,pdfcreator=HAL,pdfproducer=PDFLaTeX,pdfsubject=Computer Science [cs]/Human-Computer Interaction [cs.HC]

Adaptations et applications de modèles mixtes de réseaux de neurones à un processus industriel

Georges Schutz

▶ **To cite this version:**

Georges Schutz. Adaptations et applications de modèles mixtes de réseaux de neurones à un processus industriel. Interface homme-machine [cs.HC]. Université Henri Poincaré - Nancy I, 2006. Français. NNT : . tel-00115770

HAL Id: tel-00115770
https://theses.hal.science/tel-00115770

Submitted on 23 Nov 2006

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.

Département de formation doctorale en informatique **École doctorale IAEM Lorraine**
UFR STMIA

Adaptations et applications de modèles mixtes de réseaux de neurones à un processus industriel

THÈSE

présentée et soutenue publiquement le 5 octobre 2006

pour l'obtention du

Doctorat de l'université Henri Poincaré – Nancy 1

(spécialité informatique)

par

Georges Schutz

Composition du jury

Président :	...	
Rapporteurs :	Louis WEHENKEL	Professeur, Université de Liège, Belgique
	Gilles VAUCHER	Professeur, Supélec, Campus de Rennes, France
	Claude GODART	Professeur, Université Henri Poincaré Nancy 1, France
Examinateurs :	Frédéric ALEXANDRE	Directeur de recherche, INRIA-Lorraine, France
	Serge GILLÉ	Ingénieur de recherche, CRP Henri Tudor, Luxembourg
	Jean-Claude BAUMERT	Ingénieur de recherche, ProfilARBED Recherche, Luxembourg

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thloria.

Remerciements

En premier lieu j'aimerais remercier les deux personnes qui ont encadré ce travail :
Frédéric ALEXANDRE, mon directeur de thèse et responsable de l'équipe Cortex au LORIA de Nancy et Serge GILLÉ du Centre de Recherche Public Henri Tudor au Luxembourg et responsable de l'équipe d'accueil MODSI. Il m'ont aidé par de nombreuses discussions et par beaucoup de soutien scientifique et moral.

Un merci va également à Jean-Claude BAUMERT, responsable du projet de recherche chez ProfilARBED Recherche, pour la collaboration et les aides au niveau de l'application industrielle.

Je tiens aussi à remercier les membres du jury : Louis WEHENKEL, Professeur à l'Université de Liège, Gilles VAUCHER, Professeur à Supélec et Claude GODART, Professeur à l'Université Henri Poincaré de Nancy.

Même si durant toutes ces années de thèse je n'étais que rarement présent à Nancy, je remercie les membres de l'équipe Cortex pour leur accueil chaleureux et les discussions fructueuses.

Etant membre de l'équipe de modélisation et de simulation au LTI à Esch sur Alzette, j'étais toujours épaulé par Serge, Émmanuelle, David, Gaston et Salime. Un grand merci aussi à Frank, Daniel et Oli. Pendant de nombreuses discussions non pas purement scientifiques, on a passé des moments agréables. J'aimerais remercier Jos SCHÄFFERS, directeur du Laboratoire des Technologies Industrielles, pour m'avoir réservé une place au sein de son laboratoire.

Diverses personnes ont contribué directement dans le cadre de leur travail de fin d'études à ma thèse et j'aimerais les remercier : Magalie LEMASSON, David MARCUS et Natacha SCHAAF.

Je ne veux pas oublier ici le personnel de ProfilARBED Recherche et ProfilARBED Esch-Belval sans lequel ce travail n'aurait pas été réalisable.

Un grand merci aussi à Annie MICHAELY, Carine et Gaston SCHUTZ pour leur aide dans le domaine de la grammaire et de l'orthographie française.

Je remercie aussi l'État Luxembourgeois pour m'avoir accordé une bourse formation recherche.

Il est évident que j'ai oublié l'une ou l'autre personne et je les prie de bien vouloir m'en excuser.

Je dédie cette thèse à mes parents Carine et Gaston SCHUTZ, qui ont fait tout leur possible pour me donner l'opportunité de faire des études,
à ma conjointe Frutz, qui m'a soutenu pendant cette période.

Table des matières

Introduction

Des analyses internationales réalisées par Tufte [58] montrent que 75% des graphiques présentés dans les périodiques renommés représentent des séries temporelles. Ces graphiques traitent pratiquement tous les domaines, qu'ils soient de nature scientifique, économique ou encore privée. Les séquences temporelles d'informations numériques sont aussi un sujet d'actualité dans beaucoup de domaines de recherche, entre autres ceux de la biologie, de la chimie avec comme exemple les réactions bio-chimiques, de la médecine (signaux cardiaques). On les retrouve également dans les domaines technologiques par exemple l'automatisation, ou encore dans les domaines plus fondamentaux tels les mathématiques ou les sciences informatiques (traitement d'information).

Indépendamment de l'application industrielle spécifique qui sera analysée dans le contexte de ce travail, la dimension temporelle est un aspect primaire de beaucoup de processus industriels pour qui aussi bien le contexte actuel que l'ordre et l'historique d'événements antérieurs sont capitaux. L'évolution temporelle et la dimension temporelle en globalité sont des sujets académiques permanents, ce qui est souligné par l'étude de Tufte.

Keogh [30] soulève les propriétés suivantes comme étant des problématiques spécifiques auxquelles il faut se préparer en abordant le traitement de données temporelles. Elles seront mises en contexte dans la suite.

- Il faut savoir manipuler de gros volumes de données de façon efficace.
  Il donne les exemples suivants : le volume de données enregistré pour une heure d'ECG (électrocardiogramme) est de "1 Gigabyte", l'enregistrement du suivi d'un serveur web standard est de "5 Gigabytes" par semaine et la base de données du Space Shuttle est de "158 Gigabytes".
  La méthode la plus répandue pour affronter cette propriété est la réduction de données en utilisant des codages appropriés. Le défi, dans la réduction du volume de données, consiste à augmenter l'efficacité du traitement de ces dernières, tout en évitant de perdre des informations substantielles pour des utilisations ultérieures.
- Il faut être prêt à affronter le thème de la subjectivité.
  L'interprétation de l'aspect temps est différente pour chaque domaine d'application, pour chaque tâche concrète et même pour chaque individu. Cela nécessite une grande flexibilité d'adaptation pour des méthodes qui ont l'ambition d'être génériques. Selon Keogh, c'est une des raisons pour laquelle des outils ou méthodes peuvent donner d'excellents résultats dans un certain domaine et en même temps ne pas être utilisables dans un autre contexte.
  Du point de vue scientifique et technologique, cela peut poser des problèmes au niveau de l'évaluation des approches, puisque souvent les performances de différentes approches ne sont pas vraiment comparables.
- Il faut aborder les domaines de l'acquisition et de la gestion de données temporelles.
  Fusionner divers formats d'information, analyser des mesures échantillonnées de manière différente, maîtriser des données bruitées ou contenant des valeurs manquantes voire erronées sont des problématiques classiques et inévitables dans ce domaine. Il faut en être conscient et savoir gérer ce genre de problèmes qui peuvent avoir des influences non-négligeables sur les résultats d'analyses.

Dans ce travail, ces sujets se retrouvent à différents niveaux et vont être soulevés dans la suite.

De façon générale, la dimension temporelle est un sujet important dans le domaine connexionniste, qui sera spécialement analysé dans ce travail et qui s'intéresse aussi de manière plus fondamentale à cette dimension. En effet, le système nerveux, qui depuis toujours inspire ce domaine, est un système dynamique avec des processus hautement parallélisés pour lequel il est bien connu que les aspects temporels jouent un

rôle important à différents niveaux du système. Cependant, incorporer cette dimension temporelle dans des approches connexionnistes n'est pas une tâche triviale et parmi les diverses approches actuellement disponibles, chacune a ses avantages et ses désavantages. Les réseaux, dits dynamiques, ou réseaux à spikes, ne sont pas pris en compte dans cette étude. L'étude se limite aux approches spatiotemporelles qui sont basées sur la représentation spatiale du temps. Ces réseaux de neurones sont connus pour leur robustesse au bruit et leurs propriétés non-linéaires. Il est donc important de savoir adapter cette gamme de modèles à ces phénomènes si importants et encore mal maîtrisés, les séquences temporelles.

A côté de cette dimension temporelle un autre sujet sera particulièrement abordé dans ce travail. Il est question de la classification non-supervisée dans le contexte de l'extraction de connaissances. Dans notre travail, la classification se retrouve à deux niveaux : le premier se situe au niveau du codage où cette méthode est utilisée afin de réaliser un codage qui s'adapte de façon non-supervisée aux signaux analysés. A ce premier niveau, une méthode bien connue du domaine connexionniste est utilisée, il s'agit de la carte auto-organisatrice (SOM) de Kohonen [36]. Le deuxième recours à la classification non-supervisée se retrouve au niveau de la recherche de comportements typiques dans des séquences temporelles. Sans aller trop dans le détail ici, une particularité à ce deuxième niveau est que les séries temporelles qu'il s'agit de répertorier peuvent avoir des tailles arbitraires et qu'en plus, une certaine souplesse temporelle dans la comparaison des évolutions est indispensable.

Dans le cas présent, les données proviennent de mesures du monde réel effectuées dans un environnement de l'industrie lourde sur lequel nous allons revenir. Or ce genre de milieu est souvent hostile au traitement et au transfert de signaux et entraîne des données bruitées. De plus il est difficile de garantir la fiabilité des mesures, puisque des capteurs peuvent ou se dégrader ou mal fonctionner pendant des périodes de temps arbitraires. A ce niveau on peut faire le lien avec la problématique de l'acquisition et de la gestion de données temporelles évoquée par Keogh.

Une autre difficulté à laquelle on peut être confronté dans le domaine du traitement de séquences temporelles est la variation en taille et en forme de telles séquences, même si elles représentent des phénomènes similaires. Ici la problématique de la subjectivité évoquée par Keogh se retrouve. Afin d'illustrer la situation voici un exemple simple : faire bouillir de l'eau en utilisant un réchaud au gaz. Cet exemple est utilisé à cause de sa proximité au processus analysé dans le contexte de ce travail. Supposons que les variables mesurées sont d'un côté une mesure temporelle (le débit de gaz tout au long du processus) et d'un autre côté deux mesures ponctuelles (la température et la quantité d'eau bouillante à la fin du processus). Le résultat de ce processus dépend de plusieurs paramètres : les plus importants peuvent être la quantité et la température initiale de l'eau qu'il s'agit de réchauffer, la qualité du gaz utilisé et le type de réchaud ainsi que le récipient utilisé. Une température initiale faible et une quantité d'eau plus importante vont prolonger le processus. Un gaz avec une puissance calorifique plus élevée va diminuer la durée du processus. Mais il y en a d'autres comme la composition de l'eau, la température ambiante ou encore la température initiale du récipient ainsi que les paramètres calorifiques de ce dernier. Une eau moins salée va prolonger le processus tandis qu'une température ambiante plus élevée réduit la durée du processus. Le phénomène représenté "faire bouillir de l'eau" est toujours le même mais la durée du processus, la consommation de gaz ainsi que la température et la quantité finale d'eau dépendent fortement des paramètres. Dans un environnement de type laboratoire, on peut supposer qu'il est possible de contrôler tous ces paramètres afin de produire un processus plus ou moins semblable et donc reproductible. Cependant dans un environnement industriel cela n'est que difficilement réalisable car le fournisseur de l'eau et celui du gaz sont dépendants du marché et en plus la température ambiante dans un hall industriel varie avec les saisons et avec la météo. Selon les circonstances, il est imaginable qu'il n'est même pas possible d'atteindre la température nécessaire pour faire bouillir l'eau.

Surtout, dans un tel contexte industriel, nous n'avons pas nécessairement connaisance de l'ensemble des variables pertinentes ni des lois physiques complexes qui les relient. Chercher à acquérir et représenter ces variables et découvrir les relations temporelles qui les relient sera donc préalable à toute proposition de fonctionnement.

Dans ce travail nous allons nous intéresser à un aspect spécifique du traitement de ces signaux qui est souvent décrit avec le terme de fouille de données. Ce terme, même s'il est abusivement utilisé dans divers contextes, représente bien de quoi il est question dans ce travail.

Nous cherchons des méthodes pour aborder les problèmes de fouille de données temporelles afin d'extraire de ces données des connaissances sur la nature, le comportement et les propriétés du processus

sous-jacent qui est la source de ces données. Nous allons étudier l'application de ces méthodes avec un partenaire industriel en nous basant sur un processus industriel relativement complexe.

Au niveau applicatif, le but est d'aider le partenaire industriel d'un côté à mieux comprendre le comportement de son installation et d'un autre côté à utiliser cette connaissance afin de lui fournir des outils en vue de pouvoir réduire les coûts de fonctionnement de son installation.

Un des objectifs de tout travail de thèse est que les méthodes qui seront proposées et appliquées soient aussi génériques que possible et de ce fait puissent avoir un intérêt au-delà du travail et de l'application utilisée pour analyser et valider les méthodes.

Contexte et positionnement du travail de thèse

Le travail de thèse qui est présenté a été réalisé dans le contexte d'un projet de recherche industriel subventionné par la CECA[1], auquel quatre partenaires européens ont participé. Le projet sera décrit dans le chapitre 1, mais afin de pouvoir positionner ce travail, quelques détails du projet sont utiles. Dans la définition du projet, quatre objectifs principaux étaient spécifiés : le développement de connaissances sur le comportement d'une installation sidérurgique, la reproductibilité du processus, l'amélioration du contrôle et la minimisation de coûts de fonctionnement de l'installation. Afin d'atteindre ces buts, il a été spécifié que le processus sera approché par différentes méthodes en parallèle. Une des méthodes spécifiées dans la définition du projet était les réseaux de neurones ou plus globalement les approches connexionnistes. C'est pour cette raison que le "Centre de Recherche Public Henri Tudor" (Luxembourg), un des partenaires du projet, a élaboré un contrat de sous-traitance avec l'équipe CORTEX du "Laboratoire Lorrain de Recherche en Informatique et ses Applications" (LORIA) de Nancy (France). Issu de cette collaboration est entre autre le travail de thèse qui est exposé dans ce manuscrit.

Deux sujets principaux de ce travail de thèse découlent directement des objectifs et de l'orientation méthodologique de la partie du projet abordée dans cette étude. Il s'agit du développement de connaissances et des approches connexionnistes. Cela oriente ce travail vers un domaine spécifique de l'intelligence artificielle qui est caractérisé par les mots clefs comme l'extraction de connaissances, la classification non-supervisée et le connexionnisme.

Un troisième sujet qui joue un rôle majeur dans ce travail découle des propriétés du processus industriel qui est analysé dans le cadre du projet. En effet, il s'agit d'un processus cyclique qui se situe au début d'une chaîne de production d'acier basée sur le recyclage de métaux. Il est question d'un processus de fusion de mitrailles afin de produire de l'acier liquide qui est utilisé dans la suite de la chaîne de production. Des observations ainsi que des discussions avec les experts de l'installation indiquent que le comportement du four dépend fortement de l'évolution temporelle de l'installation dans sa totalité. Une dépendance de l'historique à court, moyen et même à long terme a été constatée aussi bien par le personnel de l'installation que par des analyses réalisées par un des partenaires dans le contexte du projet CECA voire dans des analyses préalables sur d'autres installations semblables. Le temps joue de ce fait un rôle important dans ce travail. Ce sujet va se retrouver dans l'analyse du comportement et de l'évolution dynamique du processus et peut être caractérisé par les mots clefs comme l'analyse de séries temporelles, l'analyse des aspects dynamiques ou l'évolution temporelle d'un processus.

Finalement, un objectif non négligeable pour le partenaire industriel est la reproductibilité du processus et la minimisation des coûts de fonctionnement. Cela conduit vers des aspects comme la définition d'un comportement similaire voire la similarité entre séries temporelles d'un côté et l'optimisation ou encore le contrôle et la conduite de processus de l'autre.

A ces aspects plutôt scientifiques s'ajoutent des aspects d'ordre technologique et organisationnel sous forme de propriétés ou de contraintes qui sont reliées au processus industriel. L'installation est conduite 24h/24 par plusieurs équipes qui se relaient. L'opérateur et les autres membres de son équipe ont pour objectif de garantir une production d'acier liquide afin d'alimenter les procédés en aval de la chaîne de production qui en dépendent. De ce fait la cadence, le volume, la qualité et la température de l'acier liquide qui doit être produit ne sont pas stables mais dépendent de la production globale et de ce fait aussi des processus en aval de la chaîne de production. Toute interruption prolongée de la production

[1] Communauté Européenne du Charbon et de l'Acier

d'acier liquide a des répercutions sur toute la chaîne de production et de ce fait représente un coût très important et est donc à éviter à tout prix.

En ce qui concerne l'aspect de contrôle ou de conduite de l'installation, il est important de noter qu'il n'existe pas de modèle déterministe du processus. Il faut noter cependant que, dans le cadre du projet CECA, un autre partenaire était chargé d'adapter un tel modèle au processus en question. La complexité de ce genre de modèle et la paramétrisation qui en découle se sont avérées être très difficile. Le manque d'informations spécifiques sur le processus et la nature bruitée et peu fiable des données existantes ont rendu la tâche encore plus complexe. Ce constat n'a fait qu'encourager la recherche de solutions alternatives, comme celle que représente l'extraction de connaissances à partir de bases de données.

Actuellement, le suivi et la conduite du processus se basent principalement sur la consommation en énergie électrique qui représente l'apport primaire en énergie. L'opérateur, dans la commande du processus, est guidé par une consigne statique qui donne de manière schématisée le niveau de la puissance électrique en fonction de l'énergie électrique consommée tout au long d'un cycle de production.

Selon les experts, de nombreuses variables, soit inconnues soit non mesurables de façon fiable, ont une influence non négligeable sur le comportement du processus. Celles-ci sont supposées être la cause du comportement imprévu de l'installation. L'opérateur doit dans ces situations agir, voire réagir, tout au long de l'évolution du cycle de production afin de stabiliser le processus et d'éviter des endommagements de l'installation. En fonction de son expérience, l'opérateur sait plus ou moins bien comment réagir dans de telles situations. Dans son analyse, le comportement de l'installation et la réaction du processus sur des changements de commandes dépendent en premier lieu de la situation actuelle et de l'historique à court terme de l'installation comme par exemple la répartition des températures des différentes zones de refroidissement, ou encore l'avancement et l'évolution dans un cycle de production (problèmes préalables, stabilisation ou non de l'une ou l'autre phase de fusion etc.). Mais le comportement et la réaction dépendent aussi de l'historique à moyen et à long terme de l'installation, comme par exemple la durée depuis le dernier arrêt de maintenance, et entre autres des conditions climatiques qui ont une influence sur la mitraille, la matière primaire du processus, stockée à l'air libre. Ainsi, pour être capable de conduire l'installation, l'opérateur a un modèle interne du processus et de l'installation qu'il détient en grande partie des observations qu'il a pu faire du comportement de l'installation.

Si tel est le cas, il doit être possible, en utilisant des méthodes d'extraction de connaissances, de faire émerger des propriétés concernant le comportement du processus en nous basant sur des observations, en l'occurrence les mesures réalisées sur l'installation. On voit comment, présenté de cette manière, cela devient un problème de fouille de données temporelles.

L'hypothèse principale et l'approche qui en découle

Des méthodes de fouille de données temporelles appliquées à des données représentant un comportement d'un processus physique devraient avoir la capacité d'extraire des connaissances sur les propriétés dynamiques et évolutives du processus sous-jacent. En se basant sur ces connaissances, il est d'un côté envisageable de pouvoir prédire le comportement du processus en connaissant un certain historique courant de ce dernier ; et de l'autre côté, une corrélation des propriétés évolutives avec d'autres propriétés du processus peuvent donner des informations supplémentaires pour l'interaction avec le processus.

L'observation du comportement de l'installation aussi bien que la discussion avec les différents acteurs humains de l'installation ainsi que l'hypothèse générale formulée ci-dessus ont conduit à l'hypothèse suivante :

> Des comportements typiques de l'installation existent. La consommation globale de l'installation dépend fortement de l'évolution temporelle avec laquelle le processus est alimenté en ressources. En d'autres mots : il existe une relation entre la consommation globale du processus et l'évolution temporelle de l'alimentation et le comportement du processus.

Cela conduit directement à la supposition que, durant l'évolution du processus, il existe des instants déterminants qui influent sur le résultat global. Même si cela a semblé être évident pour les opérateurs, peu de démarches ont été faites jusqu'à présent afin d'exploiter ce principe.

Basé sur l'hypothèse qu'il existe des comportements typiques de l'installation, nous allons par des méthodes de fouille de données temporelles extraire ces évolutions types du processus. Basé sur l'hypothèse qu'il existe une relation entre la consommation et l'évolution temporelle, il est envisageable de pouvoir utiliser la connaissance sous forme d'évolutions type afin de donner un retour à l'opérateur en vue de pouvoir utiliser cette information pour améliorer la conduite du four. En nous basant sur le mode utilisé actuellement pour la conduite du processus, c.-à-d. l'opérateur qui suit au mieux une consigne statique, une idée d'un contrôle prédictif basé sur l'aide à la décision a été dégagée.

Les démarches théoriques et applicatives qui sont proposées dans cette thèse essaient de fournir en premier lieu les moyens pour analyser l'hypothèse posée en faisant émerger des données, des comportements typiques du processus et en analysant ces comportements par rapport à la consommation principale. Dans un second volet sera analysée une approche par laquelle une proposition de scénarios prédictifs tout au long de l'évolution du cycle de production serait réalisable. Même si une réelle implémentation en ligne des méthodes n'a pas pu être réalisée dans le cadre de cette thèse, il sera montré sur base de quelques exemples réels qu'en interaction avec les connaissances des opérateurs et les résultats des analyses proposées, une perspective pour une réduction de la consommation en énergie électrique est envisageable.

La démarche est divisée en deux parties. Une première partie d'analyse, dans laquelle l'hypothèse sera analysée, et durant laquelle les moyens seront développés dans l'optique d'approcher et de valider cette hypothèse. Cette partie est basée sur les données provenant de l'installation et développe les méthodes et outils nécessaires pour pouvoir extraire de ces données les comportements évolutifs typiques du processus. Dans la deuxième partie il s'agit d'utiliser ces évolutions type en vue de proposer à l'opérateur, au fur et à mesure de l'avancement d'un cycle de production, des scénarios pour la suite du processus qui, à chaque instant, prend en compte l'évolution passée du cycle actuel et qui serait capable de considérer des critères d'optimisation. L'approche proposée s'aligne au mode de conduite utilisé actuellement en utilisant des scénarios sous forme de consignes temporelles. Il est prévu que c'est l'opérateur qui ferme la boucle de contrôle, en agissant sur les paramètres de l'installation en vue de réaliser l'évolution proposée. La différence avec la méthode basée sur une consigne statique est que la proposition issue de l'approche prédictive pourra s'adapter constamment à la situation actuelle en tenant compte de l'historique du cycle en cours. De plus, le choix du scénario à suivre pourra se faire sur base de différents critères qui pourront eux aussi changer durant le cycle de production.

Les sujets principaux abordés dans la première partie sont l'analyse de séries temporelles, l'extraction de connaissances de façon non-supervisée et le codage de données dans le contexte de la réduction du volume de données, tout en focalisant les méthodes connexionnistes. Avec la relation "séries temporelles" et "extraction de connaissances", les sujets comme la similarité entre séries temporelles et la comparaison de séries de taille arbitraire vont être abordés.

Durant la deuxième partie les sujets tels que la localisation de séquences dans des séries temporelles et l'interaction avec l'acteur humain seront discutés et des résultats sur base de données réelles seront présentés. En ce qui concerne l'incorporation de critères en vue d'aider dans le choix du scénario, des possibilités seront évoquées sans pour autant discuter en détail cette question.

Plan du mémoire

Ce mémoire commence par un chapitre qui présente le projet de recherche industriel et pose le cadre de ce travail de thèse. Dans ce même chapitre le processus et l'installation industrielle sont présentés. Il ne s'agit pas seulement d'une description du fonctionnement du processus et de la mécanique de l'installation, mais aussi d'une présentation des acteurs humains autour de cette installation, ainsi que des personnes activement intégrées dans cette étude. De plus, à ce niveau, le sujet évoqué par Keogh concernant la diversité des informations sur le processus, ainsi que l'acquisition et la gestion des données mesurées, les différent formats, les données bruitées, manquantes ou non valides seront abordées. Pour le lecteur ce chapitre est important afin de se familiariser avec les propriétés spécifiques de l'installation et celles des données qui seront analysées dans la suite du document.

Après la présentation de l'application industrielle et de ses propriétés, le premier chapitre conclut avec un aperçu plus détaillé du concept de contrôle prédictif basé sur l'aide à la décision, une des bases de cette étude. Une illustration schématisée des deux parties principales de l'approche est proposée.

