thèse de Doctorat de l'INSA de Rouen, France, 2005.
- (Gunes, 2003) Gunes V., Ménard M., Loonis P., Petit-Renaud S., Combination, cooperation and selection of classifiers: a state of the art. *Int. J. Pattern Recognition Artif. Intell.* 17 (8), pp. 1303-1324, 2003.
- (Hamill, 1999) Hamill J., Van Emmerik R., Heiderscheit B.C., Li L., A dynamical systems approach to lower extremity running injuries. *Clinical Biomechanics* (14), pp. 297-308, 1999.
- (Hartert, 2008) Hartert L., Sayed Mouchaweh M., Billaudel P., Riera B., Reconnaissance des formes dans un environnement non-stationnaire, *5^{ème} Conférence Internationale Francophone d'Automatique (CIFA08)*. Bucarest-Roumanie, 2008.
- (Hartert, 2009a) Hartert L., Sayed Mouchaweh M., Billaudel P., A dynamic classification method for the discrimination of evolving data. *European Society for Fuzzy Logic and Technology 09, IFSA-EUSFLAT 09*, Lisbon, Portugal, 2009.
- (Hartert, 2009b) Hartert L., Gachout A., Sayed Mouchaweh M., Fault detection based on acoustic signals using Pattern Recognition and data analysis. *EAM 09*, Reims, France, 2009.
- (Hartert, 2009c) Hartert L., Sayed Mouchaweh M., Billaudel P., Diagnostic Prédicatif par Reconnaissance Dynamique des Formes. *Workshop du GIS Surveillance, Sûreté et Sécurité des grands systèmes (GIS - 3SGS 09)*. Nancy, France, 2009.
- (Hartert, 2009d) Hartert L., Sayed Mouchaweh M., Billaudel P., Dynamic Fuzzy Pattern Matching for the monitoring of non stationary processes. *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Safeprocess 09*, Barcelona, Spain, 2009.

- (Hartert, 2010a) Hartert L., Sayed Mouchaweh M., Billaudel P., Dynamic K-Nearest Neighbors for the monitoring of evolving systems. *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010)*. Barcelona, Spain, 2010.
- (Hartert, 2010b) Hartert L., Sayed Mouchaweh M., Billaudel P., *Monitoring of non stationary systems using dynamic Pattern Recognition, Intelligent Industrial Systems: Modelling, Automation and Adaptive Behaviour*. IGI. 2010.
- (Hartert, 2010c) Hartert L., Sayed Mouchaweh M., Billaudel P., A semi-supervised dynamic version of Fuzzy K-Nearest Neighbours to monitor evolving systems, *Springer-Verlag, Evolving Systems*, 2010.
- (Hartert, 2010d) Hartert L., Sayed Mouchaweh M., Hutin E., Billaudel P., Classification dynamique et suivi de l'évolution de la coordination de la marche des patients hémiparétiques. *Société Francophone de Classification (SFC 2010)*. Saint-Denis de la Réunion, 2010.
- (Hartert, 2010e) Hartert L., Sayed Mouchaweh M., A hybrid multi-classifier to characterize and interpret hemiparetic patients gait coordination. *ICMLA 2010*. Washington DC, USA, 2010.
- (Ho, 2002) Ho T.K., Multiple Classifier Combination: Lessons and Next Steps. *Hybrid Methods in Pattern Recognition, A. Kandel and H. Bunke, eds, World Scientific*, pp. 171-198, 2002.
- (Höppner, 2000) Höppner F., Piecewise Linear Function Approximation by Alternating Optimization. *In Proc. of the 8th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU)*, Madrid, Spain, pp. 1751-1757, 2000.
- (Huguency, 2003) Huguency B., *Représentation symboliques de longues séries temporelles*, Thèse de doctorat de l'Université de Paris 6, France, 2003.
- (Hulten, 2001) Hulten G., Spencer L., Domingos P., Mining time-changing data streams. *KDD 01*, pp. 97-106, 2001.
- (Hunter, 1986) Hunter J.S., The Exponentially Weighted Moving Average. *Journal of Quality Technology* 18, pp. 203–210, 1986.
- (Ilarri, 2008) Ilarri S., Mena E., Illarramendi A., Using cooperative mobile agents to monitor distributed and dynamic environments. *Information Sciences* 178 (9), pp. 2105-2127, 2008.
- (Jain, 2000) Jain A., Duin R., Mao J., Statistical pattern recognition: a review. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22(1), pp: 4-37, 2000.