Le deuxième chapitre est réservé aux lecteurs non initiés au domaine du connexionnisme. Sans avoir l'ambition de donner un aperçu global et approfondi du domaine, ce qui est réalisé beaucoup mieux par de nombreux ouvrages comme par exemple [21],[17], le chapitre se limite aux aspects directement liés aux sujets qui seront abordés dans la suite du manuscrit.

Après ces deux chapitres de mise en contexte, les deux chapitres suivants sont consacrés à l'explication et la mise au point des méthodes qui pourront former une approche de contrôle prédictif basée sur l'aide à la décision. Toutes ces démarches seront appliquées en premier lieu à des données artificielles en vue d'analyser les propriétés et de valider les méthodes. Ensuite, elles sont appliquées aux données provenant de l'installation industrielle.

Le troisième chapitre traite la première partie de notre approche. Les sujets abordés sont l'extraction non-supervisée des comportements typiques du processus sur base d'analyses de séries temporelles, qui représentent l'évolution temporelle des cycles de production. Au début de ce chapitre de plus amples détails seront donnés sur les spécificités des séries temporelles. Auparavant la classification non-supervisée des comportements typiques, le sujet du codage aussi évoqué par Keogh, sera abordé dans le but de la réduction de données. Dans cette partie une proposition d'une extension d'un codage classique, l'approximation par parties constantes (PAA), sera réalisée. Cette extension propose d'attribuer des formes caractéristiques à des parties de signal afin de conserver plus de détails des parties codées.

La réduction du volume de données joue un rôle, parce que les temps de calcul des méthodes d'analyse pour l'extraction de connaissances sont pour la plupart directement liés au volume de données qui doit être traité. Le volume global de données à manipuler dans le cadre de ce travail est abordable avec une base de données de plus ou moins "6 Gigabytes" de données provenant d'un processus industriel, dont seulement un sous-ensemble est utilisé ici. Notez cependant d'une part, que cette base ne représente qu'une demi-année de suivi de l'installation et donc pas nécessairement toutes les conditions de travail de l'installation. D'autre part, les implémentations des méthodes et les outils ainsi que la plate-forme utilisée dans le cadre de ce travail ne sont pas spécialement optimisés sur le point de vue performance de calcul. Il s'agit plutôt d'une approche de prototypage rapide ou d'une démonstration de faisabilité pour laquelle la performance de calcul ne représente qu'un aspect secondaire. La réduction de volume de données devient donc nécessaire afin d'améliorer les conditions de recherches et d'analyses.

Basée sur ce codage, une méthode de classification non-supervisée capable de prendre en compte des séries temporelles de taille arbitraire sera proposée. Cette méthode se base sur une mesure de dissimilarité, qui utilise un algorithme du domaine de la programmation dynamique, le Dynamic Time Warping (DTW). Cet algorithme sera analysé en détail et son utilisation dans le contexte d'une méthode de classification sera présenté. À ce niveau la subjectivité évoquée par Keogh se retrouve dans le choix et dans la paramétrisation des méthodes utilisées, mais aussi au niveau de l'interprétation des résultats obtenus. L'application de la classification aux données du processus industriel permet de produire un corpus de comportements typiques, qui sont nécessaires pour la deuxième partie de l'approche proposée.

Dans le quatrième chapitre, la deuxième partie de l'approche est abordée. Une proposition pour une approche de contrôle prédictif basée sur une aide à la décision est développée. Pour cela, le sujet de la classification en ligne, qui traite le domaine de la localisation de séquences dans des séries temporelles, sera abordé. Après la mise au point d'une méthode de localisation basée sur la méthode DTW et bornée d'un côté, la thématique de proposition de scénarios appropriés est analysée. Dans ce contexte, la prise en compte de critères d'optimisation et le potentiel de réduction de la consommation globale seront discutés. Faute d'implémentation sur le site industriel, cette partie peut être vue comme spéculative. Mais sur base de quelques exemples réels, une première illustration du potentiel est réalisée, même si l'analyse du potentiel réel n'est réalisable que dans le contexte d'une boucle fermée.

Le cinquième chapitre présente les conclusions et les perspectives de ce travail de thèse. Les différents résultats et contributions scientifiques engendrés par ce travail sont repris et brièvement discutés afin d'illustrer les éventuelles applications et extensions possibles.

Dans les trois annexes, quelques détails sont donnés concernant le système d'acquisition et de gestion de données, l'apprentissage dans le contexte des cartes auto-organisatrices (SOM) et deux méthodes de validation du codage proposé.

Chapitre 1

Le projet industriel

Sommaire

La présente thèse s'est déroulée en relation avec un projet européen financé par la CECA[2]. Dans ce chapitre le projet européen est présenté brièvement et le rapport avec le sujet de la thèse sera évoqué.

Le comportement de l'installation industrielle et le déroulement des principaux processus seront expliqués. Une excellente connaissance de l'installation et des processus est primordiale afin de mener à bien un tel projet. Cette affirmation peut être motivée par le besoin de communication avec les experts de différents niveaux, le choix de méthodes d'analyses ainsi que l'interprétation des résultats obtenus voire l'argumentation des orientations de la recherche durant le projet.

Les informations sur une installation industrielle sont diverses. Un aperçu sur cette diversité est donné avec une focalisation sur l'acquisition et la gestion des données provenant de l'installation. Une partie de ces données formeront la base des analyses ultérieures.

Pour le lecteur, ce chapitre est intéressant parce qu'il fournit d'importantes informations afin de comprendre la suite de la thèse et les démarches qui y sont faites. Cependant diverses remarques sur la manière comment ce projet a été abordé semblent classiques voire de nature générale pour tous projets de ce genre.

Dans la dernière section de ce chapitre, une fois que le lecteur s'est familiarisé avec les spécificités du processus industriel, le concept global de l'approche qui est proposée par cette thèse est présenté.

1.1 Le projet européen

Le projet CECA, 7210-PR-129, s'intitule "Improved Control of Electric Arc Furnace Operations by Process Modelling". Le projet a débuté le 1.7.1999 et la fin était prévue pour le 30.6.2002. À cause de diverses raisons le projet a été prolongé d'une année du 1.7.2002 au 30.6.2003. Le rapport final du projet était présenté en mai 2004.

L'équipe CORTEX du LORIA[3] de Nancy participe à ce projet en tant que sous-traitant du CRP Henri Tudor[4] du Luxembourg, partenaire du projet. Les autres partenaires du projet sont : le coordinateur ProfilARBED Recherche[5], ACERALIA[6] et le CRM[7].

L'objectif du projet est le développement des connaissances et l'amélioration du contrôle du processus complet de la production d'acier liquide dans les fours à arc électrique, en combinant trois approches de modélisation des processus :

- un modèle objectif classique de la métallurgie basé sur des bilans de type énergétique, massique et chimique.
- un modèle subjectif s'appuyant sur les connaissances et expériences des opérateurs du four afin de mettre au point un schéma complet d'examens de robustesse du processus et d'optimiser les paramètres de contrôle.
- un modèle basé sur des réseaux de neurones et la logique floue utilisant les informations du modèle subjectif et les données provenant de l'installation afin d'extraire et d'apprendre des paramètres supplémentaires mais importants du processus qui ne sont que difficilement ou pas du tout contrôlables.

Les motifs principaux pour aborder les fours à arc électrique par une approche de modélisation étaient de mieux prévoir l'instant d'arrêt d'une charge en tenant compte du taux de charbon dans l'acier et la température du bain, de mieux contrôler le processus dans sa globalité en se basant sur les connaissances d'experts et d'apprendre à produire de façon reproductible un acier de qualité à coût minimal.

Au début du projet un four approprié a été sélectionné. Ce four est situé sur le site de ProfilARBED Esch-Belval (pour plus de détails voir chap. 1.2). Les différents modèles et approches sont validés sur ce four.

Afin de coordonner le projet et de favoriser la communication, des réunions semestrielles étaient organisées en alternance chez les quatre partenaires. Un rapport technique sur les avancements semestriels de chaque partie accompagnait ces réunions. Un rapport technique et financier par année résumait les

[2]Communauté Européenne du Charbon et de l'Acier.
[3]Laboratoire Lorrain de Recherche en Informatique et ses Applications, http ://www.loria.fr
[4]Centre de Recherche Public Henri Tudor, http ://www.tudor.lu
[5]Département de recherche du groupe métallurgique luxembourgeois ProfilARBED, http ://www.arcelor.lu
[6]Groupe métallurgique espagnol.
[7]Centre de Recherches Métallurgiques de Belgique.

activités et une présentation annuelle des travaux du projet devant le conseil "C1"[8] était imposée par la CECA.

1.1.1 Le rôle des partenaires

ProfilARBED Recherche était responsable de la coordination du projet. Ils ont mis au point le système d'acquisition de données sur le site et font la maintenance et la mise à jour de ce système afin de fournir les données nécessaires aux autres partenaires. Ils ont travaillé sur différentes méthodes d'analyse des données énergétiques du four, dont un modèle de prédiction de la fin de charge basé sur un couplage séquentiel de réseaux de neurones.

Le CRM possède un modèle classique de la métallurgie capable de modéliser les phénomènes physique et chimique à l'intérieur du four. Le travail du CRM consistait à adapter ce modèle au four cible. Différentes campagnes de mesures spécifiques étaient réalisées afin de pouvoir adapter les paramètres de ce modèle.

L'apport du CRP Henri Tudor dans le projet est entre autres l'analyse et l'adaptation de méthodes basées sur les réseaux de neurones artificiels afin d'optimiser le contrôle de l'installation en tenant compte des connaissances des experts.

1.1.2 La relation projet-thèse

Dans le cadre du projet CECA un travail de thèse en informatique à l'université Henri Poincaré a été défini sur la base d'une collaboration entre le CRP Henri Tudor et le LORIA. Cette thèse a été financée en majeure partie par une bourse de formation recherche de l'état luxembourgeois.

La partie de l'apport du CRP Henri Tudor en relation avec le domaine des réseaux de neurones est donc fortement liée à ce travail de thèse ce qui fait que les approches et méthodes analysées et proposées dans cette thèse sont directement voire indirectement liées au domaine du connexionnisme.

1.2 Le processus industriel

1.2.1 L'installation industrielle

L'installation industrielle est située sur le site de ProfilARBED Esch-Belval qui se trouve au sud du Luxembourg à la frontière française. Sur ce site la production d'acier a une longue tradition. Déjà au début du 20ème siècle une production d'acier était implantée sur ce site. On trouvait la chaîne classique de production d'acier à partir de minerai de fer c.-à-d. une agglomération, des hauts fourneaux, une aciérie, le stockage en coquilles, le réchauffement, le laminage et stockage du produit fini.

Il y a une vingtaine d'années une nouvelle technologie est apparue sur le marché de la production d'acier. Elle remplaçait toute la partie de la production avant le laminage. Cette nouvelle technologie de production d'acier réalise une approche de recyclage des mitrailles[9]. Elle consiste à fondre la mitraille dans un four en se basant principalement sur l'énergie électrique. Les avantages de cette approche sont multiples : plus aucune dépendance du minerai de fer, abolition de l'ancienne technologie des haut-fourneaux etc.

Les responsables de la société ARBED au Luxembourg ont alors décidé une migration radicale de tous leurs sites de production d'acier sur cette nouvelle technologie.

En 1997 le four à arc électrique a été mis en service sur le site de ProfilARBED Esch-Belval, le site cible de ce projet.

1.2.1.1 Informations sur l'installation

Généralement de nombreuses informations existent sur une installation industrielle. Les types d'informations sont en principe très divers et peuvent comprendre entre autres des plans de construction, schémas techniques de parties de l'installation, des consignes d'utilisation, des consignes de sécurité, des

[8] Réunion annuelle de la CECA où le coordinateur de chaque projet présente ses avancements

[9] Un ensemble de fragments métalliques divisés, provenant généralement de diverses récupérations

fiches de suivi des charges et des données provenant de capteurs autour de l'installation. Le plus souvent ces informations ne sont pas gérées de façon centralisée et des liens entre elles n'existent que pour un sous-ensemble de ces informations. Cette situation n'est pas spécifique à ce projet, elle est connue sous le terme de données hétérogènes (angl. Disparate Data) et analysée dans [7].

Une autre particularité de ces informations est la forme sous laquelle elles sont accessibles. Un mélange de fiches en papier éventuellement cataloguées dans des classeurs, de fichiers informatiques de formats divers et des bases de données sont la règle.

A cela viennent s'ajouter de nombreuses connaissances des différents acteurs humains qui n'existent pas sur un support facilement accessible, mais seulement dans les réseaux de neurones appelés cerveaux de ces acteurs. Ces informations sont basées sur des observations faites, ou elles sont reliées à des modifications faites qui ne sont pas nécessaiement documentées.

Dans le cadre de ce projet, aussi bien les données mesurées que les expériences du personnel ont une importance. Ce deuxième type d'informations est généralement regroupé sous le terme de connaissances d'experts.

Une bonne relation et une collaboration étroite avec le personnel de l'installation sont primordiales, au moins au début d'un tel projet, afin de rassembler toute information nécessaire pour acquérir les connaissances nécessaires sur le processus.

En ce qui concerne l'installation, le tableau 1.1 résume les données principales du four et donne un premier aperçu de la taille de l'installation.

Four à courant continu	
Capacité du four (t)	155
Diamètre de la cuve (m)	7.3
Diamètre de l'électrode (mm)	762
Nombre d'électrodes de sole	4
Tension maximale de l'arc (V)	800
Courant maximal (kA)	120
Nombre de brûleurs	5
Puissance totale des brûleurs (MW)	50
Nombre d'injecteurs de postcombustion	4
Mise en service	1997

TAB. 1.1 – Caractéristiques principales du four

Afin de donner une idée de l'installation, les éléments principaux sont visualisés dans le schéma (fig. 1.1) reprenant la structure générale d'une aciérie du type four à arc électrique à courant continu comme c'est le cas à ProfilARBED Esch-Belval.

Au centre de la figure se trouve le four avec à côté le bras mobile qui porte et positionne l'électrode en graphite afin d'apporter l'énergie électrique dans l'enceinte du four. Au-dessus du four à gauche le panier à mitrailles est prêt pour charger le four. Le chargement se fait en déversant la mitraille dans le four. En-dessous du four à gauche se trouve la poche qui reprend l'acier liquide en fin de charge au moment de la vidange et à droite la poche qui reprend l'excès de scorie qui coule du four. A droite du four une personne est représentée afin de donner une impression des dimensions de l'installation. Les tuyaux à gauche du four représentent une partie du système de dépoussiérage et du traitement des gaz d'échappements (voir Fig.1.5).

Plus de détails sur la structure du four sont donnés dans la figure 1.2.

Le four est composé de trois parties :

- La cuve : elle représente la partie inférieure du four et héberge les électrodes de sole qui ferment le circuit électrique. Elle est revêtue d'un réfractaire afin de résister aux températures de l'acier liquide qu'elle contient. Au bord de la cuve se trouve le trou de coulée qui sert à déverser l'acier liquide en fin de charge.
- Les panneaux : ils forment le collier latéral central c.-à-d. les parois du four. Ils reposent sur la cuve. Ils sont refroidis à l'eau et ont aussi un rôle de support pour diverses armatures comme des brûleurs à gaz ou des lances à oxygène etc.

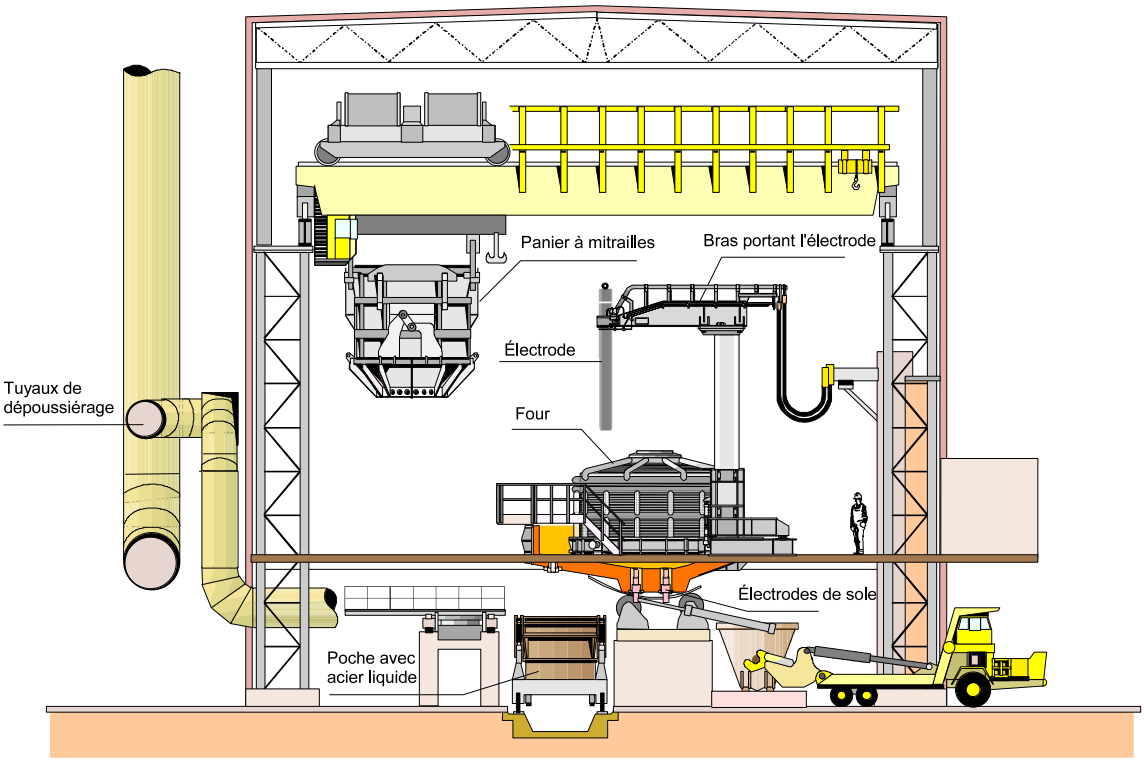


FIG. 1.1 – Structure générale d'une aciérie à arc électrique

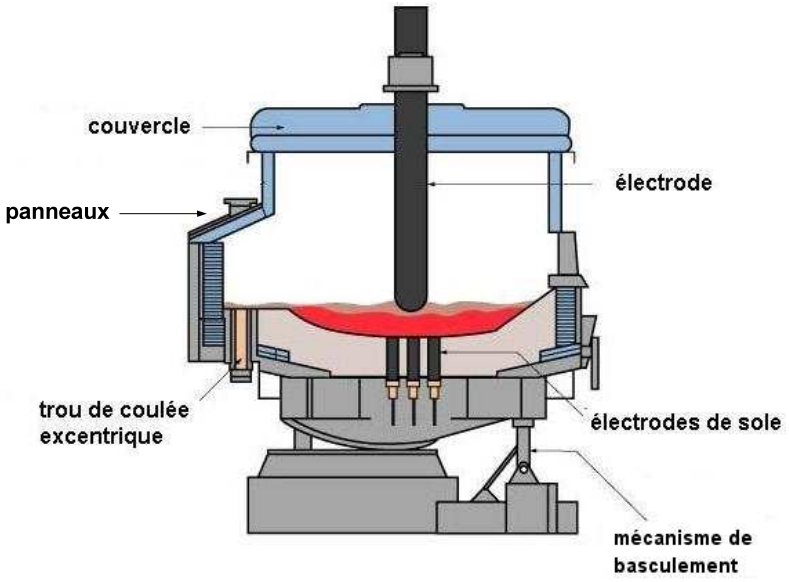


FIG. 1.2 – Vue détaillée du four

– Le couvercle : il referme le four d'en haut et est également refroidi à l'eau. Par un trou central dans le couvercle l'électrode est introduite dans le four. Afin de charger le four, le couvercle peut être soulevé et pivoté sur un côté (voir Fig.1.3 et 1.4).

Le four entier avec le bras de l'électrode et la chambre de postcombustion repose sur une plate-forme mobile qui peut basculer autour d'un axe horizontal perpendiculaire au plan de la figure 1.2 afin de pouvoir déverser l'acier liquide par le trou de coulée en fin de charge. Durant la charge, le trou est refermé par un bouchon.

La figure 1.3 montre une photo de l'intérieur de l'aciérie de Belval. On voit le four avec le couvercle ouvert, des flammes et des fumées s'échappent de la cuve et sont aspirées par une hotte sous le toit de l'aciérie (voir Fig.1.5). La figure 1.4 montre un schéma de l'aciérie avec un angle de vue similaire, afin de mieux localiser les différentes parties de l'installation.

FIG. 1.3 – L'intérieur de l'aciérie de Belval

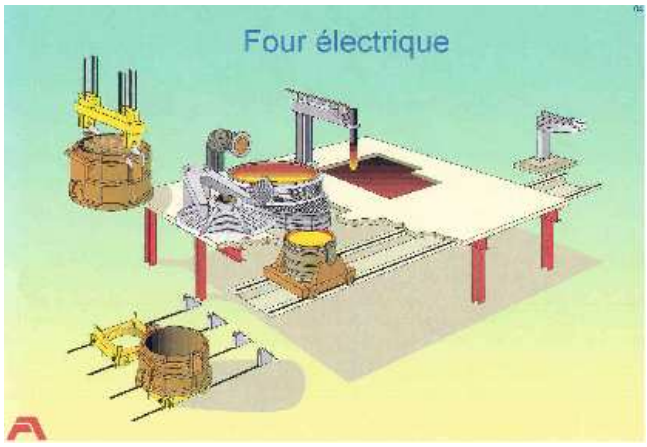


FIG. 1.4 – Schéma de l'installation

Les schémas techniques de l'installation constituent un genre d'information indispensable pour la compréhension et l'interprétation des données. Dans le schéma (Fig. 1.5) une vue globale sur les circuits de refroidissement et le système de dépoussiérage des fumées est représentée. Plus de détails sur différents circuits comme par exemple les positions et la spécificité des capteurs de mesures sont repris par d'autres schémas.

La figure 1.5 montre bien la complexité de l'installation et le nombre de paramètres qui ont une influence plus ou moins importante sur le système "four à arc électrique" qu'il s'agit d'analyser.

Connaître les détails techniques de l'installation, le fonctionnement des différents éléments et les relations qui existent entre eux est certainement nécessaire pour mener à bien un tel projet de recherche mais dépasserait le but de cette thèse. Il est plus important pour le lecteur, afin de cerner le problème posé, de connaître encore quelques spécificités de l'installation.

1.2.1.2 Quelques spécificités de l'installation

Le processus industriel en question se trouve dans un environnement sidérurgique, c.-à-d. dans un cadre d'industrie lourde. Il faut savoir que dans un tel environnement tout le système de mesure comprenant les capteurs de mesure, les câbles et les automates programmables, est mis à une rude épreuve. Des capteurs défectueux, non calibrés ou encore de type divergeant de la spécification sont fréquents. On est donc confronté à des mesures aberrantes, bruitées, incomplètes sans pour autant en être nécessairement conscient.

Une autre particularité concerne la synchronisation d'informations. Cette tâche n'est pas triviale car les systèmes de mesures sont autonomes et indépendants. La référence de temps existant sur les différents

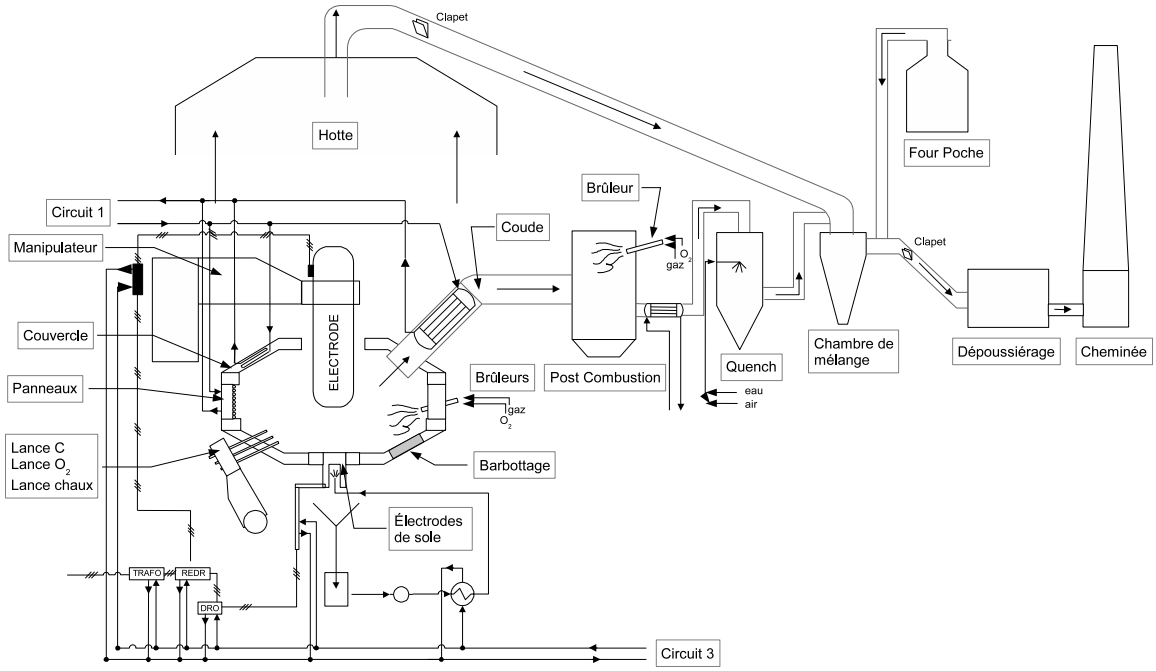


FIG. 1.5 – Schéma du système de refroidissement et du traitement des fumées

systèmes n'est pas synchronisée et les valeurs peuvent avoir d'importantes divergences. Ce qui rend la tâche encore plus difficile est que l'acquisition des informations s'effectue sur diverses échelles de temps.

De plus, le processus est incorporé dans une chaîne de production d'acier. Il y a donc des processus en amont et en aval qui ont une influence de type contrainte externe sur le processus en question. Il n'est pas rare qu'une panne quelque part dans la chaîne de production mène à un arrêt plus ou moins long du processus.

Un aspect à ne pas oublier est que le procédé ainsi que l'installation industrielle ne sont pas des éléments stables dans le temps. Ils subissent au fil du temps de petites, moyennes ou même importantes modifications, ce qui a des répercutions sur le comportement du processus.

Il a été constaté que divers éléments de mesure de l'installation ne fonctionnent pas bien voire même pas du tout. Suite à des observations faites sur le site, il est intéressant de faire remarquer qu'il existe une relation directe entre le mal- ou non-fonctionnement d'éléments de l'installation et la faible importance que les acteurs humains leur attribuent. Ainsi, des capteurs jugés peu pertinents ne sont pas maintenus de la même manière que les capteurs sur lesquels se réfèrent les différents acteurs.

Dans ce contexte il faut rappeler que le but primaire de cette installation est la production de la matière première (l'acier liquide) pour les processus en aval de la chaîne de production complète. Les réparations, les échanges de matériel défectueux et les modifications sont effectués selon leurs importances en relation avec ce but primaire. Les interventions urgentes, qui sont indispensables pour le fonctionnement du four, sont effectuées immédiatement afin de pouvoir continuer le processus. D'autres interventions en relation avec un fonctionnement restreint de l'installation sont planifiées et réalisées entre deux cycles de production (appelés charges). Les travaux de type maintenance sont reportés sur les arrêts de maintenances hebdomadaires voire même bi-annuels pour les travaux d'ampleurs.

Ce fait semble évident, mais il est souvent oublié dans le cadre de réflexions et de décisions qui vont être prises durant un projet de recherche. Pour donner un exemple, supposons que dans le cadre d'une approche de modélisation pour des raisons de connaissances a priori une variable "x" soit utilisée et que sa valeur provienne d'un capteur qui est mal maintenu et donc peu fiable. L'utilisation d'une telle variable peut faire que des méthodes, modèles ou approches ne sont pas implémentés, ou que après une implémentation, ne sont pas utilisés par les opérateurs sur un site de production, puisque la méthode se révèle être peu fiable.

Il est illusoire d'espérer que le fait qu'un modèle se base sur un tel capteur, ce dernier deviendrait pertinent aux yeux des acteurs humains, déjà parce que souvent la relation modèle-capteur n'est que peu visible pour les acteurs sur site. L'inverse est plutôt la réalité : puisque le modèle ne fonctionne pas, il est jugé non utilisable et il se retrouvera dans la liste des éléments de l'installation avec faible importance qui ne sont pas maintenus.

La conclusion de cette observation est qu'il faut éviter de baser des approches sur des mesures que les acteurs humains jugent comme peu pertinentes. Il faut prendre en considération toutes ces propriétés dans les analyses et les propositions de méthodes. La robustesse d'une méthode par rapport à des données bruitées, éronnées ou encore manquantes apporte dans ce cas un avantage non négligeable.

1.2.2 Les acteurs humains

Différents acteurs humains en relation avec l'installation peuvent être identifiés. Du côté de la société productrice, trois groupes d'acteurs peuvent être distingués : il s'agit "des opérateurs" avec leurs groupes de travail, "des responsables de l'installation sur site" et de "l'équipe de recherche et de développement" de l'entreprise. Dans le cadre du projet de recherche il est possible d'ajouter un quatrième groupe, qui est externe à la société et qui se constitue des chercheurs en charge du projet. Analysons le rôle de ces quatre groupes d'acteurs humains.

Le rôle de l'opérateur avec son groupe de travail est de conduire le four afin de produire l'acier liquide pour la coulée continue qui est le maillon suivant de la chaîne de production. Ils ont la possibilité d'agir sur le four par différents moyens. L'opérateur, grâce à son expérience, possède un modèle cognitif du système "four à arc électrique" et du processus de fusion en particulier. Il sait plus ou moins bien prédire le comportement du four basé sur l'historique du processus et les paramètres d'entrée courants.

Le rôle des responsables de l'installation sur site est de planifier en même temps la production, la maintenance et les modifications ou améliorations de l'installation. Ils connaissent le marché et peuvent fixer les critères qu'il s'agit d'optimiser afin de produire à coût minimal. De manière générale le marché est en évolution constante et avec lui en principe aussi les critères qu'il s'agit d'optimiser.

L'équipe de recherche et de développement en relation avec les responsables sur site essaie de trouver des moyens et méthodes afin d'optimiser le processus et d'acquérir également une meilleure connaissance des phénomènes qui contrôlent le processus ainsi que l'influence des différents paramètres sur son comportement.

Finalement le groupe des chercheurs peut dans ce contexte être considéré comme un observateur externe, non-expert du processus, même si sur la durée du projet, quelques connaissances ont pu être regroupées. Ce groupe interagit avec les trois autres acteurs. De nouvelles méthodes et approches sont proposées en collaboration avec l'équipe de recherche et de développement. Le rôle de ce groupe est de favoriser le transfert de connaissances et d'apporter de nouveaux éléments venant du monde scientifique, qui ne sont pas nécessairement bien connus dans le monde de l'industrie.

1.2.3 Le déroulement standard du processus

Le processus étudié peut être caractérisé par la description suivante, montrant le déroulement temporel standard[10] d'une charge. Cette description donne un aperçu du fonctionnement du système "four à arc électrique" :

1. Lorsque le four se trouve dans un régime "normal" (pas d'arrêt prolongé préalable) au départ d'une charge, il reste un pied de bain d'acier liquide de la charge précédente dans la cuve. L'électrode est sortie du four et le couvercle est fermé.
2. Le premier panier de mitrailles est enfourné. Pour cette opération, le couvercle du four est ouvert, une grue achemine un panier avec environ 100 tonnes de mitrailles au dessus du four et le déverse dans le four.
3. Le couvercle est refermé et des brûleurs à gaz sont activés. L'électrode en graphite est introduite dans le four à travers l'ouverture dans le couvercle et le courant électrique est enclenché. Environ

[10] Le terme standard dans ce contexte caractérise le déroulement comme les responsables de l'installation planifient la production d'une charge. La plupart des charges se déroulent globalement selon ce schéma.

20 minutes sont nécessaires à l'arc électrique afin de fondre les mitrailles du premier panier à tel point que le deuxième panier peut être enfourné.

4. Le courant électrique est coupé, l'électrode est ressortie du four et les brûleurs à gaz éteints afin d'ouvrir le couvercle. Un deuxième panier de mitrailles d'environ 70 tonnes est enfourné.
5. Le procédé est relancé comme pour le premier panier. Durant cette deuxième phase différents ajouts (calcaire, carbone) peuvent être injectés dans le four afin d'influencer la composition de l'acier et le comportement du processus.
6. Durant la phase finale du processus, des lances à oxygène qui se situent juste au-dessus de l'acier en fusion sont activées. Le jet au-dessus du bain provoque une oxydation des impuretés, améliore la formation d'un laitier moussant qui recouvre le bain d'acier et garantit une postcombustion des gaz dans l'enceinte du four.
7. Durant cette phase l'excès de laitier qui se forme est déversé dans une poche à scories qui se trouve à droite en-dessous du four.
8. Finalement l'acier liquide doit atteindre une température spécifique qui dépend des procédés en aval de la chaîne de production. Elle est d'environ 1600°C. Avant de vidanger l'acier liquide, sa température est vérifiée et validée à travers une mesure isolée. En fonction de la température mesurée, la vidange peut débuter, le processus doit être relancé ou une phase de refroidissement est entamée.
9. Afin de vidanger l'acier liquide, le trou de coulée est ouvert, le four entier est basculé sur le côté et l'acier est déversé dans une poche à acier posée sous le four. Afin de garantir la qualité de l'acier, il est important d'éviter que des parties de laitier soient déversées dans la poche.
10. A la fin de chaque coulée, un reste d'acier liquide et de laitier, appelé pied de bain, est gardé dans le four.

Une telle charge standard dure en moyenne entre 50 et 70 minutes. En ce qui concerne le déroulement standard il est encore important à noter qu'il arrive de temps en temps que le robot utilisé pour faire la mesure de température, en fin de charge, tombe en panne. Dans ce cas, la mesure est effectuée à la main. Pour des raisons de sécurité la puissance électrique doit être arrêtée durant la mesure manuelle ce qui à une forte influence sur le déroulement.

Des observations ainsi que des discussions avec les experts de l'installation indiquent que le comportement du four dépend fortement de l'évolution temporelle de l'installation. Une dépendance de l'historique à court, moyen et même à long terme a été constatée.

- Le court terme représente l'échelle de temps intra charge, c.-à-d. tout ce qui se passe dans le cadre d'une charge. Par exemple l'instant auquel les différents ajouts sont introduits dans le four, la masse de ces ajouts ou encore le moment où les lances à oxygène sont enclenchées et avec quel débit elles travaillent. Mais aussi la durée entre les deux phases de fusion ou celle du chargement de la mitraille.
- Le moyen terme représente l'échelle de temps d'une, voire de quelques charges. Il a été observé que le comportement d'une charge dépend des charges précédentes à plusieurs niveaux : par exemple la hauteur du pied de bain[11] ou encore la température de l'acier liquide avant la vidange influent directement, car l'énergie que représente cet acier liquide est portée d'une charge à l'autre.
- Le long terme représente l'échelle de temps de quelques jours, de semaines, voire l'influence des saisons (température de l'air et de la mitraille ou encore l'humidité) sur le processus.

L'état général de l'installation, comme par exemple l'usure du réfractaire, l'état de réchauffement du four ou encore la qualité de la mitraille influent sur le comportement de l'installation.

1.3 Les données du processus

Dans ce chapitre, seules les données reliées à la description de l'évolution temporelle du processus sont considérées. Les autres types d'information de l'installation dont il était question dans le chapitre 1.2.1 ne sont pas repris ici, ni dans la suite des analyses. Il ne faut pas pour autant oublier que ce type d'informations est utilisé implicitement dans les choix et réflexions tout au long du projet.

[11] Volume d'acier liquide restant dans la cuve après vidange

Les données relatives à l'évolution temporelle sont regroupées dans les catégories suivantes :

- les données dites cycliques. Elles proviennent de divers capteurs autour de l'installation qui fournissent en permanence des valeurs mesurées pour diverses températures, puissances, débits pour n'en citer que quelques-unes.
- les données dites statiques. Ce sont des données reliées à une charge, comme le poids de la mitraille enfournée, la masse d'acier liquide produite par la charge, ou encore l'identité du personnel de la tournée de travail, $\cdots$.
- les données dites manuelles. Ce sont des données provenant des opérateurs au poste de commande du four. Ils notent sur une fiche le degré d'activité du four à un rythme de 15 min. afin d'obtenir une vue globale (une journée sur une fiche A4) sur le comportement du four.
  Sur cette fiche, on retrouve outre le degré d'activité du four, un découpage en charges, le numéro des charges, les raisons de problèmes éventuels et quelques données de fin de charge, comme la température de l'acier au moment de la vidange.

Afin de pouvoir faire une analyse du processus, un certain nombre d'informations sur le processus doivent être disponibles. Un système d'acquisition regroupe les données et les met à disposition. Ceci est nécessaire, car une analyse de systèmes dynamiques doit disposer de données sur une certaine période de temps. Ici la diversité de comportement du processus qui est décrit dans l'historique représenté par les données est importante.

Non seulement l'acquisition des données est importante, mais aussi la gestion des données au sens large du terme doit être garantie. Ces deux sujets sont abordés ci-dessous.

1.3.1 Le système d'acquisition de données

Avant le lancement du projet, un certain nombre de données du processus ont déjà été enregistrées dans un système de gestion de base de données (SGBD) SyBase qui tourne sur un système VAX géré par la section informatique de ProfilARBED. On y trouve une partie de données cycliques et un certain nombre de données statiques.

En vue d'une analyse du processus plus détaillée, un nombre plus important de variables est nécessaire. En raison de cette nécessité et afin d'augmenter la flexibilité, un deuxième système d'acquisition a été mis en place par ProfilARBED Recherche en collaboration avec le CRP Henri Tudor (voir figure 1.6).

Le nouveau système d'acquisition réalisé sur une plate-forme de type PC industriel n'enregistre que les données de type cyclique. Deux types de données cycliques sont distingués :

- les données brutes : il est question d'environ 220 variables provenant directement de capteurs en passant par des automates programmables. A part la digitalisation, aucune manipulation secondaire n'est faite sur ces données.
- les données calculées : elles sont calculées à partir des données brutes. Il s'agit d'environ 90 variables supplémentaires qui ne sont pas directement mesurables comme par exemple des volumes ou des énergies cumulées. Les calculs sont réalisés sur l'ordinateur du système d'acquisition sur la base d'analyses de référence du domaine métallurgique faites par Koehle [34].

Le temps d'échantillonnage pour ces données a été fixé à cinq secondes par les experts de ProfilARBED Recherche estimant qu'un échantillonnage plus précis n'est pas nécessaire. Ainsi, toutes les cinq secondes les valeurs de 310 variables sont enregistrées par le système d'acquisition.

Les données de type statique ne sont pas encore traitées par ce nouveau système. Elles sont gérées dans un SGBD de type SyBase. L'accès à cette base de données passe par des requêtes SQL.

En ce qui concerne les données de type manuel, il n'y a pas longtemps, elles n'étaient pas encore traitées de manière électronique. Les fiches remplies par les opérateurs étaient envoyées par FAX à ProfilARBED Recherche. Depuis peu, ces fiches sont encodées par les opérateurs dans des fichiers MS-Excel, et elles peuvent être distribuées automatiquement par courrier électronique.

L'accès à l'ordinateur du système d'acquisition et au système VAX est possible depuis le réseau informatique de ProfilARBED. Pour des raisons de sécurité un accès à ce réseau de l'extérieur n'est pas autorisé.

Plus de détails techniques sur le système d'acquisition sont repris dans l'annexe A.1, des détails sur l'accès aux données et leur gestion sont donnés dans la section 1.3.2 ci-dessous.

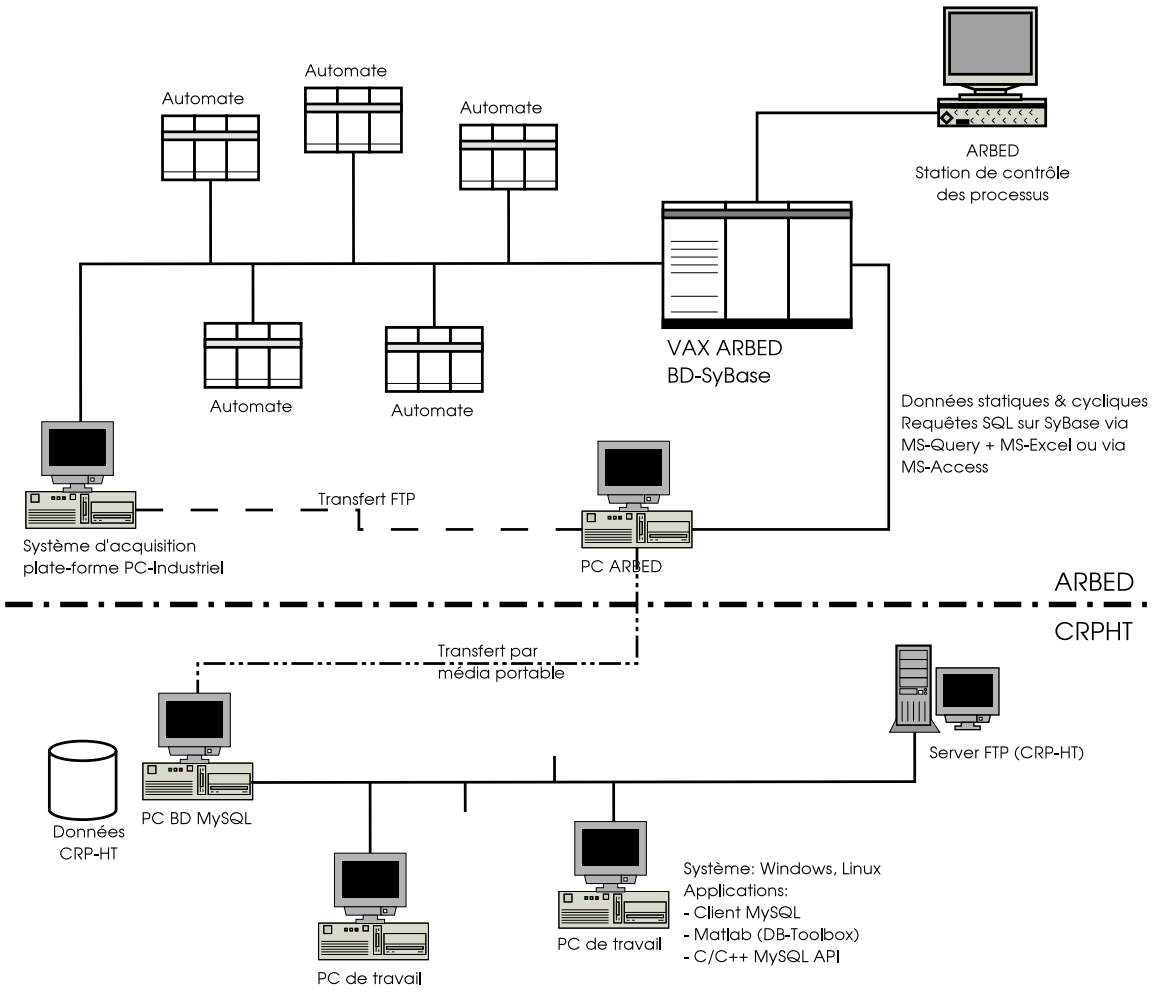


FIG. 1.6 – Flux des données provenant de l'installation

1.3.2 La gestion des données

Toutes les approches de modélisation nécessitent des données réelles pour adapter leurs paramètres au processus analysé. Souvent les données pour un tel projet sont fournies par un partenaire industriel ou institutionnel et c'est ce dernier qui s'occupe généralement de l'acquisition des données. Indépendemment de la manière dont la gestion des données est faite par ce partenaire il est important pour un projet de recherche qui est basé sur des données d'organiser et de gérer les données, informations et résultats qui concernent le projet de façon adéquate. Une référence intéressante dans ce contexte est le livre de M. Brackett [7] qui traite le sujet de comment organiser et gérer les données hétérogènes d'une entreprise afin de profiter des informations relationnelles de ces diverses sources.

En ce qui concerne la gestion des données en relation avec un projet de recherche les objectifs suivants peuvent être distingués :

- un stockage centralisé,
- une accessibilité flexible et standardisée,
- une garantie de cohérence,
- une garantie de reproductibilité d'expériences,
- une sauvegarde (backup) simplifiée,
- une garantie de confidentialité envers les partenaires,
- une facilité de mise en relation de différents types de données.

Deux niveaux de gestion de données peuvent être distingués : La gestion primaire qui est réalisée par le partenaire indépendamment du projet, et une gestion secondaire par le projet de recherche. Dans le cas présent, la gestion primaire est réalisée sur le système d'acquisition par ProfilARBED Recherche et la gestion secondaire au CRP Henri Tudor.

1.3.2.1 La gestion primaire de données

Sur le système d'acquisition, différentes méthodes de stockage ont été réalisées durant ce projet. En premier lieu les données cycliques sont stockées dans des fichiers ASCII dans un format simple et standardisé (CSV). Différents types de fichiers sont enregistrés comprenant des regroupements divers de données. Une nomenclature spécifique garantit l'unicité des fichiers et un accès ciblé. Les types de fichiers qui nous intéressent le plus dans le cadre de ce projet sont les fichiers bruts où toutes les variables sont sauvegardées de façon continue.

D'autres fichiers ne gardent qu'un sous-ensemble des données ou des informations de bilans par charge dans l'idée de faciliter leur utilisation.

L'accès à ces fichiers est possible en utilisant le protocole FTP, mais limité au réseau de ProfilARBED. Chaque utilisateur travaille donc sur une copie locale d'un ou de plusieurs fichiers dont il récupère un sous-ensemble d'informations qui l'intéresse.

Ce type de gestion primaire des données n'est pas très pratique pour l'utilisateur, car il est obligé de s'organiser une propre gestion des données à partir d'un important volume d'informations redondantes, ce qui consomme inutilement l'espace de stockage, est susceptible de nuire à la transparence de la gestion et ne garantit que difficilement la cohérence des données.

En ce qui concerne la sauvegarde des données, les fichiers ASCII comprimés, contenant les données brutes, sont gravés sur CD et archivés chez ProfilARBED Recherche.

ProfilARBED Recherche, qui a réalisé la gestion primaire des données et qui fait la maintenance du système, n'essaie pas d'atteindre strictement tous les objectifs mentionnés. Dans l'annexe A.2.1 plus de détails sont donnés concernant la gestion primaire de données réalisées sur site.

1.3.2.2 La gestion secondaire de données

En ce qui concerne la gestion de données au CRP Henri Tudor, un des buts était de mettre en oeuvre un système qui serait susceptible de garantir les objectifs cités auparavant. Cela aussi parce que les projets qui sont traités par l'équipe de modélisation du Centre ont presque toujours une partie de gestion de données qui pourrait être standardisée.

La première partie de la gestion secondaire de données est la récupération des données du partenaire. Cette partie traite le sujet de l'accès, du transfert, du format et de la confidentialité des données. Dans le cas de ce projet, le transfert des données est réalisé par des médias portables (voir Fig. 1.6).

La deuxième partie concerne l'incorporation des données numériques récupérées dans un système de gestion de données qui a le potentiel de garantir les objectifs cités auparavant. Dans ce cas un système de gestion de base de données (SGBD) MySQL est utilisé.

Un type de bases de données regroupe les différentes données provenant du processus. On trouve les données de type cycliques et statiques qui sont utilisées pour les diverses analyses. Le tableau en annexe (Tab.A.1) résume les données rassemblées pendant la période du premier décembre 2000 au sept juillet 2001.

Dans un Centre de recherche plusieurs projets avec différents partenaires sont gérés. Il est important de pouvoir donner une garantie aux partenaires en ce qui concerne la confidentialité des données qui sont mises à disposition. Un système de gestion de base de données (SGBD) est prédestiné pour remplir cette tâche. Ces SGBD offrent un système de gestion du droit d'accès basé sur un nom d'utilisateur et un mot de passe qui gère l'accès aux bases de données voire aux tableaux.

Un autre avantage d'un SGBD est la flexibilité d'accès aux données. Quelques types de méthodes d'accès sont représentés dans la figure 1.7.

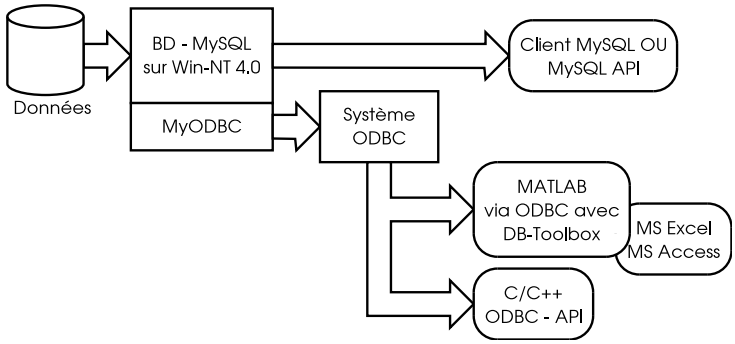


FIG. 1.7 – Méthodes d'accès aux données au CRP Henri Tudor

Principalement deux types de méthodes sont distingués :

- L'accès direct au SGBD, en se basant sur les API offerts par ce dernier. Cette méthode est généralement la plus rapide du point de vue connexion et transfert de données. Elle est en général la plus riche du point de vue fonctionnalité car elle met à disposition toutes les possibilités et propriétés offertes par le SGBD.
- L'accès indirect, en passant par une passerelle offrant un protocole standardisé pour la communication avec un SGBD. Dans la figure 1.7 le protocole standard ODBC est utilisé comme représentant de ce type de communication. Le grand avantage est que la plupart des SGBD sont compatibles avec ces standards, ce qui rend l'application indépendante d'un SGBD particulier. Si pour une raison ou une autre il faut changer le SGBD toutes les applications sont réutilisables sans aucune modification.