- (Janusz, 1991) Janusz M., Venkatasubramanian V., Automatic generation of qualitative description of process trends for fault detection and diagnosis, *Engng. Applic. Artif. Intell.*, vol. 4, pp. 329-340, 1991.
- (Joentgen, 1999) Joentgen A., Mikenina L., Weber R., Zimmermann H.J., Dynamic Fuzzy Data Analysis based on Similarity between Functions. *Fuzzy Sets and Systems*, 105 (1), pp. 81-90, 1999.
- (Jolliffe, 2002) Jolliffe I.T., *Principal Component Analysis*, 2nd edition. Springer-Verlag, 2002.
- (Kamruzzaman, 2006) Kamruzzaman J., Begg R.K., Support vector machines and other pattern recognition approaches to the diagnosis of cerebral palsy gait. *IEEE Trans Biomed Eng*, pp. 2479-2490, 2006.
- (Keller, 1985) Keller J.M., Gray M.R., Givens J.A., A fuzzy k-nn neighbor algorithm. *IEEE Trans. Syst. Man Cybern.*, SMC-15(4), pp. 580–585, 1985.
- (Kelso, 1986) Kelso J., Pattern formation in speech and limb movements involving many degrees of freedom. *Exp Brain Res (15)*, 1986.
- (Kempowsky, 2004) Kempowsky, T., Surveillances de procédés à base de méthodes de classification : conception d'un outil d'aide pour la détection et le diagnostic de défaillances, thèse du LAAS, Grenoble, France, 2004.
- (Keogh, 2001) Keogh E., Chu S., Hart D., Pazzani M., An Online Algorithm for segmenting Time series, *IEEE International Conference on Data Mining*, 2001.
- (Khalil, 1999) Khalil M., *Une approche de la détection et de la classification dans les signaux non stationnaires. Application à l'EMG utérin*. Thèse de Doctorat de l'Université de Technologie de Troyes, France, 1999.
- (Koski, 1995) Koski A., Juhola M., Meriste M., Syntactic Recognition of ECG Signals by Attributed Finite Automata. *Pattern Recognition*, 28(12), pp. 1927–1940, 1995.
- (Koski, 1994) Koski E.M.J., Sukuvaara T., Makivirta A., Kari A., A knowledge; based alarm system for monitoring cardiac operated patients—assessment of clinical performance, *International Journal of Clinical Monitoring and Computing* 11, pp. 79-83, 1994.
- (Kybic, 2006) Kybic J., Incremental updating of nearest neighbor-based high-dimensional entropy estimation. *ICASSP 06*, 2006.
- (Last, 2002) Last M., Online Classification of Nonstationary Data Streams. *Intelligent Data Analysis*, Vol. 6, No. 2, pp. 129-147, 2002.

- (Lau, 2009) Lau H.Y., Tong K.Y., Zhu H., Support vector machine for classification of walking conditions of persons after stroke with dropped foot. *Hum Mov Sci* 28(4), pp. 504-514, 2009.
- (Law, 2005) Law Y.N., Zaniolo C., An Adaptive Nearest Neighbor Classification Algorithm for data streams. *9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2005)*, Porto, Portugal, 2005.
- (Lecoeuche, 2003) Lecoeuche S., Lurette C., Auto-adaptive and Dynamical Clustering Neural Network. *ICANN 2003*, pp. 350-358, 2003.
- (Littlestone, 1994) Littlestone N., Warmuth M.K., The Weighted Majority algorithm, *Information and Computation* 108, pp.212–261, 1994.
- (Liu, 2000) Liu Q., Levinson S., Wu Y., Thomas S., Huang S., Interactive and Incremental Learning via a Mixture of Supervised and Unsupervised Learning Strategies. *Joint Conf. on Information Systems (JCIS' 00)*, NJ, USA, 2000.
- (Lughofer, 2009) Lughofer E., Angelov P.P., Detecting and Reacting on Drifts and Shifts in On-Line Data Streams with Evolving Fuzzy Systems. *IFSA-EUSFLAT*, Lisbon, Portugal, pp 931-937, 2009.
- (Lughofer, 2010) Lughofer E., On Dynamic Selection of the Most Informative Samples in Classification Problems. *ICMLA*, Washington DC, USA, 2010.
- (Lurette, 2003) Lurette C., *Développement d'une technique neuronale auto-adaptative pour la classification dynamique des données évolutives : Application à la supervision d'une presse hydraulique*. Thèse de doctorat Université des Sciences et Technologies de Lille, France, 2003.
- (MacQueen, 1967) MacQueen J.B., Some Methods for classification and Analysis of Multivariate Observations, *5th Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, University of California Press (1), pp. 281-297, 1967.
- (Mao, 2008) Mao Y., Saito M., Kanno T., Wei D., Muroi H., Walking pattern analysis and SVM classification based on simulated gaits. *Conf Proc IEEE Eng Med Biol Soc*, pp. 5069-5072, 2008.
- (Marsili-Libelli, 1998) Marsili-Libelli S., Adaptive Fuzzy Monitoring and Fault Detection. *International Journal of COMADEM*, 1(3), pp. 31-37, 1998.
- (Medasani, 1998) Medasani S., Jaeseok K., Krishnapuram R., An overview of membership function generation techniques for pattern recognition. *International Journal of Approximate Reasoning* 19, pp. 391-417, 1998.

- (Medina, 2009) Medina J.A., Gomez F., Moreno J.A., An Evolving Fuzzy Neural Network Based on the Mapping of Similarities, *IEEE Trans. Fuzzy Syst.* 17 (6), pp. 1379-1396, 2009.
- (Michie, 1994) Michie D.M., Spiegelhalter D.J., Machine learning, Neural and Statistical Classification. *C.C. Taylor (Eds), Chichester: Ellis Horwood, 1994.*
- (Min, 2005) Min R., *A non-linear dimensionality reduction method for improving nearest neighbour classification.* Thesis, DCS, University of Toronto, 2005.
- (Muller, 2008) Muller A., Suhner M.C., Iung, B., Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial System. *Reliability Engineering and System Safety*, 2008.
- (Mustapha, 2008) Mustapha O., Khalil M., Hoblos G., Détection des défauts industriels par un banc de filtres et l'algorithme DCS, GIS-3SGS, Troyes, 2008.
- (Nakhaeizadeh, 1997) Nakhaeizadeh G., Taylor C., Kunisch G., Dynamic Supervised Learning. Some Basic Issues and Application Aspects. *In R. Klar, O. Opitz (Eds.) Classification and Knowledge Organization.* Springer-Verlag, pp. 123-135, 1997.
- (Nezhad, 2010) Nezhad M.S.F., Niaki S.T.A., A new monitoring design for uni-variate statistical quality control charts. *Information Sciences* 180 (6), Special Issue on Modelling Uncertainty, pp.1051-1059, 2010.
- (Olney, 1996) Olney S.J., Richards C., Hemiparetic gait following stroke. Part I: Characteristics. *Gait Posture* (4), pp. 136-148, 1996.
- (Olszewsky, 2001) Olszewski R.T., *Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data.* PhD dissertation, Carnegie Mellon University, USA, 2001.
- (Parzen, 1962) Parzen E., On the estimation of a probability density function and mode. *Annals of Mathematical Statistics* 33, pp. 1065-1076, 1962.
- (Pinquier, 2004) Pinquier J., *Indexation sonore : recherche de composantes primaires pour une structuration audiovisuelle.* Thèse soutenue à Université Toulouse III - Paul Sabatier, France, 2004.
- (Quinlan, 1993) Quinlan J.R., *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.
- (Rafiq, 2004) Rafiq L., Chryssanthopoulos M., Onoufriou T., Performance updating of concrete bridges using proactive health monitoring methods. *Reliability Engineering and System Safety*, 86(3), pp. 247-256, 2004.