  Cependant souvent les standards ne définissent qu'un sous-ensemble commun des propriétés offertes par les SGBD. De plus le fait d'implémenter un protocole standardisé implique automatiquement un certain surplus au niveau du protocole, ce qui engendre une réduction de la performance du côté du SGBD.

Un autre aspect de la gestion des données est la gestion des résultats d'analyse. Dans ce contexte il est important de rappeler qu'afin de pouvoir reproduire les expériences, il est indispensable de sauvegarder tous les paramètres des expériences.

La mise en place du système d'acquisition, de la gestion primaire et secondaire des données provenant de l'installation industrielle a été indispensable afin d'entamer une phase d'analyse basée sur des exemples. Etre capable d'accéder de façon structurée et reproductible à des données et en même temps pouvoir augmenter au fur et à mesure le corpus de données représente une bonne base pour tout type d'analyse

basée sur des exemples. Une telle partie de gestion de données est souvent oubliée ou sous-estimée dans la planification de projets de recherches.

1.4 Aperçu sur le concept de contrôle prédictif basé sur une aide à la décision

Avant de regarder en détail les méthodes utilisées pour l'analyse de données temporelles et avant de détailler les propositions qui sont faites dans cette thèse, concernant une approche de contrôle prédictif basée sur une aide à la décision, les idées globales de cette approche sont introduites dans cette section transitoire. Le but de cette section est de familiariser le lecteur avec ce concept afin de mieux préparer l'argumentation et les décisions qui seront prises dans les chapitres suivants. Elle souligne aussi la structure et le choix des chapitres 3 et 4, car l'approche peut être divisée en deux parties distinctes. Dans cet aperçu les deux parties sont présentées dans l'ordre inverse des chapitres de la thèse parce qu'il a été constaté que cet ordre facilite la compréhension de l'approche globale. L'ordre de la suite du document est structuré différemment, parce que les éléments qui se succèdent se construisent les uns sur les autres.

Commençons par le but global posé par le partenaire industriel, qui consiste en premier lieu à trouver un moyen pour réduire les coûts de production, mais aussi à mieux comprendre le comportement du processus. Dans ce projet l'unité four à arc électrique de la chaîne de production complète est visée. Une constatation déjà évoquée dans le chapitre 1.2.3 est que le comportement du processus dépend fortement de son historique. De plus, l'expérience montre qu'il existe des régularités dans le comportement du système "four à arc électrique". Une partie des paramètres qui ont une influence sur ce comportement sont connus, mais la façon dont ils influent n'est que très mal perçue.

Aujourd'hui l'opérateur a des consignes en forme de profils d'évolutions schématisés globalement pour une charge. Ces profils ont été élaborés par le fournisseur de l'installation et adaptés par les responsables de l'installation sur site selon leurs besoins et expertises. Des observations ont montré que souvent il n'est pas possible pour l'opérateur de suivre ces consignes à cause de divers paramètres qui influent sur le comportement de l'installation. Les opérateurs réagissent alors intuitivement (selon un modèle interne) de manière à contrôler plus ou moins bien la situation.

Il serait avantageux de connaître les types de comportement que peut avoir le processus et de pouvoir informer l'opérateur à l'avance afin qu'il puisse éventuellement éviter des modes de comportement qui sont connus pour générer d'importants coûts de production. L'hypothèse est que par une telle interaction il est possible de changer la distribution de consommation globale de manière à réduire les coûts de production (voir figure 1.8).

Dans la suite de cette section une telle approche de contrôle prédictif basé sur une aide à la décision va être proposée. L'approche proposée se divise en deux parties :

- **la partie du contrôle en ligne.**
  Dans cette partie la situation actuelle du processus avec un certain historique est évaluée afin de proposer une marche à suivre en vue de finir la charge en cours, tout en intégrant la capacité de tenir compte d'un certain nombre de critères qu'il s'agit d'optimiser.
- **la partie analyse évolutive du processus.**
  Cette partie fournit les outils et méthodes élémentaires sur lesquels repose la partie de contrôle en ligne. Un nombre représentatif de données du processus est analysé dans l'objectif d'extraire les comportements-types du processus et d'être capable de pouvoir les mettre en relation avec les critères d'optimisation.
  Il s'agit de faire émerger des prototypes de comportement temporel du processus en se basant sur les données mesurées.

Cette approche de contrôle a été choisie après avoir constaté par des analyses préliminaires, des observations sur site et des discussions avec les différents acteurs, qu'il doit y avoir une relation entre le comportement temporel du processus et les critères qu'il s'agit d'optimiser. La figure 1.9 résume l'approche de contrôle prédictif proposée par cette thèse.

Afin d'expliquer la démarche, posons la situation suivante schématisée dans la figure 1.9 : Un nouveau cycle de production à commencé il y a environ 12 min. La phase de fusion du premier panier est en cours.

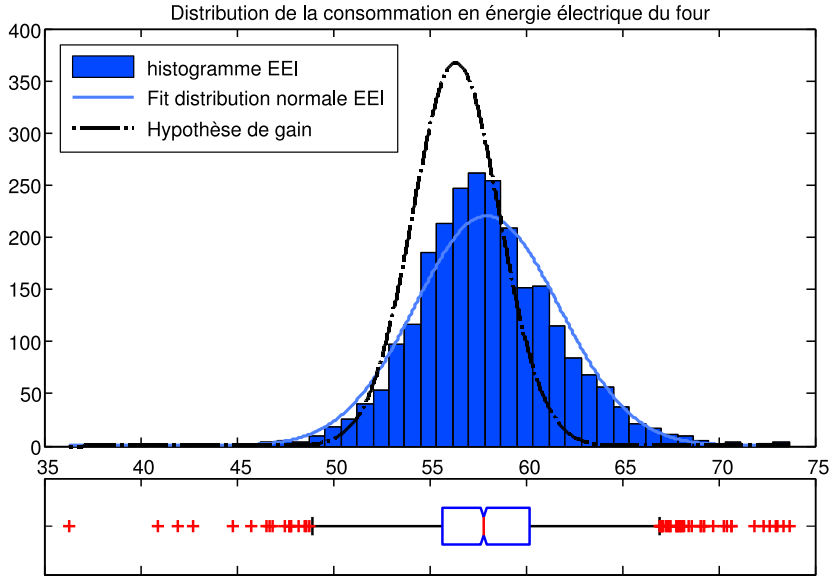


FIG. 1.8 – Principe et hypothèse afin d'engendrer une baisse de consommation.

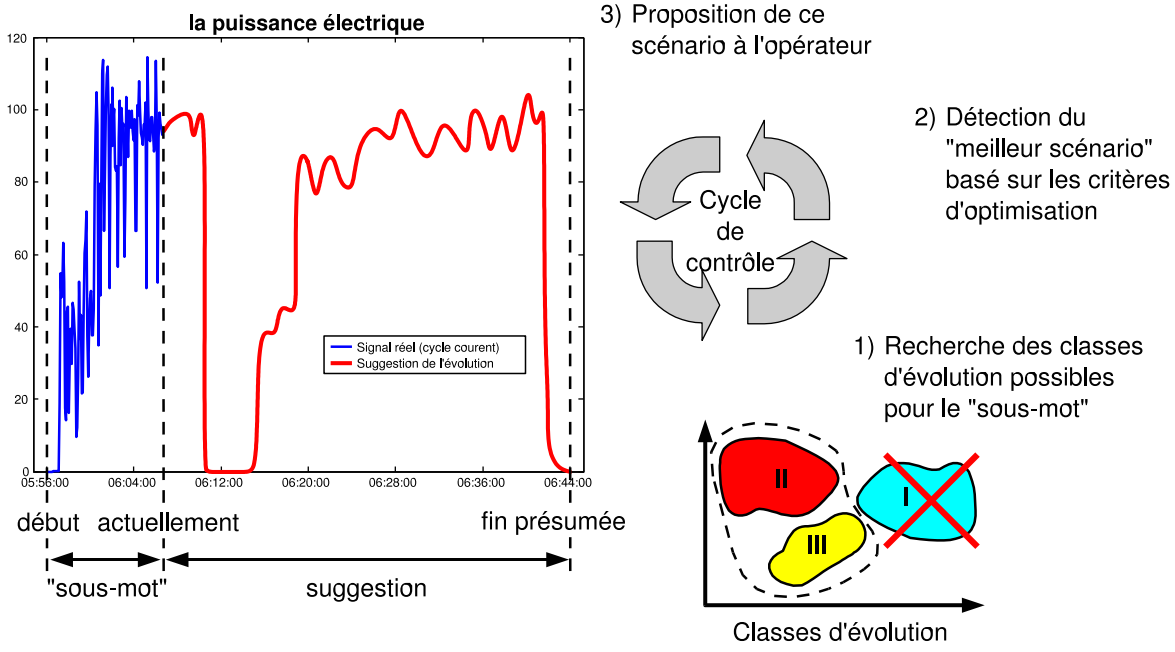


FIG. 1.9 – Concept du contrôle prédictif en ligne

Le graphe de la puissance électrique montre l'évolution de cette première partie du processus du début de la charge jusqu'à l'instant actuel. Cette partie décrit l'historique à court terme c.-à-d. l'évolution connue de cette charge. Elle ne représente qu'une partie de l'évolution complète de la charge qui n'est pas encore connue à cet instant. Après codage du signal qui sera expliqué dans la deuxième partie de l'approche, ce début de charge est représenté par une sous-séquence appelée "sous-mot". Ce terme vient du fait qu'une évolution complète d'une charge représentée dans sa forme codée est appelée un "mot" (voir Fig. 1.10).

Afin de pouvoir faire une proposition pour la marche à suivre en vue de finaliser la charge en cours, tout en gardant la possibilité de tenir compte d'un certain nombre de critères, quelques suppositions sont faites :

- Première supposition : il existe une classification des évolutions temporelles des charges complètes réalisées par l'installation et chaque classe peut être représentée par une évolution type.
- Deuxième supposition : il est possible de trouver un sous-ensemble de classes d'évolutions qui sont proches de la partie connue ("sous-mot") de l'évolution de la charge en cours.
- Troisième supposition : il existe une relation entre les critères à optimiser et les classes d'évolutions.

Ces trois suppositions reflètent qu'une certaine connaissance a pu être extraite des données analysées. Le détail sera analysé dans la deuxième partie de l'approche.

Avec ces trois suppositions, il est possible de proposer à l'opérateur à tout moment de l'évolution d'une charge ("sous-mot") un ou plusieurs scénarios ("suggestions") de la marche à suivre pour conduire le four, afin de terminer la charge en cours en tenant compte des critères qu'il s'agit d'optimiser.

Le concept de contrôle prédictif tel qu'il est proposé ici pourrait aussi appartenir à la classe des approches d'aide à la décision. Ceci parce que le contrôle est classiquement une opération autonome se référant à des mesures du processus et à un modèle du processus réel. De plus, le contrôle agit de façon autonome sur les paramètres d'entrée du processus afin d'influencer son comportement.

Le terme "contrôle" a été adopté pour souligner la boucle fermée de l'approche. Dans le concept proposé, la boucle est fermée par l'opérateur qui premièrement choisi la proposition qu'il va suivre et deuxièmement agit sur les paramètres de l'installation afin de réaliser cette proposition. Les scénarios proposés s'adaptent en permanence au comportement du processus. Si l'opérateur décide de conduire le four autrement ou si, en raison d'un événement inattendu, il est obligé de changer de procédure, cette approche est capable de détecter le changement d'évolution, de le prendre en compte et de faire une proposition adaptée pour la suite du processus.

Au début d'un cycle de production, l'impact de la proposition sur l'optimisation est plus important parce que la majeure partie du cycle est ouverte et peut encore être influencée. Par contre la détection d'une classe d'évolution adéquate est moins évidente quand la partie connue du cycle ("sous-mot") est courte. Cela s'inverse avec l'avancement du processus. Avec l'augmentation de la partie connue du cycle, la détection de classe d'évolution adéquate se précise, cependant la marge d'influence sur la suite de l'évolution de la charge se réduit et avec elle le potentiel d'une éventuelle amélioration.

Ce concept de contrôle prédictif se base sur les trois suppositions faites auparavant. La dernière supposition affirme qu'il existe une relation entre les critères qu'il s'agit d'optimiser et les classes d'évolutions. En d'autres termes il existe une relation entre l'évolution d'un cycle de production et ces critères. Non seulement les experts de l'installation soulignent l'existence de cette relation mais d'autres analyses réalisées dans le cadre du projet de recherche CECA soulignent l'existence d'une telle relation [2]. Faute de posséder au départ du projet les moyens de fournir plus de garantie sur l'existence de la relation, cette supposition est considérée comme étant démontrée pour la suite de ce travail.

Les deux autres suppositions ne sont pas indépendantes. La première supposition doit être garantie avant d'aborder la deuxième qui se base sur l'existence d'une classification. Afin d'aborder le sujet de la classification, une deuxième partie de l'approche est proposée, "la partie analyse évolutive du processus" qui va fournir les moyens pour parvenir aux besoins des deux premières suppositions. La figure 1.10 donne une vue globale de cette deuxième partie.

L'analyse évolutive du processus est divisée en deux étapes :

1. L'étape du codage des signaux.
   Le but de cette étape est d'obtenir une forme codée du signal analysé qui réduit le volume de données à manipuler sans pour autant perdre des informations pertinentes. La méthode de codage propose une extention d'une méthode connue (PAA) et est brièvement présentée ici. Elle est constituée de

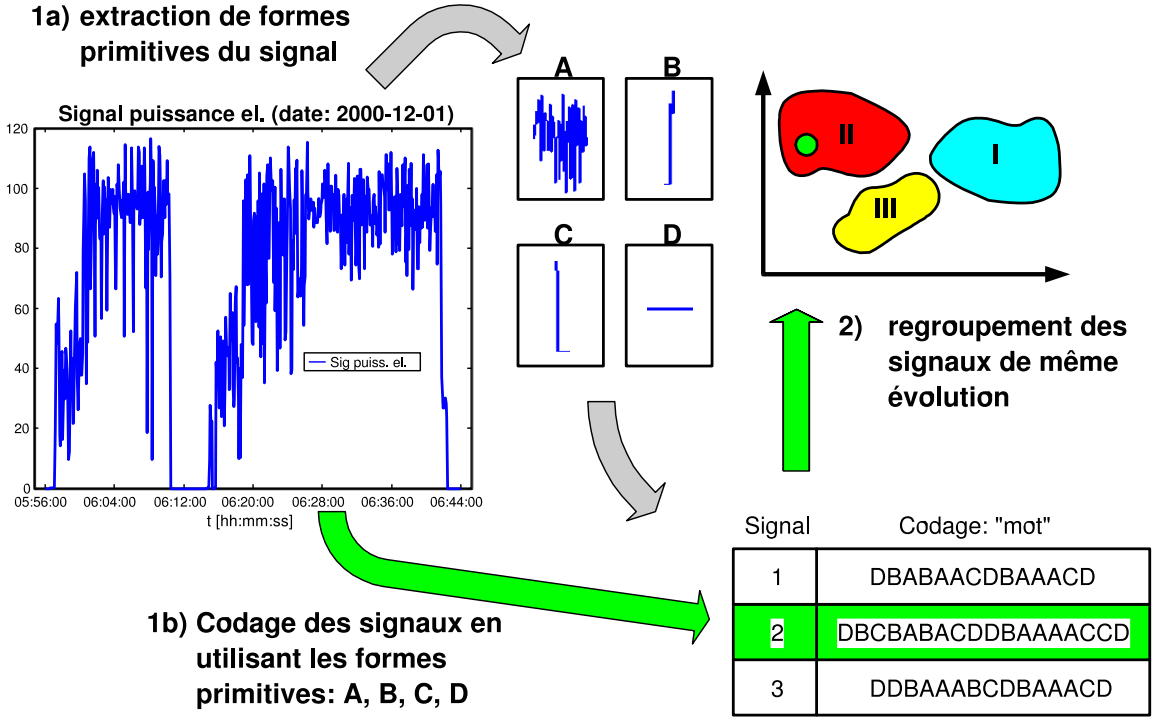


FIG. 1.10 – Le codage et l'analyse du comportement du processus

deux phases :

(a) La phase d'extraction de formes primitives qui composent le signal analysé (1a fig. 1.10).
Le but de cette phase est d'analyser un échantillon représentatif du signal afin d'en tirer les formes les plus caractéristiques. Dans la figure ce sont les formes identifiées par les lettres A, B, C et D. Un ensemble de telles formes primitives est appelé un alphabet.

(b) La phase d'encodage des signaux (1b fig. 1.10).
Après la création d'un alphabet, le signal de tous les cycles de production du corpus analysé est encodé en se basant sur ces formes primitives ce qui produit une suite d'identifiants de formes pour chaque cycle. Une telle suite est appelée un "mot".

2. L'étape de la classification d'évolutions (2 fig 1.10).
Cette étape se base sur une représentation en mots des cycles de production après le codage. Dans cette deuxième étape il s'agit de regrouper les charges qui ont une évolution temporelle semblable dans des classes d'évolution, ce qui revient à regrouper les mots qui sont similaires.

En globalité, le résultat de la partie "analyse évolutive du processus" est un classificateur pour les cycles de production qui se base sur la similarité de leur l'évolution temporelle. Chaque classe est représentée par un prototype qui sera appelé dans la suite : un "prototype d'évolution" ou une "évolution type". Dans cette deuxième partie divers sujets sont abordés avant de pouvoir réaliser un tel classificateur.

Afin de progresser dans la démarche vers la possibilité de proposer à l'opérateur un ou plusieurs scénarios, le chapitre 2 va présenter et discuter la relation entre les approches connexionnistes et les principaux sujets de cette thèse à savoir l'extraction de connaissances, les données temporelles et les approches de contrôle.

Dans le chapitre 3 les différentes étapes respectivement phases évoquées ci-dessus seront détaillées, commençant par une analyse globale des spécificités de données temporelles, et suivi de la méthode utilisée pour le codage avec l'extraction des formes caractéristiques du signal analysé et l'encodage des signaux. Le codage se base sur une méthode de classification par des cartes auto-organisatrices. Le chapitre finit avec une section sur la classification de séries temporelles avec en particulier le sujet de la similarité et les

séquences de taille variable. Une méthode appelée "Dynamic Time Warping" (DTW) issue du domaine de la programmation dynamique sera utilisée en tant que mesure de distances pour alimenter une approche de classification non-supervisée basée sur des prototypes. Les classes d'évolutions qui regroupent des cycles de productions (charges) sont produites et fournissent la base pour le chapitre suivant.

De plus ce chapitre fournit les méthodes et informations de base pour la résolution de la deuxième supposition, à savoir la possibilité de trouver un sous-ensemble de classes d'évolutions proches en ne connaissant qu'une première partie de l'évolution d'une charge.

C'est dans le chapitre 4 que ce sujet de la classification en ligne sera abordé afin de pouvoir en déduire des propositions d'évolutions pour l'opérateur. La méthode proposée réalise une localisation d'une sous-séquence temporelle en se basant sur la mesure de similarité utilisée auparavant.

Les sujets relatifs à l'utilisation des critères d'optimisation et au contrôle en boucle fermée ne sont malheureusement pas concrètement discutés par défaut d'implémentation sur le site. Cependant les sujets sont effleurés à la fin de ce chapitre.

Chapitre 2

Réflexions sur le connexionnisme

Sommaire

Ce travail propose des approches connexionnistes, afin de fournir des outils qui peuvent aider dans l'analyse de processus complexes et ainsi raffiner la compréhension des phénomènes sous-jacents qui influent sur le comportement du processus analysé.

Beaucoup de processus industriels sont partiellement automatisés. En effet, souvent des parties bien définies d'un processus global, celles qui sont plus ou moins bien maîtrisées et autonomes, sont contrôlées en automatique. Cette automatisation partielle se base sur diverses mesures réalisées autour de l'installation. Afin de gérer le processus global, le plus souvent, un groupe d'opérateurs humains surveille l'ensemble depuis un poste de contrôle. En fonction du degré d'automatisation, les opérateurs interagissent de manière plus ou moins directe avec le processus afin de garantir le bon fonctionnement de l'installation.

L'opérateur est confronté à un nombre d'informations important qui ont souvent différents niveaux d'abstraction, comme par exemple des mesures provenant directement de capteurs de l'installation, ou des informations plus abstraites ou plus globales issues de parties du processus automatisé, voire d'autres informations globales provenant d'autres unités de la chaîne de production. Il doit surveiller ces informations et en tirer des conclusions en vue de réagir de façon appropriée.

C'est dans ce contexte que des approches avancées du domaine de l'analyse de données et de la modélisation peuvent intervenir, afin d'aider l'opérateur dans ses choix. Les approches connexionnistes sont connues pour leur faculté d'analyse non-supervisée, et l'extraction de connaissances qui en découle, leur propriétés d'analyse non-linéaire ainsi que la faculté de traiter des informations complexes et de pouvoir les représenter dans différents niveaux d'abstraction, sans oublier leur robustesse par rapport aux données bruitées.

Ce chapitre propose quelques bases du domaine connexionniste, survole la relation entre connaissances et connexionnisme, introduit les séries temporelles et plus spécialement la représentation du temps dans les approches connexionnistes, et analyse le domaine du contrôle sous l'angle du connexionnisme.

2.1 Quelques bases

Un nombre important d'ouvrages existe où les bases du connexionnisme sont représentées sous différents angles tels que les relations mathématiques [20], ou méthodologiques [21]. D'autres sont basés sur des aspects pédagogiques bien précis comme des explications détaillées d'implémentations avec des exemples de programmes en Matlab ou la conception de réseaux [17]. Un travail qui donne un aperçu sur un grand nombre de domaines qui sont abordés par les approches connexionnistes, est la thèse de Laurent Bougrain [6].

Il n'est pas le devoir ni l'objectif de cette thèse d'ajouter quoi que ce soit à ces ouvrages. Cependant afin de compléter le cadre du travail, les aspects du connexionnisme en relation avec les domaines abordés ici seront récapitulés dans ce chapitre. Cela aussi en vue de montrer quelques faiblesses qui existent actuellement dans ces domaines et auxquelles ce travail a l'ambition d'apporter des idées voire des propositions primaires.

Le connexionnisme tel qu'il est perçu dans ce travail représente un calcul distribué où les unités locales détiennent des informations primaires. La fonctionnalité de ces unités est normalement assez limitée mais peut avoir des aspects de non-linéarités. La fonctionnalité globale résulte de l'interaction entre ces unités qui sont connectées entre elles.

Les premières idées de telles structures artificielles sont issues d'un travail de collaboration entre le neurophysiologiste Warren McCulloch et le mathématicien Walter Pitts [47] en 1943. Ils se sont inspirés de la structure du cerveau humain où les unités (neurones) sont composées d'un noyau ayant des entrées et des sorties par lesquelles ils sont interconnectés entre eux. C'est la raison pour laquelle on parle de connexionnisme, de réseaux connexionnistes, de réseaux de neurones artificiels et du domaine de l'intelligence artificielle.

Un aspect qui a une influence importante sur la fonctionnalité globale d'un réseau connexionniste est la topologie du réseau c.-à-d. la structure qui est représentée par les unités et leurs connexions. Souvent les neurones sont regroupés dans des couches. Pour chaque couche la propagation de l'information est considérée comme étant instantanée. Deux catégories principales sont distinguées si on se limite à la structure du réseau :

Réseaux feed-forward : Pour ce type de topologie, le flux d'information est unidirectionnel et va des entrées du réseau vers sa sortie. Les connexions entre unités reprennent la même structure, il n'existe pas de connexion latérale entre unités d'une même couche ni de connexions entre une couche et la couche précédente. On parle généralement de perceptrons ou perceptrons multicouches.

Réseaux récurrents : Ce type de topologie peut entre autres contenir des connexions récursives. Les cycles qui sont formés par ces connexions peuvent être locaux en affectant uniquement des unités d'une même couche ou globaux en connectant des unités d'autres couches. Il est question de réseaux du genre : Hopfield[24], Jordan[28] ou Elman[12]. Même les cartes auto-organisatrices de Kohonen[37] qui sont souvent répertoriés dans une catégorie propre peuvent être vue comme un réseau récurrent à cause du voisinage qui peut être interprété comme une récursion locale. Ce dernier type de topologie sera analysé plus en détails par après (chap. 2.2.2) et aura plus de poids dans la suite de ce travail de thèse.

Un autre aspect d'une grande importance qui est associé avec les méthodes connexionnistes est la faculté d'apprentissage sur base d'exemples. C'est ce genre d'aspect qui est à la base de l'intérêt que portent aujourd'hui aussi bien le domaine scientifique que le domaine industriel envers ces méthodes. On distingue trois formes d'apprentissage :

l'apprentissage supervisé : ce type d'apprentissage se base sur le fait qu'un corpus existe qui contient la relation entre observations et résultats. Prenons comme exemple une approche de classification simple, un nombre d'images affiche ou bien un triangle ou bien un cercle. En montrant l'image on précise à chaque fois de quelle forme il s'agit, la forme et l'étiquette associée sont données. Le résultat souhaité de la classification est connu pour le corpus d'apprentissage et le réseau est guidé durant son apprentissage.

l'apprentissage non-supervisé : ce type d'apprentissage est utilisable si le corpus ne comprend que des observations sans résultats. Il est possible d'extraire du corpus des observations typiques. Reprenons l'exemple précédent avec le corpus qui ne contient que les images de triangles et de cercles. Il est possible d'apprendre que deux différentes formes sont présentes dans le corpus et il est possible de ressortir un représentant de ces deux formes. Puisque dans ce contexte le résultat (la catégorie) n'est pas connu au préalable on parle d'apprentissage non-supervisé.

l'apprentissage semi-supervisé : ce type d'apprentissage est un mélange entre les deux précédents. C'est le genre d'apprentissage le plus répandu chez l'humain. En reprenant l'exemple précédent, un enfant joue avec des cartes qui représentent les images d'un côté et les étiquettes de l'autre côté. Le but est de mettre ensemble les cartes des images avec les bonnes étiquettes. De temps en temps la mère regarde et lui donne une récompense pour chaque bonne association. La récompense fait en sorte que la bonne relation est plus attractive et qu'elle sera apprise.

Les paramètres principaux d'un réseau connexionniste sont la topologie des couches, le nombre de couches, le nombre de neurones par couche et les types de fonctionnalités des unités.

Après cet aperçu sur les bases des réseaux connexionnistes, le vocabulaire utilisé dans ce domaine a été introduit. La suite de ce chapitre va plus spécialement traiter les propriétés que ces structures ont dans le domaine des connaissances et du traitement des aspects temporels.

2.2 La connaissance dans le connexionnisme

Avant de rentrer dans des détails sur la relation entre la connaissance et les approches connexionnistes, il est important de bien définir le terme "connaissance", les classes de connaissances qui sont considérées et les formes sous lesquelles une connaissance peut apparaître. Différents points de vue sur ce sujet existent et peuvent mener vers des problèmes de communication et de compréhension.

Une définition générique du terme connaissance a été proposée par Fischler et Firschein en 1987 [14] :

> *"Knowledge refers to stored information or models used by a person or machine to interpret, predict, and appropriately respond to the outside world."*

Selon cette définition la base de la connaissance est donc constituée des informations préexistantes ou des modèles connus. Fischler et Firschein distinguent de plus entre deux classes de connaissances :

– Les informations préliminaires :

ce sont des faits connus du monde externe. Des exemples peuvent être : des méthodes mathématiques, un domaine théorique comme la thermodynamique, des schémas d'une installation ou encore des croyances personnelles ou collectives.
- Les observations du monde externe :
  elles peuvent être réalisées à travers des méthodes de mesures directes ou indirectes ou venir d'observations personnelles. En général, les connaissances provenant d'observations ne sont pas directement accessibles. La plupart du temps, des analyses détaillées sont nécessaires.

Ensuite ils affirment que les connaissances regroupées dans une même classe peuvent apparaître sous diverses formes :

- La connaissance sous forme d'un modèle numérique.
  Un modèle numérique peut d'un côté être explicable, si la relation entre ses entrées et ses sorties sont décrites de façon explicite par des fonctions mathématiques. D'un autre côté un modèle est non explicable, si ces relations n'existent pas sous une forme explicite. On parle dans ce cas de modèles boîtes noires.
- La connaissance sous une forme symbolique.
  En général les connaissances expertes peuvent être attribuées à la classe des observations du monde externe. Si elles existent de façon explicite elles apparaissent sous une forme symbolique.

Ceci ne donne qu'un vague aperçu du sujet qui est étudié de façon scientifique par les *sciences cognitives*. Déjà la définition du terme "sciences cognitives" est délicate et source de nombreux débats. La cognition est un synonyme de connaissance et en tant que telle, pas un acquis figé. C'est pourquoi on utilise souvent le terme de "processus cognitif" [29]. Dans ce terme sont regroupées deux notions, celle de processus qui réfère au "mécanisme actif et organisé dans le temps" et celle de la cognition c.-à-d. de la connaissance.

Ce n'est pas l'objet de cette thèse de creuser en détails ce sujet mais afin de positionner ce travail quelques relations seront évoquées.

Dans le cadre de cette thèse, et sans objectif universel, la définition suivante de Jean-Daniel Kant [29] pour le terme de "processus cognitif" est adoptée :

> *"Un processus cognitif est un processus de traitement d'informations, où l'information est de nature variée : signaux sensoriels ou moteurs, symboles d'un langage, croyances personnelles ou collectives, etc. Ces traitements sont utilisés par un sujet dans un but donné : celui de réaliser certaines capacités fondamentales et caractéristiques de l'esprit humain comme le langage, le raisonnement, la perception, la décision, la planification, etc."*

Le domaine de Intelligence Artificielle (IA) distingue essentiellement deux types de modèles informatiques [13] : On a d'un côté les modèles du type symbolique. Ils se basent entre autres sur des systèmes à base de connaissance, des systèmes multi-agents ou encore des systèmes à objets. De l'autre côté on trouve les modèles du type réseaux connexionnistes qui ont déjà été évoqués.

Les modèles du type symbolique, comme par exemple les systèmes experts, sont adaptés de facto au traitement des connaissances sous une forme symbolique. Par contre le traitement de données réelles et continues comme celles que fournissent des capteurs de mesure leur pose des problèmes.

Les modèles du type réseaux connexionnistes sont particulièrement adaptés au traitement de données réelles, à la réalisation du passage du continu au discret et à faire émerger des structures comme c'est le cas dans la reconnaissance de formes (la catégorisation perceptive). Cependant leur habilité dans le traitement de données symboliques et discrètes est limitée.

Un avantage des modèles connexionnistes est leurs capacités multiples dans le domaine de l'apprentissage à partir d'exemples. Posé dans le contexte de la science cognitive, les modèles connexionnistes ont le potentiel de création ou d'extraction de connaissances en se basant sur des observations du type mesures.

2.2.1 Le modèle boîte-noire s'explique

Dans le domaine de la modélisation et de l'identification de systèmes, les modèles boîte-noire sont bien connus. Il est question d'une approche qui se base sur des informations d'entrée et de sortie d'un système afin de construire un modèle du système qui implémente la relation entrée sortie. L'utilisateur

d'un tel modèle sait identifier les entrées et sorties qui sont utilisées, mais il ne connaît pas les relations internes du modèle.

Ce genre de méthode est couramment appliqué dans le domaine industriel. Dans le domaine du connexionnisme la structure par excellence pour l'implémentation de modèles boîte-noire est le perceptron multicouches (MLP). Il a été démontré qu'un perceptron à une couche cachée est capable de réaliser toute fonction continue et multivariée [9], [25]. Globalement un MLP est capable de réaliser sur la base d'un apprentissage supervisé des modèles de boîte-noire très complexes qui réalisent un traitement non-linéaire multivarié à sorties multiples.

En faisant référence à la définition de Fischler et Firschein, cette tâche peut déjà être mise en relation avec la création de connaissances, car le modèle qui est créé est capable de donner une réponse sur le comportement d'un système dans des circonstances données.

Il est souvent difficile de se contenter de ce niveau d'abstraction de la connaissance, parce qu'elle n'explique pas les relations internes qui constituent un tel modèle boîte-noire. L'utilisateur aimerait se servir de cette connaissance afin de pouvoir tirer des conclusions sur le processus sous-jacent décrit par un tel modèle.

En principe tous les paramètres internes du modèle connexionniste sont connus, mais le nombre élevé de connexions et les relations non-linéaires multiples sont un obstacle quasi infranchissable.

Néanmoins pour les réseaux connexionnistes du type perceptron des méthodes ont été développées avec la capacité de répondre en partie à ces besoins. Deux groupes peuvent être distingués :

- Les passerelles neuro-symboliques [39].
  Le but de ces approches est de réaliser une interface entre des modèles connexionnistes et des modèles symboliques afin de pouvoir connecter les deux modèles et les utiliser chacun pour ses compétences.
  Issue de ces travaux est la possibilité de créer une base de règles décrivant un réseau connexionniste particulier (après apprentissage). On retrouve de telles approches aussi dans le domaine de la logique floue combinée aux réseaux connexionnistes [8].
- La construction dynamique de réseaux connexionnistes [6].
  Le but primaire de ces approches est de savoir combien de neurones et quelles connexions sont nécessaires pour un problème donné. Ici deux approches existent : On commence par un réseau de grande taille et on réalise un élagage afin de réduire la complexité [19], [41], ou on commence par le bas avec un minimum d'unités et on ajoute au fur et à mesure de l'apprentissage des unités afin de raffiner le réseau [15].
  Une autre retombée de ces approches est de pouvoir analyser quelles sont les relations pertinentes d'un système qui est représenté par un tel modèle.

Les modèles boîte-noire implémentent en gros une régression de fonctions qui, dans le contexte des réseaux connexionnistes, peut avoir des caractéristiques très complexes. A côté de ce domaine d'application, un autre domaine avec une importance similaire est la catégorisation ou encore la classification. Les réseaux connexionnistes ont une place indiscutable dans ce domaine.

2.2.2 Catégorisation et classification

Dans le domaine des sciences cognitives, on utilise le terme de "catégorisation" pour le "regroupement d'objets de même nature dans une classe". Un des buts de la catégorisation est d'assurer la transition du continu au discret afin d'améliorer l'adaptation à l'environnement. Aussi dans la représentation de connaissances la catégorisation joue un rôle non négligeable. Le processus de formation de catégories est directement associé à l'apprentissage.

Le terme de "classification" réfère à la détermination de la classe d'appartenance d'un objet donné. La classification intervient dans le processus de raisonnement et de prise de décision. Deux activités cognitives sont distinguées : le choix qui intervient dans la sélection d'un objet appartenant à la même classe qu'un autre objet, sans se préoccuper des caractéristiques de la classe et le jugement où la sélection d'un objet se fait par le fait d'appartenir à une catégorie spécifique.

Dans le domaine du connexionnisme les deux tâches sont connues. Cependant la terminologie utilisée dans le connexionnisme est légèrement différente en ce qui concerne la catégorisation. La catégorisation est directement liée à la phase d'apprentissage par laquelle passe la construction d'un réseau connexionniste.

C'est pourquoi deux types de catégorisations existent pour le connexionnisme et ils se distinguent au niveau de l'apprentissage : il est question de la classification supervisée et de la classification non-supervisée. Notons qu'il est question de catégorisation, même si le terme classification est utilisé.

Le terme de "'classification supervisée"' est utilisé quand il s'agit d'un apprentissage supervisé. Au préalable, un corpus de données existe, qui contient des exemples de caractéristiques avec les classes auxquelles ils appartiennent. Dans ce cas le réseau apprend la relation qui existe entre caractéristiques et classe.

Le terme de classification non-supervisée est utilisé quand un apprentissage non-supervisé est utilisé. Il s'agit dans ce cas de regrouper des exemples qui ont des caractéristiques similaires sans savoir au préalable quelles sont les classes qui existent. Dans le contexte de cette étude c'est ce type de regroupement qui va jouer un rôle important et il sera analysé plus en détail dans la suite.

Berned Fritzke, dans [15] regroupe et analyse diverses méthodes utilisées pour la classification non-supervisée en se référant au mécanisme d'apprentissage commun, qu'il regroupe sous le terme d'apprentissage compétitif. Pour toute personne intéressée à ce sujet ce papier est fortement conseillé.

Dans la suite de ce travail certains aspects de ces méthodes vont revenir et être analysés plus en détails. Il est donc important de donner au lecteur un résumé des méthodes les plus utilisées afin de le familiariser avec le sujet.

2.2.2.1 Définition de l'environnement mathématique

Les méthodes analysées par Fritzke se limitent à un certain type en matière de classifications qui est défini dans l'environnement mathématique ci-dessous. D'autres domaines proposent d'autres approches, mais pour le cadre de ce travail ils ne seront pas considérés.

Le but global de toutes ces méthodes de classification non-supervisée est de distribuer un nombre N de vecteurs de référence, aussi connus sous le nom de prototypes, dans l'espace analysé.

$$\mathcal{A} = \{c_1, c_2, \cdots, c_N\} \tag{2.1}$$

$$w_c \in \mathcal{R}^n \tag{2.2}$$

La distribution des prototypes doit refléter d'une manière ou d'une autre la probabilité de distribution des observations qu'il s'agit d'analyser.

Entre ces prototypes peut exister une relation de voisinage qui généralement est représentée par un ensemble de connexions.

$$\mathcal{C} \subset \mathcal{A} \times \mathcal{A} \tag{2.3}$$

De ce fait les voisins topologiques peuvent être trouvés par la relation suivante.

$$N_c = \{i \in \mathcal{A} | (c, i) \in \mathcal{C}\} \tag{2.4}$$

Les observations sont représentées par un corpus d'exemples.

$$\mathcal{D} = \{\xi_1, \cdots, \xi_M\}, \text{ avec } \xi_i \in \mathcal{R}^n \tag{2.5}$$

Ceci décrit l'environnement commun pour toutes les méthodes analysées par Fritzke. Une autre approche commune de ces méthodes est la notion de prototypes gagnants. Pour une observation donnée ξ le prototype gagnant $s(\xi)$ est celui avec le vecteur de référence le plus proche de l'observation.

$$s(\xi) = \arg\min_{c \in \mathcal{A}} d(\xi - w_c) \tag{2.6}$$

Le plus souvent la métrique euclidienne est utilisée pour définir la distance entre les deux vecteurs.

Si les i plus proches prototypes sont utilisés la notation sera $s_i(\xi)$ ou tout simplement s_i.

Toutes les observations pour lesquelles un prototype est gagnant composent un sous-ensemble qui est représenté par la relation suivante.

$$\mathcal{R}_c = \{\xi \in \mathcal{D} | s(\xi) = c\} \tag{2.7}$$

Un tel sous-ensemble est aussi appelé un "ensemble de Voronoi". Fritzke dans son papier fait référence à deux concepts du domaine de la géométrie algorithmique qui sont étroitement liés aux méthodes discutées ici. Il est question des diagrammes de Voronoi et de la triangulation de Delaunay. Ce sujet ne sera pas approfondi ici, le lecteur intéressé est renvoyé au papier de Fritzke [15].

2.2.2.2 Les mesures classiques pour l'optimisation

Afin de positionner les prototypes dépendant des observations, deux méthodes principales sont analysées par Fritzke : la minimisation de l'erreur et la maximisation de l'entropie.

La méthode qui est utilisée le plus souvent est de minimiser l'erreur de distorsion globale représentée par la formulation suivante.

$$E(\mathcal{D}, \mathcal{A}) = 1/|\mathcal{D}| \sum_{c \in \mathcal{A}} \sum_{\xi \in \mathcal{R}_c} \|\xi - w_c\|^2 \tag{2.8}$$

Il s'agit de placer les prototypes de telle façon qu'ils représentent au mieux la distribution de l'ensemble des observations.

Si le but est de distribuer les prototypes de façon à ce que chacun ait la même probabilité d'être sélectionné comme gagnant, il faut pour une observation aléatoire ξ que

$$P(s(\xi) = c) = \frac{1}{|\mathcal{A}|} \tag{2.9}$$

Si on considère l'opération d'attribution d'une observation ξ au prototype le plus proche c comme une opération aléatoire qui attribue $x \in \mathcal{A}$ à une variable aléatoire $\mathcal{X}$, la relation (2.9) est obtenue en maximisant l'entropie.

$$H(X) = -\sum_{x \in \mathcal{A}} P(x) log(P(x)) = E(log(\frac{1}{P(x)}) \tag{2.10}$$

Ces deux approches ne peuvent normalement ne pas être atteintes en même temps, sauf éventuellement dans le cas où les observations sont uniformément distribuées, ce qui n'est généralement pas le cas.

2.2.2.3 Approches où un seul prototype est considéré

En ce qui concerne les algorithmes, Fritzke distingue entre deux types d'approches. Pour le premier type d'approches, pour une observation donnée un seul prototype est gagnant et sera modifié dans la phase d'apprentissage. Le deuxième type d'approches comporte plusieurs cas et va être discuté ensuite.

Pour la première approche, on distingue deux méthodes d'adaptation des prototypes : la méthode batch et la méthode en ligne.

La méthode batch considère l'ensemble des observations avant d'adapter les prototypes, ce qui est seulement faisable si toutes les observations sont disponibles au départ. La méthode en ligne réalise pour chaque observation une adaptation du prototype gagnant et est de ce fait utilisable aussi dans le cas où les observations apparaissent successivement et ne peuvent être stockées.

La méthode batch est aussi appelée LBG (generalized Lloyd algorithme) [43]. L'algorithme répète itérativement le positionnement des prototypes comme moyenne arithmétique de leur sous-ensemble de Voronoi respectif.

$$w_c = \frac{1}{|\mathcal{R}_c|} \sum_{\xi \in \mathcal{R}_c} \xi \tag{2.11}$$

Un problème général de cette méthode est le cas des prototypes dits morts. Ces prototypes ne sont jamais gagnants et de ce fait ils n'interviennent pas dans la distribution.

Pour la méthode en ligne une autre approche est adoptée. Elle consiste à bouger le prototype gagnant vers l'observation qui lui est attribuée.

$$\Delta w_s = \epsilon(\xi - w_s) \tag{2.12}$$

ϵ est appelé le taux d'apprentissage qui détermine de combien le prototype sera bougé. De nombreuses approches existent afin de fixer le taux d'apprentissage. Le plus souvent le taux est diminué avec le nombre d'itérations afin de stabiliser la position vers la fin de l'apprentissage.

La méthode la plus utilisée pour ce type d'approches est connue sous le nom de K-means. Cependant différentes implémentations de cette méthode existent sous ce même nom [56]. Celle à laquelle ce travail se réfère utilise un taux d'apprentissage spécifique pour chaque prototype, qui est diminué avec le nombre de fois que le prototype est sélectionné comme gagnant.

2.2.2.4 Approches avec une structure de réseau dynamique

La méthode la plus connue pour les réseaux adaptatifs est l'approche appelée "Growing Neural Gas" [15] qui combine deux approches classiques, "Neural Gas" [46] et la méthode d'apprentissage compétitif de Hebb. La méthode commence par un nombre restreint de prototypes auxquels s'ajoutent des prototypes durant l'apprentissage. Ils sont ajoutés près des prototypes qui représentent les plus grandes erreurs locales.

Par un mécanisme basé sur l'âge des connexions entre les prototypes qui n'ont pas été modifiés, la topologie du réseau est modifiée en enlevant des connexions trop vieilles. En même temps, des prototypes qui n'ont plus de connexions sont éliminés.

L'adaptation des prototypes est réalisée selon (2.12) pour le prototype gagnant avec un taux d'apprentissage spécifique pour le gagnant. En même temps les voisins topologiques sont adaptés avec un autre taux d'apprentissage. Ces taux d'apprentissage restent fixes durant tout l'apprentissage.

Le résultat est un réseau pour lequel aussi bien la topologie que le nombre de prototypes sont adaptés aux observations.

Deux autres méthodes existent, qui sont similaires à celle de "Growing Neural Gas". "Growing Cell Structures" diffère en gros seulement par le fait que les connexions entre les unités doivent toujours former des cellules de taille k. Les unités représentent les noeuds de ces cellules. Des cellules en forme de triangle sont obtenues pour $k = 3$. L'ajout de prototypes est réalisé en intercalant une nouvelle unité à l'endroit de la plus longue connexion. Cette connexion est réformée par des connexions à la nouvelle unité. "Growing Grid" utilise une structure en grille rectangulaire et peut être comparé aux cartes auto-organisatrices qui sont traitées dans la partie suivante.

2.2.2.5 Les cartes auto-organisatrices de Kohonen

Les cartes auto-organisatrices ont été proposées par Kohonen [37] qui s'était inspiré des travaux de Willshaw and Von der Malsburg [60]. Elles sont aussi couramment appelées SOM (Self-organizing Maps). Ce type de réseau sera présenté un peu plus en détail parce que cette approche sera utilisée plus tard.

La topologie de ce réseau est définie au départ et reste inchangée durant l'apprentissage. Elle représente le plus souvent une grille en deux dimensions (une ou trois dimensions sont possibles). Pour l'explication, le cas classique de deux dimensions sera adopté avec $m = m_1 \cdot m_2$ le nombre de prototypes sur la grille de taille $[m1 \times m2]$. La localisation du prototype sur la grille est donnée par $a_{i,j}$, donc aux coordonnées $[i, j]$ de la grille 2D (fig. 2.1).

Ce qui peut créer confusion est que le vecteur représentatif du prototype, c.-à-d. le vecteur des poids $(\vec{w}_k)$ qui connecte le prototype k du réseau au vecteur d'entrée $(\vec{x})$, a la dimension du vecteur d'entrée n comme pour les méthodes précédentes. Avec $k \in 1 \cdots m$ étant le numéro de référence du prototype.

L'espace topologique est quasi indépendant de l'espace d'entrée dans lequel se trouvent les observations. La figure 2.1 montre la différence entre ces deux espaces.

Cette séparation des espaces se retrouve aussi dans la notion de voisinage qui est définie par l'espace de la topologie. Normalement la distance de "Manhattan" est utilisée pour définir le voisinage.

$$d(r, s) = |i - k| + |j - m| \quad \text{pour } r = a_{k,m} \text{ et } s = a_{i,j} \tag{2.13}$$

Dans le cas spécial où l'espace des observations est aussi de dimension deux, il est possible de représenter la topologie et la position du vecteur de référence dans un même repère. Ce cas est souvent utilisé pour l'explication de la méthode d'adaptation des prototypes pour l'apprentissage. Notons cependant que la séparation de l'espace topologique et l'espace d'entrées reste vraie.

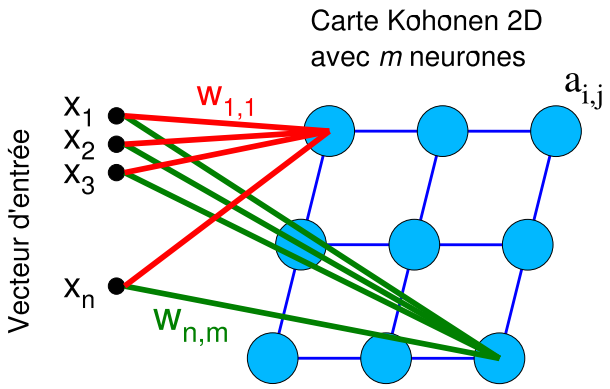


FIG. 2.1 – Carte auto-organisatrice de Kohonen (SOM) 2D avec le vecteur d'entrée $\vec{\xi} = \{x_1, \cdots, x_n\}$ et les vecteurs de référence représentés par $\vec{w}_k = \{w_{1,k}, \cdots, w_{n,k}\}$

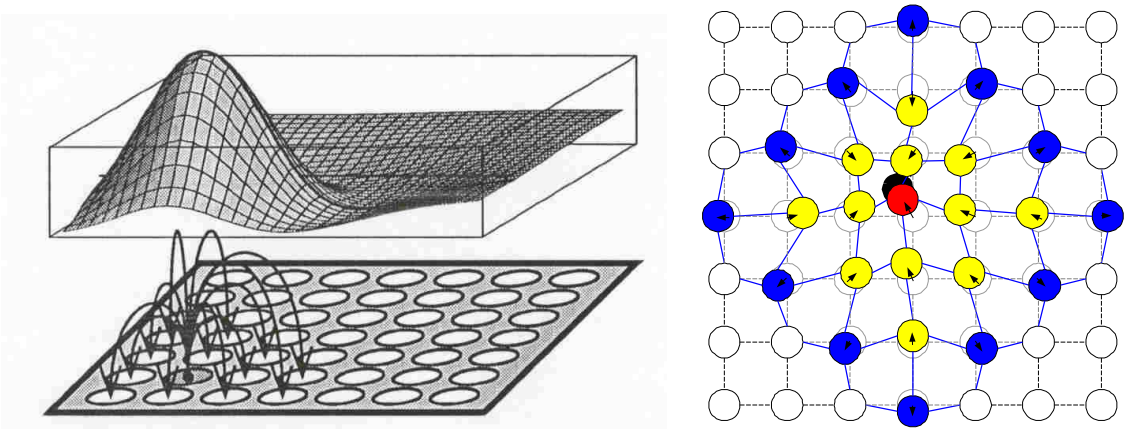
Cette séparation des espaces est la source de la propriété de réduction de complexité en gardant une information de topologie, ce qui est intéressant pour la représentation d'observations multidimensionnelles dans un espace de deux dimensions. Deux observations attribuées à deux prototypes qui sont voisins sur la carte sont voisines dans l'espace d'entrée. Le contraire n'est pas vrai : deux observations qui sont attribuées à deux prototypes qui ne sont pas voisins peuvent cependant être voisines dans l'espace d'entrée.