- (Rengaswamy, 1995) Rengaswamy R., A syntactic pattern-recognition approach for process monitoring and fault diagnosis. *Engineering Applications of Artificial Intelligence*, pp. 35-51, 1995.
- (Ribot, 2009) Ribot P., *Vers l'intégration diagnostic/pronostic pour la maintenance des systèmes complexes*. Thèse de doctorat de l'Université Paul Sabatier - Toulouse III, 2009.
- (Ripley, 1996) Ripley, B.D., *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, 1996.
- (Roberts, 1959) Roberts S.W., Control chart tests based on geometric moving averages. *Technometrics* 1, pp. 239-250, 1959.
- (Rodda, 2001) Rodda J., Graham H.K., Classification of gait patterns in spastic hemiplegia and spastic diplegia: a basis for a management algorithm. *European Journal of Neurology* 8 (5), pp. 98-108, 2001.
- (Roncaglia, 2004) Roncaglia A., Elmi I., Dori L., Rudan M., Adaptive K-NN for the Detection of Air Pollutants With a Sensor Array. *IEEE Sensor Journal*, Vol. 4 (2), 2004.
- (Rüping, 2001) Rüping S., Incremental Learning with Support Vector Machines. *IEEE International Conference on Data Mining (ICDM 01)*, San Jose, CA, 2001.
- (Ruta, 2000) Ruta D., Gabrys, B., An overview of classifier fusion methods. In: Crowe, M. (Ed.), *Computing and Information Systems*, vol. 7, no. 1. pp. 1-10, 2000.
- (Sahel, 2007) Sahel Z., Bouchachia A., Gabrys B., Rogers P., Adaptive Mechanisms for Classification Problems with Drifting Data Source. *International Conference on Knowledge Based and Intelligent Information & Engineering Systems (KES 2007)*, 2007.
- (Sakoe, 1978) Sakoe H., Chiba S., Dynamic programming algorithm optimization for spoken word recognition, *Trans. on ASSP* 26, pp. 43-49, 1978.
- (Salomon, 2001) Salomon J., *Support Vector Machines for Phoneme Classification*. Master of science, School of Artificial Intelligence, University of Edinburgh, 2001.
- (Sammon, 1969) Sammon Jr.J.W., A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers* (18), pp.401-409, 1969.
- (Sayed Mouchaweh, 2002) Sayed-Mouchaweh M, Devillez A, Lecolier VG, Billaudel P (2002) Incremental Learning in fuzzy pattern matching. *Fuzzy Sets and Systems* 132 (1), pp. 49-62.

- (Sayed Mouchaweh, 2010) Sayed Mouchaweh M., Semi-supervised classification method for dynamic applications. *Fuzzy Sets and Systems*, vol. 161 (4), pp. 544-563, 2010.
- (Schwabacher, 2007) Schwabacher M., Goebel K., A Survey of Artificial Intelligence for Prognostics. *Working Notes of 2007 AAAI Fall Symposium: AI for Prognostics*, 2007.
- (Shewart, 1939) Shewhart W.A., *Economic Control of Quality Manufactured Product*. Princeton, NJ, D. Van Nostrand Reinhold, 1931.
- (Simpson, 1993) Simpson P.K., Fuzzy min-max neural networks. Part 2: clustering. *IEEE Transactions on Fuzzy Systems*, vol. 1(1), pp: 32-45, 1993.
- (Song, 2007) Song Y., Huang J., Zhou D., Zha H., Lee Giles C., IKNN: Informative K-Nearest Neighbor Pattern Classification. *PKDD 07*, Warsaw, Poland, 2007.
- (Stephanopoulos, 1997) Stephanopoulos G., Locher G., Duff M., Kamimura R., Fermentation data base mining by pattern recognition. *Biotechnology and Bioengineering*, vol. 53-5, pp. 443-452, 1997.
- (Struzik, 1995) Struzik, Z.R., The Wavelet Transform in the Solution to the Inverse Fractal Problem. *Fractals*, Vol. 3 (2), pp. 329-350, 1995.
- (Suykens, 1999) Suykens J.A.K., Vandewalle J., Least squares support vector machine classifiers, *Neural Processing Letters* 9 (3), pp. 293-300, 1999.
- (Tanaka, 1995) Tanaka E., Theoretical aspects of syntactic pattern recognition, *Pattern Recognition* 28, pp. 1053-1061, 1995.
- (Therrien, 1989) Therrien C.W., *Decision Estimation and Classification: An Introduction to Pattern Recognition and Related Topics*. John Wiley & Sons, New York, 1989.
- (Tong, 2001) Tong S., *Active learning: Theory & Applications*. PhD Thesis, Department of Computer Science of Stanford University, USA, 2001.
- (Triki, 2005) Triki S., *Contribution à la conception d'outils de supervision « active » pour les procédés continus et manufacturiers*. Thèse de doctorat de l'Université de Valenciennes et du Hainaut Cambresis, 2005.
- (Trouilhet, 1994) Trouilhet J., Coustaux I., Guilhot J., Présentation d'une méthodologie pour la prise de décision. Application à la reconnaissance automatique de signature acoustique, *Journal de Physique IV 04*, 1994.
- (Tsymbal, 2004) Tsymbal, A., *The problem of concept drift: definitions and related work*, Technical Report TCD-CS-2004-15, Department of Computer Science, Trinity College Dublin, Ireland, 2004.
- (Uncu, 2007) Uncu Ö., Türksen I.B., A novel feature selection approach: combining feature wrappers and filters. *Information Sciences* 177, pp. 449-466, 2007.