En ce qui concerne l'adaptation des prototypes, le prototype gagnant pour une observation donnée est adapté, ainsi que son voisinage, en utilisant une forme légèrement modifiée de celle utilisée au (2.12).

$$\Delta w_r = \epsilon h_{rs}(t)(\xi - w_r) \tag{2.14}$$

Avec ϵ le taux d'apprentissage, $h_{rs}(t)$ le niveau d'adaptation pour un prototype quelconque r de la grille en sachant que s est le prototype gagnant (fig. 2.2(a)).

$$h_{rs}(t) = exp\left(\frac{-d(r,s)^2}{2\sigma(t)^2}\right) \tag{2.15}$$



(a) (b)

FIG. 2.2 – L'adaptation des prototypes pour la SOM :
(a) La fonction qui définit le niveau d'adaptation dépendant du voisinage ;
(b) L'effet de voisinage sur l'adaptation des prototypes

Avec $\sigma(t)$ la variance de la Gaussienne (2.15) qui normalement est modifiée avec l'avancement de l'apprentissage.

$$\sigma(t) = \sigma_i(\sigma_f/\sigma_i)^{t/t_{max}} \tag{2.16}$$

σ_i étant la valeur initiale et σ_f la valeur finale de la variance.

La figure 2.2(b) montre le résultat dans le cas spécial de l'espace d'entrée à deux dimensions, où les prototypes sont uniformément distribués dans l'espace d'entrée. Le cercle noir montre la position de l'observation donnée pour laquelle le prototype rouge est le gagnant. Supposons que la fonction $h_{rs}(t)$ soit positive pour un voisinage $d(r,s) \leq 2$, négative pour $2 < d(r,s) \leq 3$ et nulle pour le reste.

Les prototypes jaunes se trouvent dans le voisinage positif et sont attirés par l'observation et les prototypes bleus dans le voisinage négatif et repoussés par l'observation. Tous les autres prototypes ne sont pas modifiés. A noter que seuls les vecteurs représentatifs des prototypes sont modifiés, pas les liens de voisinage.

Normalement l'apprentissage commence avec un voisinage large afin de positionner la carte entière dans l'espace réellement occupé par les observations. Avec l'avancement de l'apprentissage, les prototypes se spécialisent de plus en plus afin de représenter un regroupement d'observations spécifiques. Ainsi en fin d'apprentissage les vecteurs représentatifs de la carte représentent la distribution des observations dans le corpus utilisé pour l'apprentissage.

Avec la présentation des cartes auto-organisatrices de Kohonen la partie connaissance et connexionnisme est terminée. Un autre domaine du connexionnisme qui joue un rôle important pour les études faites dans le cadre de cette thèse est le temps et la considération des effets temporels dans les approches connexionnistes.

2.3 Le temps et le connexionnisme

Le temps est un élément principal de notre univers. Il est considéré comme la quatrième dimension. Un grand nombre de domaines qui sont analysés traitent directement ou indirectement de phénomènes qui varient dans le temps comme par exemple la vision, la parole, le traitement de signaux et le contrôle. Le temps joue aussi un rôle non négligeable dans toutes les approches d'apprentissage.

Le temps est en principe un phénomène continu mais peut par des méthodes d'échantillonnage aussi apparaître sous une forme discrète. Indépendamment de la forme d'apparition du temps il représente une suite d'apparitions ordonnées où l'ordre est un des aspects les plus importants.

Afin d'utiliser pour la suite des approches temporelles, la terminologie utilisée dans le contexte de l'identification de système cette dernière est introduite. Le modèle classique pour approcher un système dynamique est représenté dans la figure 2.3

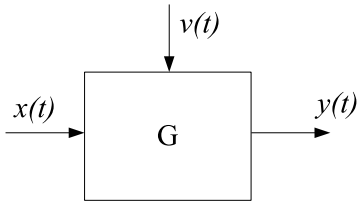


FIG. 2.3 – Représentation classique d'un système

Avec $x(t)$ les entrées mesurables du système, $y(t)$ la sortie du système et $v(t)$ les perturbations non mesurables du système. G est appelé la fonction de transfert du système. Pour un système

$$y(t) = G(x(t), v(t)) \tag{2.17}$$

Dans la théorie du contrôle, souvent une simplifications est faite, dans laquelle est supposé que les perturbations, qui sont généralement considérées comme inconnues, peuvent être ajoutées à la sortie du système (2.18).

$$y(t) = G(x(t)) + \tilde{v}(t) \tag{2.18}$$

L'avantage de cette simplification est que la fonction de transfert ne dépend que des entrées $x(t)$.

Pour l'approche connexionniste, la question principale est comment il est possible d'incorporer la notion de temps dans la structure des réseaux connexionnistes. En principe, deux méthodes sont distinguées [21] :

- *L'incorporation implicite.* Dans cette approche le temps est représenté par une proposition sans exister de façon formelle dans le réseau. Par exemple pour un signal échantillonné uniformément il est possible de montrer au réseau connexionniste différents pas de temps en même temps sur différentes entrées. On parle dans ce cas aussi de représentation spatio-temporelle, parce que la structure temporelle du signal est représentée dans une forme spatiale à l'entrée du réseau.
- *L'incorporation explicite.* Dans ce cas le temps obtient une représentation concrète dans la structure du réseau connexionniste. Deux approches complètement différentes sont actuellement connues : la première, qui existe depuis une vingtaine d'années, utilise la récursion afin d'incorporer de manière explicite une composente de mémoire temporelle (un historique) qui est appelé contexte, et la deuxième, qui est plus récente, se base sur des modèles plus proches du neurone biologique en modélisant la dynamique du flux d'informations à l'intérieur du réseau neuronal.

Les réseaux dynamiques encore appelés réseaux à spikes (spiking neurons) sont relativement récents et jusqu'à présent pas encore bien maîtrisés. Ce type de réseaux ne sera pas présenté ici.

2.3.1 L'approche spatio-temporelle

Les approches spatio-temporelles se basent principalement sur des représentations du temps dans une forme discrète. Une certaine fenêtre représentant une séquence donnée de pas de temps est présentée à un instant donné en entrée d'un réseau connexionniste. L'instant d'après la fenêtre est décalée d'un pas de temps. Pour le réseau l'information du temps est spatiale et le réseau traite cette information comme un vecteur d'entrée multidimensionnel.

Le fondement mathématique pour cette approche vient du domaine de l'identification de système par réponse impulsionnelle [44]. Un outil utilisé pour décrire des systèmes à temps discret est la "transformation en z". Pour une séquence temporelle $\{x(t)\}$ qui peut théoriquement aller à l'infini dans le passé, la transformation en z est définie par :

$$X(z) = \sum_{t=-\inf}^{\inf} x(t) z^{-t} \tag{2.19}$$

avec z^{-1} l'opérateur qui réalise un décalage unitaire dans le temps. C.-à-d. appliquer z^{-1} à $x(t)$ donne $x(t-1)$.

Appliquons $\{x(t)\}$ à un système représenté par sa réponse impulsionnelle $h(t)$. La réponse du système est représentée par la somme de la convolution.

$$y(t) = \sum_{k=-\inf}^{\inf} h(k) x(t-k) \tag{2.20}$$

Après une transformation en z la convolution temporelle est transformée en multiplication dans z.

$$Y(z) = H(z) X(z) \tag{2.21}$$

Où $H(z)$ est la fonction de transfert du système.

Une des structures de réseau qui s'appuie sur cette théorie est connue sous le nom de TLFN (Time Lagged Feedforward Network) (fig. 2.4) [21]. Ce type de réseau a l'avantage d'utiliser la structure d'un réseau aux propriétés d'apprentissage bien connues et maîtrisées : le perceptron multi couches (MLP)[23].

L'avantage de cette approche est que contrairement aux approches classiques les réponses impulsionnelles sont représentées par un modèle non-linéaire et qu'un apprentissage sur base d'exemples est possible.

Une structure qui représente une approche similaire mais pour laquelle aussi à l'intérieur du réseau (pour les couches cachées) un tel décalage spatio-temporel est réalisé est appelée "Time Delayed Neural Networks" (TDNN) [21], [20], [11].

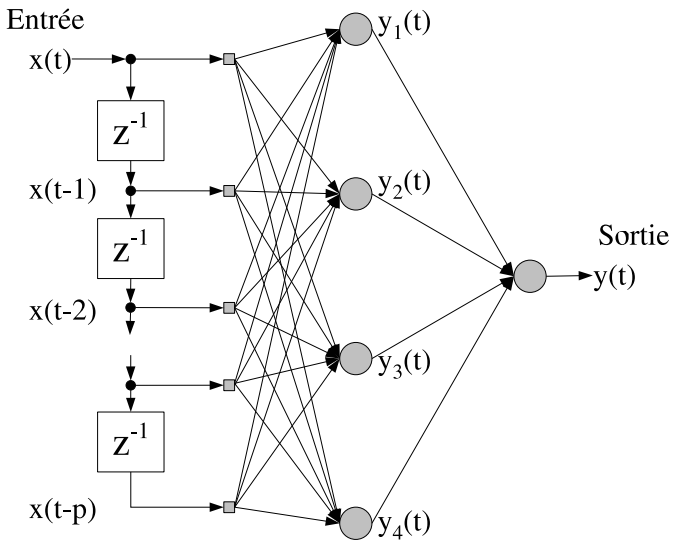


FIG. 2.4 – Structure du réseau TLFN

Une autre approche est de représenter le comportement temporel en forme de carte de séquences d'indices spatiaux et de regrouper ces cartes dans des catégories. Cette approche est appelée "Temporal Organization Map" (TOM) [11] et se base sur des modèles de la colonne corticale du cerveau.

2.3.2 Les réseaux récurrents

La définition d'un réseau récurent est qu'il contient au moins une connexion entre un neurone et son prédécesseur ou entre deux neurones d'une même couche. L'idée de base est que cette récursion intègre un certain contexte, une mémorisation du passé et avec cela une forme explicite du temps. En général ce genre de réseaux ne sont pas couramment utilisés à cause de la difficulté de bien maîtriser le comportement de telles structures.

Il existe cependant deux types de réseaux récurrents qui sont proches de la structure des MLP. Ces types de réseaux récurrents sont vus comme une extension des réseaux feed-forward [23] qui ajoutent au vecteur d'entrées un vecteur de contexte interne du réseau. Ce contexte représente un certain historique de l'activation du réseau. Les deux types se distinguent par le genre de contexte qui est représenté. La figure 2.5 représente un schéma de principe des deux structures.

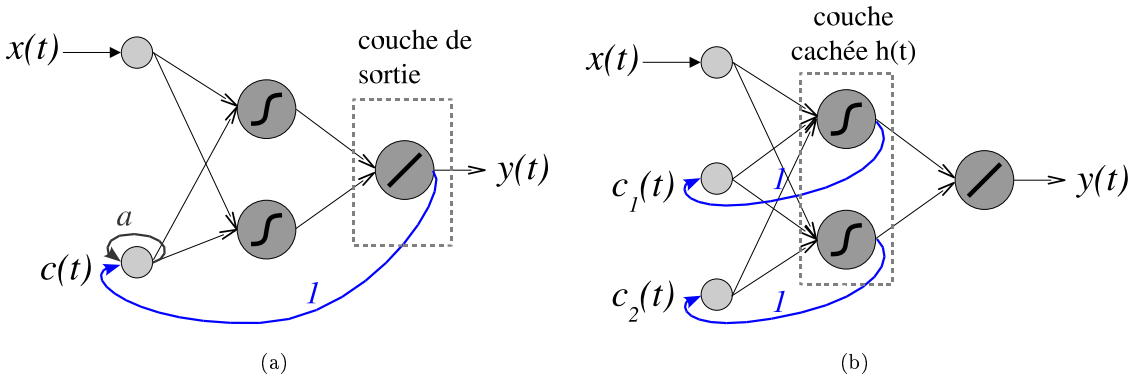


FIG. 2.5 – Deux types de réseaux récurrents : (a) Le réseau de Jordan ; (b) Le réseau d'Elman.

Pour les deux réseaux, $x(t)$ représente le vecteur d'entrées. Il peut représenter une approche multiva-

riable ou une structure de TLFN, ou les deux combinées. La sortie du réseau $y(t)$ peut aussi représenter une sortie multiple. La partie $c(t)$ représente le contexte.

Le réseau de Jordan [28] (fig. 2.5(a)) présenté en 1986 utilise deux types de récursions : la première, qui copie l'information de sortie dans une couche de contexte qui pourra être utilisée pour la prochaine entrée. La deuxième récursion est aussi appelée auto-récursion et copie l'information de contexte sur elle-même avec un certain poids α. Cela garantit de manière explicite qu'un certain historique du contexte reste dans la mémoire du réseau. Avec cette structure l'information de contexte pour un instant t est définie par la formule (2.22). A cause de l'auto-récursion cette formule est récursive et peut être développée (2.23).

$$\begin{aligned} c(t) &= y(t-1) + \alpha \cdot c(t-1) && (2.22)\\ c(t) &= y(t-1) + \alpha(\underline{y(t-2) + \alpha c(t-2)}) \\ c(t) &= y(t-1) + \alpha y(t-2) + \alpha^2 c(t-2) \\ c(t) &= y(t-1) + \alpha y(t-2) + \alpha^2 y(t-3) + \cdots + \alpha^n c(t-n) && (2.23) \end{aligned}$$

Avec $\alpha < 1$ l'histoire du contexte perd de la valeur avec le temps. La sortie la plus récente est donc prise en compte plus fortement.

Le réseau d'Elman [12] (fig. 2.5(b)) n'utilise qu'un seul type de récursion. Il copie l'information contenue dans la couche cachée $h(t-1)$ sur la couche de contexte. Pour un instant donné cela représente :

$$c(t) = h(t-1) \tag{2.24}$$

Cela a l'air d'être moins complexe mais c'est trompeur car :

$$h(t-1) = F(x(t-1); c(t-1)) \tag{2.25}$$

où $F()$ est en général une fonction non-linéaire. De plus le contexte précédent est pris en compte pour le calcul de l'activation de la couche cachée.

Dans le réseau de Jordan la sortie désirée est généralement une grandeur qui peut être interprétée. Le réseau d'Elman utilise la couche cachée pour la récursion. Cette couche représente généralement une structure complexe qui résulte de l'apprentissage et qui représente un codage interne du réseau qui peut difficilement être interprété.

En général on dit que ce genre de réseaux implémente une mémoire à court, voire très court terme.

Ces quelques approches représentent différentes techniques pour prendre en compte les effets temporels dans les systèmes connexionnistes. Dans des contextes bien précis il a été montré que ces approches donnent des résultats appréciables. Cependant dans le cadre de cette étude deux aspects sont importants : la possibilité de représenter un système très complexe, et la prévision sur le long terme. Ces deux aspects ensemble font que les méthodes présentées ne sont pas directement utilisables.

2.4 Le contrôle et le connexionnisme

Dans le contexte de processus industriels le contrôle est une tâche classique. Ce domaine est analysé et discuté depuis longtemps dans le monde scientifique. Un grand nombre de fondements mathématiques existent sur le contrôle. Dans l'industrie les tâches de contrôle sont aujourd'hui encore dominées par les approches classiques et souvent linéaires.

Il est clair que ce domaine a été approché par des méthodes connexionnistes avec le but de pouvoir utiliser des modèles non-linéaires, afin de pouvoir améliorer le contrôle des systèmes qui sont souvent de nature non-linéaire. Néanmoins, le domaine du contrôle ne joue qu'un rôle secondaire dans le monde du connexionnisme.

Cette section va donner un aperçu sur quelques approches connexionnistes dans le domaine du contrôle [26], [50]. S'il est question d'un réseau connexionniste, il est souvent question d'un réseau MLP ou des variantes qui ont été présentées dans le chapitre 2.3. L'acronyme qui sera utilisé pour un tel réseau est ANN (Artificial Neural Network).

2.4.1 Le contrôle par apprentissage supervisé

Le contrôle par apprentissage supervisé se base sur le fait que la tâche de contrôle est déjà réalisée par un autre contrôle. L'apprentissage supervisé du réseau connexionniste est réalisé sur la base des entrées et sorties de ce contrôleur préexistant. La figure 2.6 montre la structure qui est utilisée.

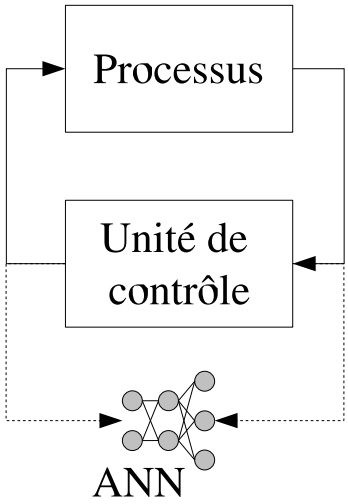


FIG. 2.6 – Contrôle par apprentissage supervisé

Les intérêts de ce type d'approche sont :
- le transfert de connaissance d'experts. Le système peut apprendre ce qu'un expert, ou un système existant fait réellement, ce qui peut être différent de ce qui est écrit ou encore de ce que l'expert explique.
- La tâche de contrôle peut en suite éventuellement être réalisée à une vitesse plus élevée.
- Il est possible de prendre en compte d'autres informations en parallèle aux entrées et sorties d'un contrôle existant. Cela est encore plus intéressant si le contrôle est réellement fait par un opérateur.

2.4.2 Le contrôle par modèle inverse

Cette approche consiste à apprendre le système inverse d'un processus et à utiliser ce réseau comme unité de contrôle. L'avantage est que l'opérateur peut spécifier les consignes dans l'espace de sortie du processus. La figure 2.7 montre les deux phases de l'approche.

Souvent cette approche n'est pas simple car diverses manipulations des entrées d'un processus peuvent conduire à un même résultat, ce qui fait que le modèle inverse du processus n'est pas bien défini.

C'est une méthode intéressante pour les situations où aucune unité de contrôle n'existe. Le corpus d'apprentissage peut être construit en utilisant différents types d'activation du processus et en enregistrant les sorties.

2.4.3 Structure de contrôle avec modèle interne

Cette approche utilise un modèle inverse du processus pour l'unité de contrôle et un modèle directe du processus pour une tâche de prédiction. Par le retour de l'erreur de prédiction, une approche de contrôle adaptative est possible. La figure 2.8 montre le schéma de cette approche.

Le modèle du processus est appris par les approches classiques d'identification de système, en utilisant un réseau du type TLFN ou récurrent afin d'intégrer une approche de prédiction. D'autres approches de contrôle prédictif utilisent la rétro-propagation de l'erreur en passant par le modèle du processus afin de connaître l'erreur au niveau de la sortie de l'unité de contrôle.

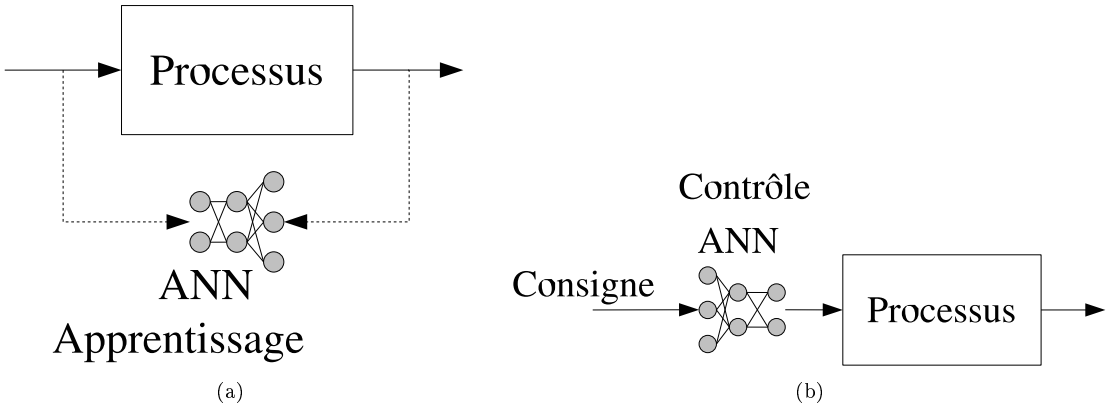


FIG. 2.7 – Le contrôle par modèle inverse :
(a) Pour l'apprentissage le processus est utilisé de façon inverse ;
(b) Après l'apprentissage le réseau est utilisé comme unité de contrôle.

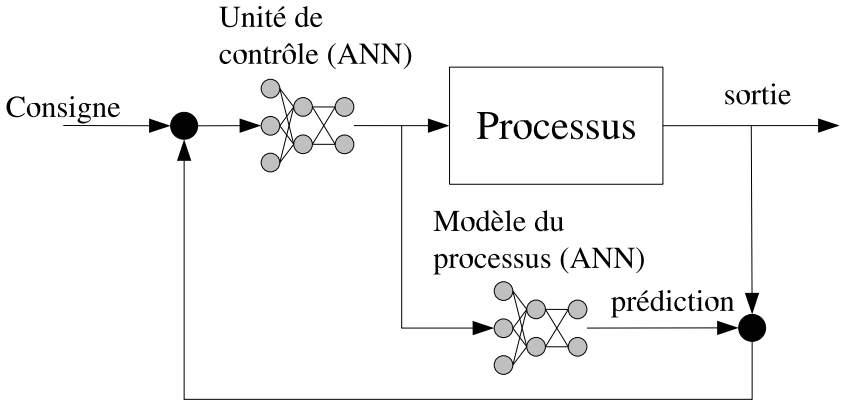


FIG. 2.8 – Contrôle basé sur un modèle interne et un modèle inverse.

2.5 Conclusion de ce chapitre

Dans ce chapitre le langage du domaine connexionniste est introduit et quelques méthodes de base du domaine sont évoquées. Une grande partie du chapitre traite la relation entre le connexionnisme et les connaissances. Les approches qui existent pour la représentation ou l'extraction de connaissances sont citées et un point fort est mis sur le sujet de la catégorisation par le biais de la classification non-supervisée du domaine connexionniste. Le rôle du temps et les moyens pour considérer les aspects temporels par les approches connexionnistes sont survolés. Finalement quelques approches du domaine du contrôle sont abordées.

Ce chapitre avait pour but de fournir toutes les bases nécessaires afin de pouvoir suivre les démarches qui sont faites dans les chapitres suivants.

Chapitre 3

Analyse de séries temporelles

Sommaire

Le temps est partout, et chacun est quotidiennement confronté à des séries temporelles. Chacun mesure des choses : son propre poids le matin, la température de l'extérieur, la vitesse de la voiture, la tension artérielle, le cours des actions à la bourse, le nombre de visites sur un site web etc. et toutes ces mesures, qui sont prises à un instant donné, changent avec le temps.

C'est pourquoi de nombreuses recherches sur le traitement et l'analyse de séries temporelles existent. Cependant elles sont souvent spécifiques à un domaine, et les méthodes utilisées dans un domaine ne sont pas nécessairement adaptées aux autres domaines, et des fois elles y sont même inconnues.

Ce chapitre va traiter en premier lieu quelques spécificités des données temporelles afin de préparer le terrain et de familiariser le lecteur avec le langage spécifique. Ensuite un codage basé sur une extraction des formes primitives qui compose le signal analysé et qui de ce fait s'adapte aux signaux analysés sera présenté (voir 1a Fig.1.10). Dans la dernière section une méthode de classification non-supervisée capable de traiter des séries de taille quelconque basée sur une mesure de similarité est proposée (voir 1b Fig1.10).

Toutes les démarches seront accompagnées d'exemples simples afin de les illustrer. A la fin de chaque section, l'approche est appliquée aux données de l'installation industrielle afin, de construire au fur et à mesure les outils pour la réalisation d'une approche de contrôle prédictif basé sur l'aide à la décision telle que introduite dans le chapitre 1.4.

3.1 Quelques spécificités des données temporelles

En général, le volume de données à manipuler dans le domaine de l'analyse de séries temporelles est important. Afin de simplifier le traitement de ces données et de limiter les temps de calcul d'analyses, une des premières approches est la réduction efficace de données.

Le domaine de la réduction de données est un domaine de recherche à part entière avec beaucoup de facettes. Le but primaire est de réduire au maximum le volume de données sans pour autant perdre des informations importantes. Définir ce qui a de l'importance dans un échantillon de données est une tâche non triviale et elle est fortement corrélée à la source qui est représentée par ces données et au domaine d'application. Dans certains domaines, comme la compression de fichiers informatiques, aucune perte de données ne peut être acceptée, dans d'autres domaines, comme par exemple la compression de musique ou d'images, un certain niveau de perte peut être accepté au détriment de la qualité du son respectivement de l'image. Pour certaines applications une extraction de quelques caractéristiques suffit au besoin d'une analyse ultérieure, comme par exemple, les points caractéristiques du signal cardiaque.

Une autre tâche importante dans le traitement de données temporelles est l'indexation. Elle est directement liée à la propriété de la taille des données. Retrouver dans un pool de données temporelles les séquences temporelles avec des propriétés spécifiques est seulement réalisable dans un temps raisonnable s'il existe une indexation adéquate.

Une tâche très liée aux séries temporelles est la localisation d'une séquence donnée dans un flux de données temporelles. C'est-à-dire, quels sont les instants où la séquence donnée ressemble "bien" à une partie du flux analysé.