- (Vachkov, 2009) Vachkov G., Online Classification of Machine Operation Modes Based on Information Compression and Fuzzy Similarity Analysis. *IFSA-EUSFLAT 09*, Lisbon, Portugal, 2009.
- (Vapnik, 1995) Vapnik, V.N., The Nature of Statistical Learning Theory. *Springer-Verlag New York*, 1995.
- (Vedam, 1997) Vedam H., Venkatasubramanian V., A wavelet Theory-based Adaptive Trend Analysis System for Process monitoring and diagnosis. *American Control Conference*, Albuquerque, USA, 1997.
- (Wang, 2003a) Wang L., Tan T., Hu W., Ning H., Automatic gait recognition based on statistical shape analysis. *IEEE Trans Image Process*. 12 (9), pp. 1120-1131, 2003.
- (Wang, 2003b) Wang H., Fan W., Yu P.S., Han J., Mining concept-drifting data streams using ensemble classifiers. *KDD 03*, 2003.
- (Webb, 2002) Webb A., *Statistical pattern recognition*, 2nd ed. John Wiley and Sons, 2002.
- (Widmer, 1996) Widmer G., Kubat M., Learning in the Presence of Context Drift and Hidden Contexts, *Machine Learning* 23, pp. 69-101, 1996.
- (Wu, 2007) Wu J., Wang J., Liu L., Feature extraction via KPCA for classification of gait patterns. *Hum Mov Sci*. 26 (3), pp. 393-411, 2007.
- (Xie, 1991) Xie X.L., Beni G., A validity measure for fuzzy clustering. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 12:pp.841-847, 1991.
- (Xu, 2007) Xu D., Yan S., Tao D., Lin S., Zhang H.J., Marginal Fisher analysis and its variants for human gait recognition and content-based image retrieval. *IEEE Trans Image Process* 16 (11), pp. 2811-2821, 2007.
- (Yager, 2006) Yager R.R., An extension of the naive Bayesian classifier, *Information Sciences* 176 (5), pp. 577-588, 2006.
- (Yang, 2010) Yang G., Lin Y., Bhattacharya P., A driver fatigue recognition model based on information fusion and dynamic Bayesian network. *Information Sciences* 180 (10), pp. 1942-1954, 2010.
- (Yazdani, 1996) Yazdani N., Ozsoyoglu Z.M., Sequence Matching of Images. 8th *International Conference on Scientific and Statistical Database Management*, Stockholm, pp. 53-62, 1996.
- (Ye, 2002) Ye N., Li X., A scalable, incremental learning algorithm for classification problems. *Computers and Industrial Engineering*, 43(4), pp. 677-692, 2002.

- (Zemouri, 2003) Zemouri R., *Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance*. Doctorat de l'Université de Franche-Comté, 2003.
- (Zhang, 2002) Zhang B., Srihari S.N., A Fast Algorithm for Finding k-Nearest Neighbors with Non-metric Dissimilarity. *8th International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, 2002.
- (Zhou, 2001) Zhou S., *Incremental Document Classification in a Knowledge Management Environment*, Thesis, University of Toronto, 2001.

Résumé

De nombreux systèmes actuels sont évolutifs, i.e. leur comportement est dynamique et il entraîne des changements dans leurs caractéristiques de fonctionnement. Le suivi des modes de fonctionnement des systèmes évolutifs est un problème majeur pour les méthodes de diagnostic. En effet, dans ces conditions il est nécessaire d'utiliser ou de développer des méthodes tenant compte des nouvelles informations caractéristiques du comportement actuel d'un système et permettant l'adaptation des modes de fonctionnement connus. Nous avons choisi de travailler avec les méthodes de reconnaissance des formes pour leur capacité à fonctionner sur des applications pour lesquelles seules des observations sont connues. Plusieurs méthodes de classification dynamique ont été proposées afin de tenir compte des changements de caractéristiques des formes et des classes au cours du temps. D'une part la méthode Fuzzy Pattern Matching Dynamique (FPMD) a été développée pour tenir compte du changement graduel des caractéristiques des classes après la classification de chaque nouvelle forme. La méthode intègre plusieurs mécanismes tels que des indices de représentativité des données, un résidu permettant de suivre les évolutions des classes, et des procédures de scission et de fusion permettant d'adapter les classes dynamiques. D'autre part, une version Supervisée et une version Semi-Supervisée des K-Plus Proches Voisins Flous Dynamique (KPPVFD) ont également été proposées : KPPVFD-S et KPPVFD-SS. Ces méthodes KPPVFD utilisent des procédures de détection et de confirmation des évolutions de classes puis elles réalisent l'adaptation de ces dernières en utilisant les formes les plus caractéristiques de la tendance actuelle du système. Les méthodes proposées permettent de détecter en ligne l'évolution du comportement d'un système, de valider cette évolution et de procéder à l'adaptation d'une classe lorsque ses caractéristiques ont changées. Deux approches de reconnaissance des formes (structurelle et mixte) ont également été proposées. L'approche structurelle repose sur une méthode de segmentation ne nécessitant pas la définition d'un seuil d'erreur d'approximation et sur un nombre adaptatif de primitives défini par rapport à chaque phase caractéristique d'une forme dynamique. Une fois la segmentation des formes dynamiques réalisée, la méthode estime leur tendance à l'aide des primitives sélectionnées puis la phase de classification peut avoir lieu en utilisant une mesure de similarité. La méthode mixte repose sur l'utilisation de données statistiques et structurelles