3.1.1 Le codage de données

Dans le domaine du traitement du signal, les signaux à manipuler apparaissent à la base, dans la plupart des cas, sous une forme analogique. Des approches de traitement des signaux analogiques existent, mais elles ne font pas l'objet de cette thèse. Afin de pouvoir manipuler les signaux sur base d'ordinateur, la première étape consiste à digitaliser les signaux analogiques. Déjà cette première étape, qui est aussi appelée "échantillonnage des signaux analogiques", est une forme de codage qui dans les applications réelles est souvent une source de perte d'informations. L'échantillonnage de signaux est un sujet complexe en soi. En général les fréquences de grandeurs physiques représentant un phénomène complexe ne sont pas connues. Afin d'éviter une perte d'information au niveau de ce codage primaire, selon le théorème de Shannon (théorie de l'information), la fréquence d'échantillonnage minimale devrait être le double de la fréquence maximale qui apparaît dans la grandeur physique mesurée [40], [49]. Afin de garantir cette contrainte pour des mesures réelles, un filtrage analogue adapté à l'acquisition de données et aux analyses

prévues serait nécessaire. Il faut cependant souligner qu'un tel filtrage réalise un lissage des signaux et de ce fait enlève la partie des hautes fréquences (perte d'information potentielle).

Ce travail se base sur un échantillonnage uniforme avec un temps d'échantillonnage constant. Pour cette forme d'échantillonnage, il est important d'adapter le temps d'échantillonnage aux problèmes posés. D'un côté, il faut éviter des pertes d'informations indispensables à l'analyse, ce qui mène vers une fréquence d'échantillonnage élevée, de l'autre côté, le volume de données généré par la digitalisation augmente avec la fréquence d'échantillonnage.

La digitalisation est normalement localisée dans la partie de l'acquisition de données qui est souvent imposée par un système déjà en place. Il est important de connaître les paramètres du système d'acquisition et de la digitalisation afin de pouvoir interpréter l'allure des signaux, leurs évolutions et les résultats d'analyses basés sur ces signaux. L'inertie d'un capteur, un filtrage du signal analogique, la méthode d'échantillonnage ou un filtrage digital du système d'acquisition, pour n'en citer que quelques-uns, ont une forte influence sur l'allure des données et donc forcément aussi sur les résultats d'analyses.

Après l'acquisition et avec celle-ci la digitalisation des signaux, le codage proprement dit est utilisé dans beaucoup de domaines à des fins de réduction de données. Mais la réduction de données n'est pas la seule finalité du codage. Le codage peut aussi être utilisé à des fins de représentation de données sur différents niveaux d'abstraction. Une représentation macroscopique de données peut par exemple faire émerger des aspects globaux qui se noient dans les détails des données brutes. Une autre finalité d'un codage peut être de fournir des propriétés spécifiques au niveau de l'indexation ou/et de la localisation.

La relation forte qui existe entre le type de codage utilisé et le domaine d'application est la cause primaire du nombre important et de la diversité des méthodes et approches qui existent dans ce domaine. Il est impossible et voire inutile de les énumérer toutes. Seules les méthodes les plus connues et qui forment une approche généraliste et donc utilisable dans tous les domaines sont brièvement présentées ici.

3.1.1.1 Transformée de Fourier

Un codage classique dans le traitement de signaux temporels est la transformée de Fourier (TF). Il s'agit de transformer le signal temporel dans le domaine fréquentiel (3.1) et vice-versa (3.2).

$$F(s) = \int_{-\infty}^{+\infty} f(x)\, e^{-isx}\, dx \tag{3.1}$$

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(w)\, e^{iwx}\, dw \tag{3.2}$$

En principe on représente un signal temporel par un spectre de fréquences. Pour chaque fréquence les coefficients expriment l'importance de cette fréquence dans le signal d'origine. A ce stade aucune réduction de données n'est faite, seule la représentation a été changée. Afin de réduire la dimension, une possibilité est de sélectionner seulement les coefficients qui dépassent un certain seuil. Il est possible de reconstituer le signal à partir de ces coefficients sélectionnés en relation avec les fréquences correspondantes, mais avec une certaine perte d'information.

Un avantage de ce codage est que de nombreuses études autour de cette méthode ont été faites, et il existe des algorithmes de conversion performants.

Un désavantage de ce codage est que le choix des coefficients se fait dans la représentation fréquentielle dans laquelle aucun lien avec le domaine temporel n'est visible. L'information de localisation d'événements sur l'axe du temps est perdue.

Une méthode qui se base sur la TF et qui garde une certaine information temporelle est la STFT (Short-Time Fourier Transformation). Elle consiste à balayer le signal avec une fenêtre de taille fixe et d'en extraire les fréquences [1].

3.1.1.2 Le codage en ondelettes

Le codage basé sur des ondelettes est une approche qui conserve l'information temporelle des fréquences. De plus, elle intègre la possibilité de coder aussi bien de petites parties à haute fréquence que de grandes plages avec des comportements quasi statiques [38].

La méthode des ondelettes consiste à représenter un signal par une superposition de séries temporelles appelées "les ondelettes". Chaque ondelette est construite à partir d'une onde de base, en utilisant des compressions linéaires et des translations qui utilisent des paramètres respectifs $a = 2^j$ et $b = 2^j k$, ce qui correspond à des octaves.

Le codage en ondelettes contient normalement des coefficients pour la représentation à haute résolution et en parallèle des coefficients pour une représentation à faible précision.

3.1.1.3 Approximation par parties constantes (PAA)

La méthode d'approximation par parties constantes (angl. Piecewise Aggregat Approximation, PAA) a été introduite indépendamment par deux auteurs [32], [63]. L'approche est intéressante, parce qu'elle est simple, intuitive et compétitive par rapport à d'autres approches plus complexes. Ici la méthode va brièvement être présentée, parce qu'elle représente les idées de bases du codage adapté qui sera proposé au chapitre 3.2.

Afin de réduire le nombre de données, la méthode propose simplement de remplacer des segments du signal par des plages de valeurs constantes. Pour cela, un signal est divisé dans des segments équidistants, qui sont simplement remplacés par leur valeur moyenne locale.

En principe cela ne représente rien d'autre qu'un deuxième échantillonnage en utilisant la moyenne.

Il a été prouvé que, pour deux séquences Q et S, la distance Euclidienne entre leurs codages PAA, représente une borne inférieure de la distance Euclidienne des séquences réelles [32].

Pour cette raison ce codage simple peut être utilisé pour réaliser une indexation de séquences afin d'accélérer les recherches de séquences.

Deux variantes de cette approche existent :

- PLA (angl. Piece-wise Linear Approximation) : contrairement à la méthode PAA qui se sert de la moyenne pour le codage de la sous-séquence, cette approche utilise une régression linéaire. Cela augmente la ressemblance entre le signal original et la reproduction sur base du codage. Mais en même temps cela divise par deux le taux de réduction pour une même taille de sous-séquences car pour la régression linéaire deux paramètres sont nécessaires.
- APCA (angl. Adaptive Piece-wise Constant Approximation) : Cette approche diffère de son prédécesseur par l'utilisation de plages à valeurs moyennes qui ne sont pas équidistantes [33]. L'avantage est que des segments quasi-stationnaires peuvent être représentés par une seule valeur, et que pour des parties hautement dynamiques un plus grand nombre de segments peut être utilisé, ce qui améliore considérablement la qualité d'approximation, sans pour autant diminuer le facteur de réduction.

Ces types de codages sont explicitement dédiés à des fin d'indexation pour accélérer la recherche de séquences dans les bases de données de grandes tailles. Mais toute approche générique est utilisable à d'autres fins.

3.1.2 Indexation de données

Le but primaire d'une indexation peut être expliqué par un exemple. Supposons que vous faites une recherche dans une base de données qui représente des mesures de température de l'air ambiant, que la base soit organisée par journées (avec quelques mesures par journée) et que les mesures ne soient pas prises régulièrement. Il en résulte que ni les instants de mesures ni le nombre de mesures par journée soient fixes. Supposons ensuite, que vous êtes intéressés à connaître les jours auxquels la différence entre la température maximale et minimale dépasse un certain seuil et pour lesquels la température moyenne de la journée soit dans une plage donnée.

En principe cela ne pose pas de problèmes, car toutes les informations existent dans la base de données. Mais le temps qu'il faut pour calculer la différence min-max et la moyenne pour chaque journée, afin de la comparer à la requête, augmente avec la taille de la base de données.

Or, si vous voulez connaître les jours auxquels la valeur minimale dépasse un certain seuil et auxquels la moyenne soit plus grande que x, il faut recommencer l'opération.

Afin d'éviter de devoir refaire ces calculs à chaque fois, il est possible de définir un certain nombre d'indices, comme par exemple les valeurs statistiques de base (moyenne, min, max, variance, etc.), et de

les calculer pour l'ensemble des jours dans la base. Pour chaque nouvelle entrée, ces indices sont recalculés et mis à jour.

Ces indices représentent un codage spécifique pour ces données et la recherche d'informations devient beaucoup plus rapide. Souvent les critères d'une requête ne sont pas aussi explicites que dans l'exemple : vous voulez par exemple trouver les jours auxquels l'évolution de la température est semblable à celle d'une journée x.

Avec la complexité des critères de requêtes, la relation entre critères et indices devient plus complexe et souvent l'utilisateur veut formuler sa requête au niveau des critères. Cela engendre que la définition des indices dépend du domaine d'application et doit répondre à un certain nombre de critères. Les deux critères principaux sont : éviter de détecter sur base de l'index des exemples qui ne répondent pas aux critères initiaux d'une requête (false alarms), et éviter de ne pas détecter des exemples qui répondent aux critères initiaux (false dismissals). Se tromper selon l'un ou l'autre critère n'a pas la même signification selon l'application.

3.1.3 Localisation de séquences

Dans le cas de la localisation, le but est de trouver dans une séquence C les endroits pour lesquels un ensemble de conditions soit vérifié. Reprenons l'exemple précédent de la température. Vous voulez savoir, par exemple, à quelle date d'une année x la température minimale a été atteinte.

En principe, il est possible de parcourir l'ensemble des mesures d'une année, afin de trouver la valeur minimale et de se rappeler les dates pour lesquelles cette valeur a été atteinte.

Une requête plus complexe pourrait être de trouver les instants pour lesquels la température a chuté de x dans un intervalle de temps donné.

Il est encore plus difficile de trouver les endroits pour lesquels on retrouve une évolution similaire à une certaine évolution représentant par exemple deux heures de mesures.

Sans aller dans les détails, un certain nombre de problèmes peut être rencontré : des données bruitées, des échantillonnages différents ou encore des données manquantes.

Dans ce domaine aussi, l'indexation peut avoir des effets bénéfiques pour réduire la durée de telles requêtes. Supposant l'existence d'un index basé sur les valeurs statistiques de base pour la journée, la semaine et le mois, la recherche de la date avec la température minimale de l'année se limite à chercher le mois de l'année en question avec le t_{min} le plus petit (12 possibilités) puis la semaine de ce mois avec cette valeur (4 à 5 possibilités) et finalement la journée qui correspond à cette valeur (7 possibilités). Cela réduit la recherche à un maximum de 24 comparaisons.

3.2 Un codage adapté aux données

Dans cette section un codage, qui prend en compte les spécificités des signaux qui sont analysés et qui s'adapte de ce fait aux données qu'il s'agit de coder, est proposé. Il s'agit d'une approche qui se base sur les méthodes PAA et PLA en ajoutant une composante de forme. Ce travail se retrouve aussi dans les publications suivantes [42], [54].

Le but du codage est la réduction des données, sans pour autant perdre les informations pertinentes du signal et de garder la relation évènement et instant temporel. Ce qui est pertinent dans un signal dépend du traitement et de l'application que va utiliser ce codage. Dans ce contexte le but est de pouvoir faire une analyse du comportement dynamique du processus, qui est représenté par les données, en analysant l'évolution des signaux dans le temps. Le codage utilisé doit donc en tenir compte.

Le principe du codage adapté a déjà été abordé dans le chapitre 1.4 où les idées de base sont illustrées dans la figure 1.10. La partie reliée au codage est reprise par la figure 3.1.

Le codage adapté est composé de deux phases[12] :

1a) La première phase consiste à extraire, d'un échantillon représentatif de signaux qu'il s'agit de coder, les formes primitives c.-à-d. les formes les plus caractéristiques qui composent ces signaux. Un ensemble de formes extraites est appelé un "alphabet". Il est possible de construire différents alphabets qui se distinguent par les paramètres utilisés durant l'extraction.

[12] La numérotation des phases correspond à celle utilisée dans les figures 1.10 et 3.1

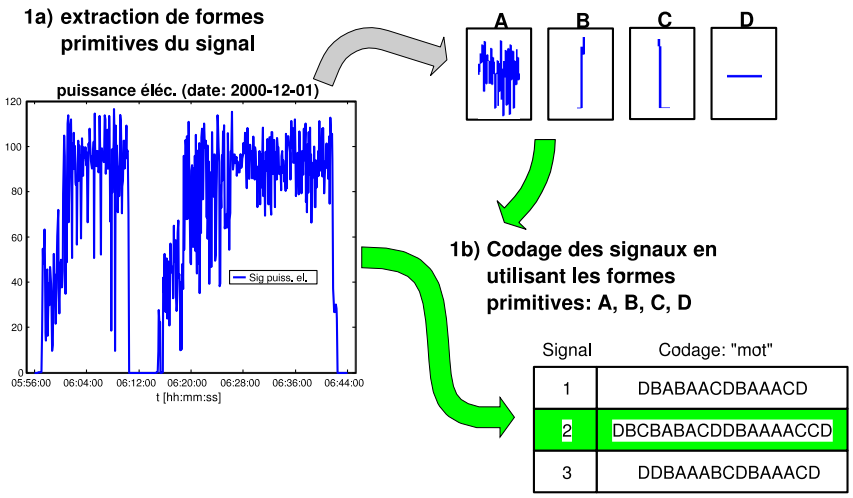


FIG. 3.1 – Première partie de l'analyse évolutive : le codage adapté aux signaux

1b) La deuxième phase consiste à encoder les signaux en se basant sur les alphabets de formes primitives. Les signaux codés sont représentés par une suite de formes primitives qui sera appelée un "mot".

Les sections suivantes vont détailler ces deux phases et expliquer les méthodes et algorithmes utilisés. Afin de pouvoir valider les méthodes, les explications sont accompagnées de résultats basés sur un signal artificiel. Le chapitre clôturera par les résultats obtenus par l'application des méthodes aux signaux réels du four à arc électrique.

3.2.1 Le signal artificiel

Le signal artificiel utilisé pour la validation des méthodes doit d'un côté représenter les spécificités du signal provenant du four en vue de l'application finale, d'un autre côté il doit être entièrement explicable afin de pouvoir discuter les résultats.

Le signal se base sur la fonction de $f(x) = sin(1/x)$ avec $x = \in [1 * 10^{-6}, 0.05]$. Afin de produire les données temporelles, un échantillonnage avec un intervalle fixe $ts = 1.4285 * 10^{-4}$ est utilisé (voir figure 3.2).

Le signal obtenu contient une première partie avec une allure stochastique (bruitée) où la fréquence de $f(x)$ est élevée, ce qui est un résultat de l'utilisation d'un échantillonnage à taille fixe avec une fréquence d'échantillonnage inférieure à la fréquence de cette partie, et une deuxième partie périodique avec fréquence variable pour laquelle la fréquence d'échantillonnage est adaptée. Ces deux parties, avec le passage continu entre les deux, représentent un grand nombre de spécificités qui ont été retrouvées dans les signaux provenant du four.

3.2.2 Extraction des formes primitives

La technique qui est utilisée pour l'extraction des formes primitives qui composent le signal est illustrée dans la figure 3.3. La figure montre un signal uni-variable quelconque, afin de simplifier l'explication de la technique. Notez qu'en principe la méthode est utilisable aussi pour des approches multivariables.

Il s'agit d'une approche non-supervisée pour laquelle il est important que les échantillons du signal qui sont utilisés pour l'extraction de formes primitives, appelées le "corpus d'apprentissage", soient représentatifs et qu'ils comportent les formes les plus caractéristiques.

Afin d'extraire des formes d'une certaine taille c.-à-d. d'un certain nombre d'échantillons, une fenêtre avec une largeur de cette taille balaie de façon aléatoire l'ensemble du corpus d'apprentissage. Les sections du signal ainsi sélectionnées forment un nouveau corpus ou plutôt un autre format du même corpus. Ce nouveau format représente toutes les allures de courbes ayant la taille de la fenêtre.

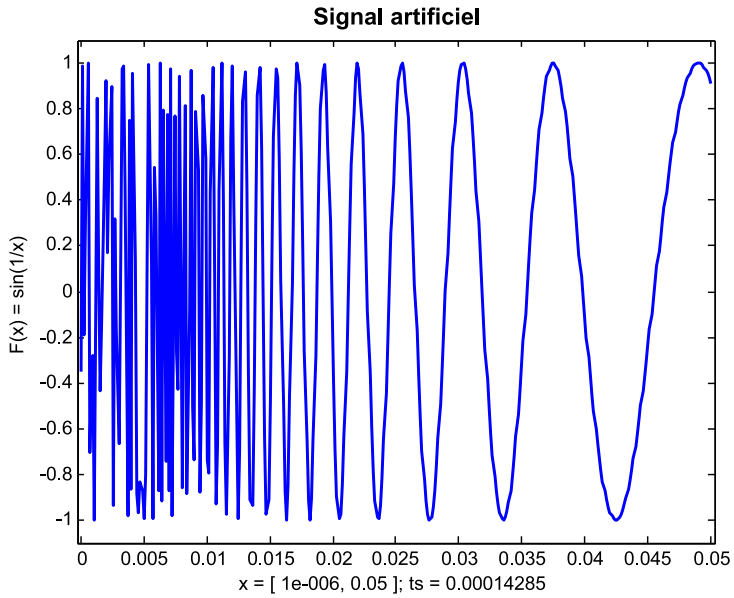


FIG. 3.2 – Le signal artificiel utilisé pour la validation des méthodes.

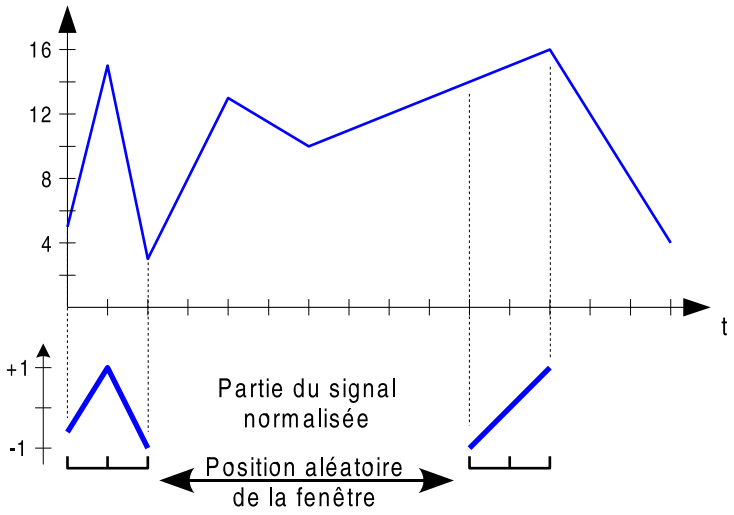


FIG. 3.3 – Technique d'extraction des formes primitives

L'intérêt majeur de cette étude repose sur l'analyse de l'évolution temporelle et par conséquent sur les formes primitives du signal. Afin d'éviter l'influence d'autres aspects comme par exemple la position en abscisse d'une forme ou l'amplitude de la dernière, chaque section sélectionnée par la fenêtre est normalisée de façon centrée réduite (3.3). Cette normalisation enlève la moyenne de la section et réduit sa variance à 1.

$$X^* = \frac{(X - mean(X))}{std(X)} \tag{3.3}$$

Les allures des courbes ainsi normalisées forment le corpus d'entrée d'un classificateur non-supervisé qui va regrouper les formes proches dans des classes. Le prototype de chaque classe est le représentant de la classe et représente en quelque sorte une moyenne des éléments regroupés dans la classe. Dans le contexte de ce projet, le prototype représente une allure moyenne c.-à-d. la forme qui représente au mieux les allures de la classe. Cette forme est définie comme une forme primitive du signal analysé. En spécifiant le nombre de classes à produire par le classificateur, le nombre de formes primitives à extraire du signal est fixé.

En principe tout type de classificateur capable de regrouper de façon non-supervisée des vecteurs de taille fixe peut être utilisé. Dans cette étude la classification est assurée par un réseau de neurones de la famille des cartes auto-organisatrices qui a été introduit par Kohonen en 1978 [36] (SOM). Les raisons principales pour lesquelles la SOM a été retenue comme classificateur sont :

- la facilité d'extension en vue d'une approche multivariable,
- la relation de voisinage entre classes. Les cartes auto-organisatrices assurent une topologie entre les classes. Les classes voisines sur la carte sont en même temps les plus proches en ce qui concerne leurs distances.
- le fait que le projet de recherche industriel CECA spécifie une orientation connexionniste.

La SOM produit, de manière non-supervisée, une cartographie des échantillons présentés en entrée du réseau (voir chap. 2.2.2.5). Plusieurs méthodes existent pour extraire d'une telle carte les formes primitives. La méthode utilisée ici consiste à spécifier une topologie unidimensionnelle de la carte et à utiliser une unité de la carte pour chaque classe à obtenir. Chaque unité de la SOM représente, après l'apprentissage, le prototype d'une classe et donc une forme primitive du corpus d'apprentissage.

Afin d'obtenir des classes bien distinctes tout en gardant la propriété de voisinage, l'apprentissage de la carte doit suivre un cheminement en quatre phases :

1. La carte est initialisée de façon linéaire sur le domaine couvert par le corpus d'apprentissage sélectionné.
2. Afin de positionner la carte au centre du nuage formé par les échantillons du corpus, la première phase de l'apprentissage est réalisée avec un taux d'apprentissage et une relation de voisinage élevés.
3. Durant la deuxième phase d'apprentissage, les unités de la carte s'adaptent à la distribution qui existe dans le corpus d'apprentissage tout en garantissant la relation de voisinage entre les unités. Le taux d'apprentissage et le voisinage sont diminués avec l'avancement de l'apprentissage. A la fin de cette deuxième phase le voisinage se limite aux unités voisines directes.
4. La dernière phase d'apprentissage est une phase de raffinement où les unités de la SOM sont censées se positionner dans les centres respectifs des cluster. Il est donc nécessaire de relâcher les forces d'attraction entre les unités dues au voisinage. Le taux d'apprentissage reste faible durant cette phase.

Plus de détails sur les motivations de ce cheminement à cet endroit n'aideraient pas à la compréhension de la méthode et sont pour cette raison repris dans l'annexe B.

La figure 3.4 montre un résultat de cette méthode en utilisant six unités SOM et une fenêtre de taille sept (sept échantillons du signal). Les différentes classes sont représentées par leur prototype en ligne grasse pointillée. Les membres des classes, c.-à-d. les sections normalisées du signal attribuées à chaque classe, sont représentés en lignes fines colorées.

Par cette méthode il est possible d'extraire, de façon non-supervisée, les formes primitives dont est constitué le signal analysé.

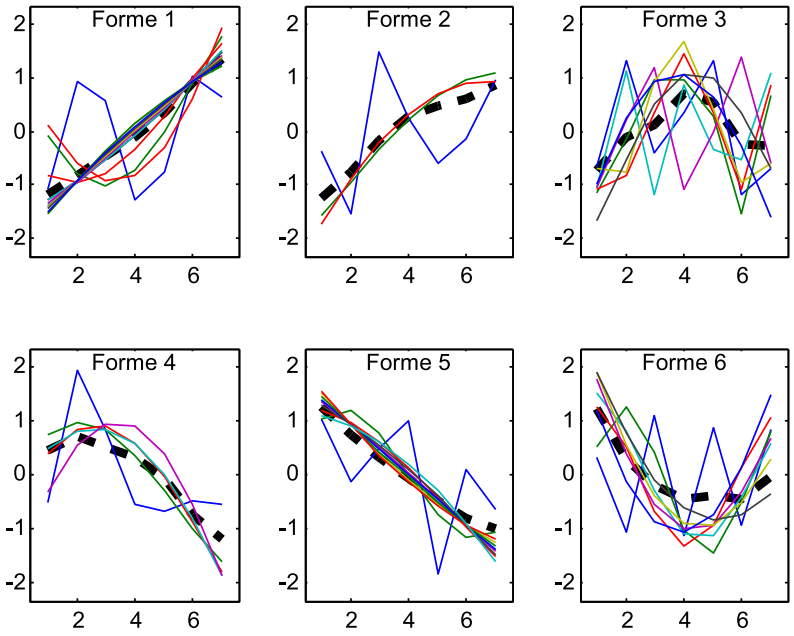


FIG. 3.4 – Prototype et membres de classes

L'ensemble des formes primitives obtenu avec une taille de fenêtre donnée est appelé un "alphabet", à cause de l'utilisation de ces formes pour le codage des signaux. Chaque forme primitive est en conséquence appelée une "lettre".

Un avantage de la normalisation des sections sélectionnées avant le classificateur est le fait que la taille de "l'alphabet" représentant au mieux les formes primitives du signal peut rester petite, car la position et l'étendue de la forme sont enlevées.