pour réaliser la classification des données dynamiques. Une signature mixte est obtenue pour chaque forme dynamique traitée. Cette signature mixte permet d'obtenir une information caractéristique interprétable. Une mesure de similarité mixte, basée à la fois sur la similarité entre caractéristiques structurelles et statistiques, est ensuite calculée à partir de cette signature mixte pour mesurer l'appartenance d'une forme à une classe. Cette mesure permet également de quantifier l'évolution que peut réaliser une forme suite à un changement de caractéristiques d'un système. L'ensemble des méthodes proposées a été utilisé sur plusieurs applications simulées et réelles. Ces applications concernaient le milieu industriel (détection de soudures de mauvaise qualité, détection d'une fuite dans un générateur de vapeur) et le milieu médical (caractérisation de la coordination inter-segmentaire des patrons de marche des patients hémiparétiques).

Mots clés : apprentissage en ligne, classification dynamique, données non statiques, reconnaissance des formes, environnement dynamique, systèmes évolutifs.

Abstract

Several systems are evolving, i.e. their behavior is dynamic and it leads to changes in their functioning characteristics. The monitoring of the evolving systems functioning modes is a major problem for the diagnosis methods. Indeed, in these conditions it is necessary to use or to develop methods taking into account the new characteristic information of the current system behavior and permitting to adapt the known functioning modes. We have chosen to use pattern recognition methods for their ability to work on applications for which they only have observations. Several dynamic pattern recognition methods have been proposed in order to take into account of the changes realized in patterns and classes characteristics according to the time. The method Dynamic Fuzzy Pattern Matching (DFPM) has been developed to consider the gradual change of classes characteristics after the classification of each new pattern. The method integrates several mechanisms such as indicators of the patterns usefulness, a residual permitting to follow classes evolutions, and splitting and merging procedures in order to adapt dynamic classes. Several versions of the Dynamic Fuzzy K-Nearest Neighbours (DFKNN) have also been proposed: Supervised-DFKNN (S-DFKNN) and Semi-Supervised DFKNN (SS-DFKNN). These DFKNN methods use procedures to detect and confirm classes evolutions. Then, they realize the adaptation of the evolved classes by using the most characteristics patterns of the current trend of a system. The proposed methods permit to online detect the evolution of a system behavior, to validate this evolution and to proceed to the adaptation of a class when its characteristics have changed. Two others approaches of pattern recognition (structural and mixed) have also been proposed to deal with dynamic patterns. The structural approach is based on a segmentation method which does not necessitate the definition of an approximation error threshold, and on an adaptive number of primitives defined according to each characteristic phase of a dynamic pattern. Once, the dynamic patterns segmentation is realized, the method estimates their trend using the selected primitives. Then, the classification of all patterns can be realized using a measure of similarity. The mixed pattern recognition proposed method is based on the use of statistical and structural characteristics to realize the classification of dynamic patterns. A mixed signature is obtained for each dynamic pattern treated. This signature permits to obtain characteristic and interpretable information. A mixed similarity measure, based on structural

and statistical characteristics, is then calculated from the mixed signature, to measure the membership of a pattern to a class. This measure also permits to quantify the evolution a pattern can realize after a change in the characteristics of a system. The set of proposed methods has been used on several simulated and real applications. These applications concerned the industrial domain (detection of a welding's quality, detection of a leak in a steam generator) and the clinical domain (characterization of the inter-segmental coordination of the gait patterns for hemiparetic patients).

Keywords: online learning, dynamic classification, non stationary data, pattern recognition, dynamic environment, evolving systems.