Les paramètres principaux de cette méthode sont la taille de la fenêtre utilisée, c'est-à-dire la taille des formes primitives ("lettres") et le nombre de classes à produire par le classificateur, c'est-à-dire le nombre de "lettres", ce qui représente la taille de "l'alphabet".

Il n'est pas trivial de fixer ces paramètres afin d'obtenir un ou plusieurs "alphabets" représentatifs. La tâche peut être comparée à la validation du nombre idéal de classes à utiliser pour une classification, mais ici avec une dimension en plus à considérer pour l'optimisation. Car non seulement le nombre de classes, mais aussi la taille des "lettres" qui représentent au mieux le signal, doivent être fixées. Il est possible de constater que dans l'exemple donné dans la figure 3.4, quelques classes sont relativement compactes (1,4,5), pour d'autres par contre (2,3,6), la variance de formes à l'intérieur de la classe est plus importante. Comme pour toute approche de validation de classification, il s'agit de trouver le plus petit nombre de classes, tout en gardant les classes les plus compactes possibles.

Deux approches de validation ont été analysées : l'une basée sur la variance intra-classe cumulée et l'autre est basée sur l'index de Davies et Bouldin [10]. Les détails des approches sont repris dans l'annexe C.

3.2.3 Encodage basé sur les formes primitives

Ce chapitre reprend la deuxième phase du codage adapté, c.-à-d. l'encodage des signaux (1b fig. 1.10 ou 3.1). Il s'agit d'encoder les signaux en se basant sur les "alphabets", qui ont été produits au chapitre 3.2.2. La méthode utilisée est illustrée par la figure 3.5.

L'objectif est de trouver une suite de "lettres", c.-à-d. de formes primitives, qui représente au mieux le signal analysé. Pour cela, une fenêtre de taille fixe balaie le signal, comme précédemment durant l'extraction des formes primitives, mais cette fois-ci, de manière consécutive, sans recouvrement, afin de produire une suite continue non redondante de sections du signal.

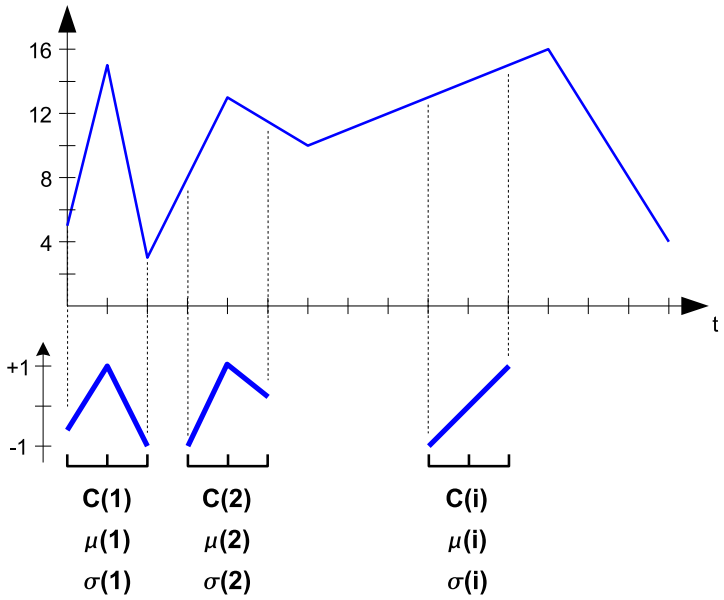


FIG. 3.5 – Méthode d'encodage des signaux

Chaque section est normalisée de façon centrée réduite (voir 3.3), afin de pouvoir comparer sa forme primitive aux "lettres" de "l'alphabet". La "lettre" correspondant le mieux à chaque section est recherchée dans "l'alphabet" sélectionné. Les valeurs de la normalisation, la moyenne et la variance, sont conservées et forment ensemble avec la "lettre" un triplet qui suffit à reproduire une approximation de la section originale du signal.

La recherche de la "lettre" la plus représentative est réalisée par la même carte auto-organisatrice (SOM) qui représente "l'alphabet". La carte est dans cette phase utilisée en tant que simple classificateur. Dans cette phase plus aucun apprentissage n'est réalisé. La SOM attribue chaque section montrée en entrée du réseau à la classe ("lettre") qui lui ressemble le plus.

Finalement, le signal est représenté par une suite de triplets qui constitue un "mot" représentant le codage du signal. Chaque triplet $[C, \mu, \sigma]$ contient un nombre entier, se référant à la classe ("lettre"), et deux valeurs réelles pour la moyenne et la variance de la normalisation locale.

En se basant sur les "alphabets" produits dans le chapitre précédent, ce codage est appliqué au signal artificiel. Le tableau 3.1 montre une partie du "mot" du signal artificiel. La ligne C(i) représente l'identifiant (l'index) de la "lettre" qui correspond au mieux à la partie du signal. La forme primitive correspondante peut être reprise de l'alphabet qui est montré dans la figure 3.4. Les lignes $\mu(i)$ et $\sigma(i)$ représentent les valeurs de la normalisation, la moyenne et la variance de la section du signal.

i	1	2	3	4	$\cdots$	49	50
C(i)	3	2	5	3	$\cdots$	2	5
$\mu(i)$	0.0979	-0.0547	0.2548	0.2348	$\cdots$	0.9783	0.9632
$\sigma(i)$	0.6837	0.6153	0.6561	0.7879	$\cdots$	0.0233	0.0311

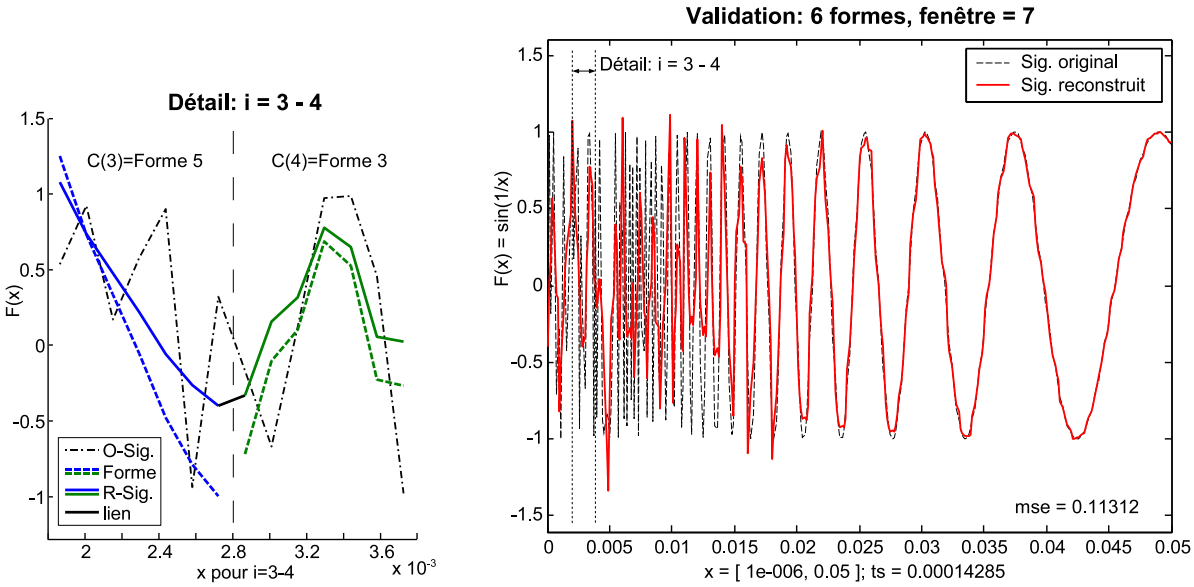
TAB. 3.1 – Exemple de triplet du codage du signal artificiel

Afin de valider la méthode, le signal codé est reconstruit en se basant sur "l'alphabet" utilisé pour le codage et comparé au signal original (voir figure 3.6). La reproduction du signal passe par une concaténation de formes approximées des sections du signal. La forme approximée d'une section est le résultat de la procédure inverse du codage, c.-à-d. la dénormalisation de la forme primitive issue de "l'alphabet" qui a été attribuée à la section en se basant sur les valeurs de normalisation locale (3.4).

$$\vec{x}(i) = (\vec{x}^*_{C(i)} * \sigma(i)) + \mu(i) \tag{3.4}$$

Avec $\vec{x}^*_{C(i)}$ la forme primitive $C(i)$ de l'alphabet.

Par exemple pour $i = 3$ (voir tab. 3.1) la forme primitive numéro "5" de la figure 3.4 est utilisée. Cette forme "5" est dans ce cas réduite en taille par la multiplication avec $\sigma(3) = 0.6561$ et positionnée sur l'axe des y par l'addition de $\mu(3) = 0.2548$. La figure 3.6(a) montre les détails de la reconstruction pour les positions $i = 3$ et $i = 4$ en représentant le signal d'origine, les deux formes primitives les formes dénormalisées (reconstruites) et le lien entre les deux parties reconstruites qui résulte du fait que le codage a été réalisé sans recouvrement.



(a) (b)

FIG. 3.6 – Reconstruction du signal artificiel après le codage : (a) Détail de la reconstruction pour les position $i = 3, i = 4$; (b) Comparaison du signal original avec le signal reconstruit.

Il est clairement visible dans la figure 3.6(b) que la première partie du signal artificiel est reconstruite avec un taux d'erreur plus important que la deuxième partie. Cette constatation s'explique par la propriété de filtrage local de la méthode. En fait, la "lettre", c.-à-d. la forme primitive qu'elle représente, qui est produite durant la première phase, c.-à-d. l'extraction des formes primitives, est soumise à des contraintes représentées par les propriétés de "l'alphabet" :

- la taille fixe des séquences analysées et
- le nombre restreint de "lettres" à produire.

Chaque lettre de "l'alphabet" représente une forme moyenne des sections normalisées qui sont regroupées dans la classe (voir 3.4). Cette forme moyenne représente une forme filtrée (lissée) des sections regroupées dans la classe.

En réduisant la taille des "lettres", la qualité du signal reproduit augmente, mais l'avantage d'un codage compact se perd. Ceci est également valable pour l'augmentation du nombre de lettres de l'alphabet. Ces deux possibilités sont détaillées dans le chapitre 3.2.4 sur l'application de la méthode aux données du four.

En vue d'une analyse du comportement macroscopique du processus sous-jacent, la réduction de la taille l'emporte sur la qualité de la reproductibilité du signal.

3.2.3.1 Possibilité d'extension pour une approche multivariable

Dans ce travail, un seul signal est analysé afin de limiter la complexité en vue de l'analyse des méthodes proposées. Mais toutes les méthodes sont réalisées dans l'optique d'une extension à une approche multivariable. Dans le cas d'un codage multivariable, les deux éléments de normalisation locale du triplet

deviennent des vecteurs. Il reste un nombre entier pour la "lettre" et deux vecteurs de valeurs réelles de taille identique pour les valeurs de normalisation (moyenne et variance) des formes pour chaque variable encodée. Dans ce cas la "lettre" d'un "alphabet" contient un ensemble de formes, une pour chaque variable qui est prise en compte pour le codage. Ces formes sont apprises ensemble afin d'incorporer les relations intra-variables. Si par exemple la variable v_1 a souvent une courbure positive quand la variable v_2 reste constante, cette propriété sera apprise durant la phase d'extraction de formes. Avec le nombre de variables la possibilité de formes multi-dimensionnelles augmente et il faudra prévoir un "alphabet" de plus grande taille. Dans le cas d'une approche multiéchelle plusieurs codages sont réalisés pour un même signal.

3.2.3.2 Comparaison avec les codages PAA et PLA

Il est relativement facile de comparer l'approche proposée aux méthodes PAA ou PLA, pour lesquelles des séquences d'un signal sont approximées par une valeur constante (PAA) ou par une régression linéaire (PLA). La figure 3.7 montre trois différents niveaux de reconstruction du signal d'origine possibles sur base du codage proposé, qui peuvent être attribués aux codages PAA et PLA. Les niveaux sont :

1. si seulement la moyenne $\mu(i)$ du triplet est utilisée le codage PAA est réalisé,
2. si au niveau de la moyenne une régression linéaire de la forme est reconstruite un codage comparable à celui du PLA est réalisé,
3. si la reconstruction complète, en utilisant la forme dénormalisée, est faite le codage adapté est réalisé.

Le codage proposé, qui pourra être appelé PPA (Piecewise Pattern Approximation), ajoute la notion de forme (pattern) et améliore de ce fait l'approximation au signal original. Dans le codage proposé les formes sont issues d'une SOM et donc obtenues de façon non-supervisée en ce basant sur le signal analysé.

La figure 3.7 montre une comparaison des trois méthodes de codage utilisant la même taille de fenêtre à chaque fois. Cela représente en même temps les différents niveaux de reconstruction possible en ce basant sur le codage PPA. La valeur "'mse"' donne l'erreur quadratique moyenne réalisée par la reconstruction pour le signal artificiel (voir l'équation 3.5).

Les méthodes basées sur les ondelettes peuvent aussi être comparées à la méthode présentée ici. La méthode PPA utilise comme fonction de base des formes primitives du signal analysé pour l'approximation des séquences. De par leur nature, ces formes sont probablement mieux adaptées pour approcher l'allure du signal. Mais une réelle comparaison n'a pas été analysée dans le cadre de ce travail.

3.2.4 Application aux données du processus industriel

Pour tout le document, le signal de la puissance électrique est utilisé afin d'illustrer l'application des méthodes. La base de donnée utilisée pour la gestion des données provenant de l'installation contient presque 3.3 millions d'échantillons de mesures pour cette variable (tab. A.1). Cela représente en tout 2566 cycles de productions. Des analyses préliminaires, réalisées dans le cadre du projet CECA, qui n'ont pas été reprises dans ce document, montrent que 2471 de ces cycles sont réellement utilisables [45], [16].

En premier lieu l'extraction de formes qui composent ce signal de la puissance électrique est illustrée, ce qui va produire divers "alphabets" de différentes échelles qui vont être utilisés pour le codage et les phases suivantes de l'analyse globale. Dans une deuxième partie les résultats du codage seront analysés.

3.2.4.1 Les formes qui composent le signal de la puissance électrique

Dans ce contexte il est important de préciser que l'approche d'apprentissage par méthode batch ne doit pas utiliser tout le corpus à cause de la validation du résultat qui doit être réalisée sur un corpus non utilisé pour l'apprentissage. C'est pourquoi itérativement différentes parties du corpus ont été utilisées tout au long de l'apprentissage de la carte SOM. En tout 50% du corpus ont été utilisés et la sélection des parties s'est faite de façon aléatoire.

Concernant la production des alphabets, quelques réflexions de principe peuvent être faites concernant les deux paramètres principaux qui définissent chaque "alphabet". Il est question de la taille des formes (taille de fenêtre) et du nombre de formes c.-à-d. de la taille de "l'alphabet".

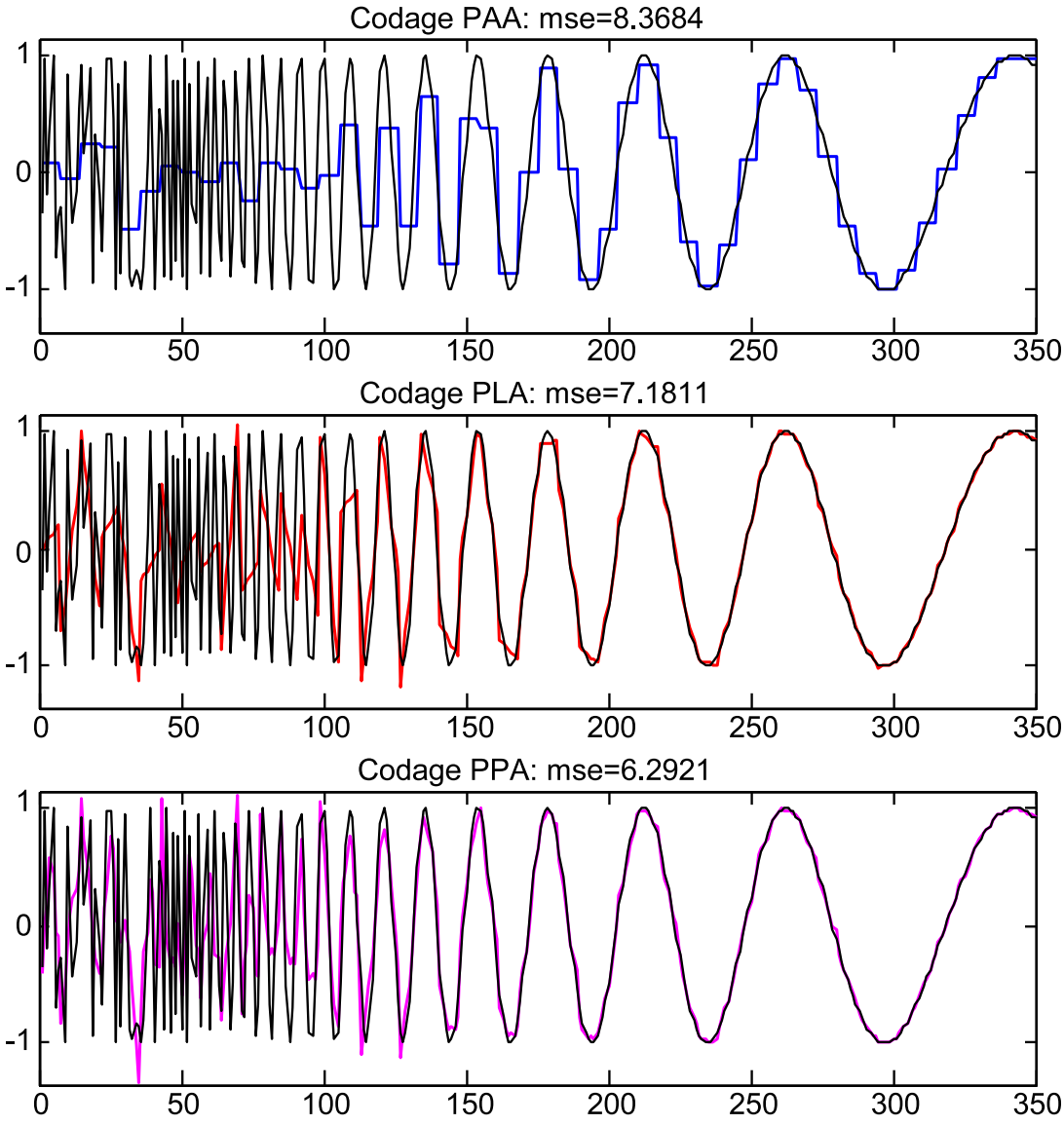


FIG. 3.7 – Niveau de reconstruction basé sur le codage

Le tableau 3.2 montre un aperçu de la mémoire utilisée après codage pour un cycle donné, qui en forme originale nécessite 6296 [Bytes]. Deux tailles de fenêtre sont analysées pour chaque fois quatre tailles d'alphabet.

	taille alpha	mémoire pour l'alphabet [Bytes]	mémoire pour le codage [Bytes]	mémoire totale d'un cycle [Bytes]
taille fenêtre 60	4	1920	312	2232
	9	4320	312	4632
	16	7680	312	8002
	25	12000	312	12312
taille fenêtre 12	4	384	1560	1944
	9	864	1560	2424
	16	1536	1560	3096
	25	2400	1560	3960

TAB. 3.2 – Réduction de mémoire en relation avec les paramètres de "l'alphabet"

Notez que la mémoire nécessaire pour l'alphabet est utilisée une seule fois pour tous les signaux codés, et la mémoire utilisée pour le codage est utilisée pour chaque cycle qui sera codé. La mémoire appelée totale dans le tableau est la somme de l'alphabet et du codage. Puisque l'alphabet est réutilisé pour chaque codage, sa mémoire peut être ignorée dans le cas d'un grand nombre de cycles codés.

Si le but est la réduction de taille pour un grand nombre de données, il est intéressant de travailler avec des fenêtres de grande taille.

Durant les analyses un grand nombre d'alphabets a été produit, et en partie analysé et vérifié par des méthodes de validations (Annexe C). Pour l'illustration des résultats qui sont obtenus par les méthodes, il est plutôt gênant de prendre en compte un nombre trop important de différentes réalisations. Dans la suite du document seul cinq alphabets différents vont être utilisés, afin de limiter la complexité des représentations. Le plus souvent, les illustrations se basent seulement sur un ou deux de ces alphabets.

Le tableau 3.3 montre les paramètres principaux pour les cinq "alphabets" utilisés. A0 a été ajouté aux alphabets sélectionnés dans l'annexe C pour des raisons de représentation car seulement 4 formes primitives de faible taille sont utilisées. Cependant cet alphabet n'est pas optimal selon les critères de validation.

Nom	taille fenêtre	taille alpha
A0	12	4
A1	6	12
A2	24	5
A3	42	6
A4	60	6

TAB. 3.3 – "Alphabets" utilisés avec leurs paramètres

La figure 3.8 montre les 12 formes de "l'alphabet" A1 et les six formes de A4. La représentation est un peu trompeuse, puisque une carte SOM unidimensionnelle est utilisée, et que de ce fait il faudrait représenter les formes en ligne l'une après l'autre, afin de mieux voir la relation qui existe entre elles.

Le but des alphabets est d'être utilisés pour le codage. Il est donc plus intéressant de discuter de la qualité des alphabets après l'application du codage.

3.2.4.2 Application du codage aux données du four

Pour l'analyse le corpus complet des 2471 cycles a été codé par la méthode présentée. Représenter le codage brut n'a aucune valeur et comme pour tout codage, il est difficile pour l'utilisateur d'en tirer des informations. C'est pourquoi quelques méthodes ont été développées.

Une première approche est d'analyser, pour un ou plusieurs cycles sélectionnés aléatoirement, de quelle manière un "alphabet" ou plutôt les formes d'un "alphabet" sont utilisées. La figure 3.9 montre quelques

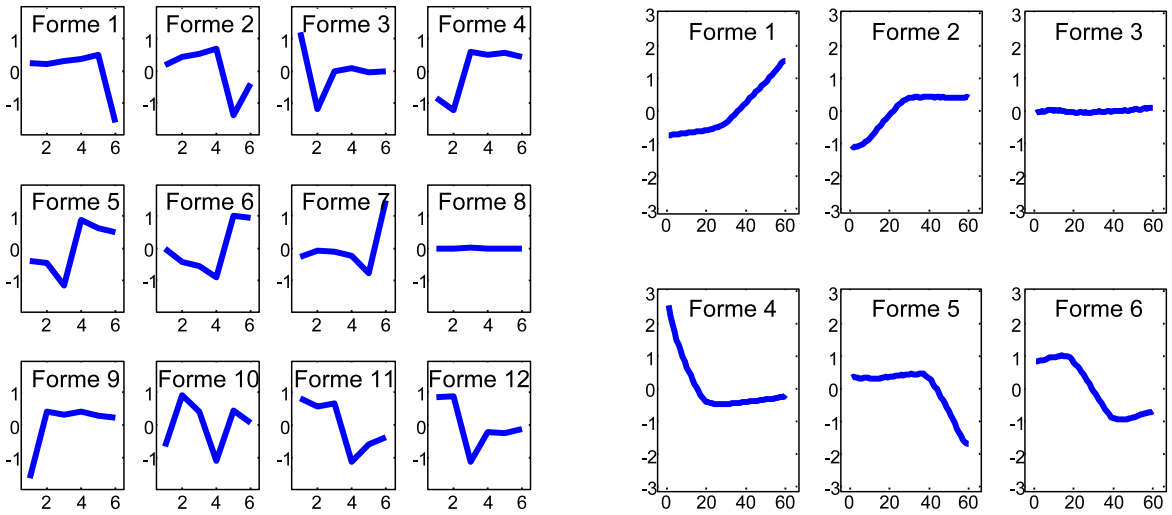


FIG. 3.8 – Exemples des alphabets utilisés : (a) les douze formes de l'alphabet A1 ; (b) les six formes de l'alphabet A4

graphiques pour un cycle en utilisant "l'alphabet" A0

La distribution de formes utilisées montre si oui ou non toutes les formes sont utilisées de façon équivalente ou si éventuellement il existe des formes qui sont peu utilisées pour le codage. Puisque un alphabet représente les prototypes d'une classification non-supervisée, cela donne une information sur le fait s'il existe des classes d'évènements rares.

Le diagramme représentant la forme prototype ainsi que les formes du signal attribuées donne une information visuelle sur la compacité des classes. De plus, cela montre le volume de filtrage qui est réalisé par l'alphabet. Dans l'exemple (fig.3.9) où l'alphabet "A0" est utilisé pour simplifier l'illustration, la variance intra-classe semble plus importante que la variance inter-classes. Cela montre que cet alphabet n'est probablement pas un premier choix pour une application concrète.

Ces deux informations donnent une impression sur la qualité de l'alphabet. D'autres méthodes automatiques sont discutées dans l'annexe C qui traite la validation d'alphabets.

Les distributions de la moyenne et de la variance donnent une impression sur les déformations des formes primitives qui sont nécessaires afin de représenter au mieux le signal d'origine. Pour le cycle analysé la moyenne se trouve sur deux plages, une relativement petite et une autour de 90. La variance est davantage distribuée avec une valeur maximale près de 50. Que la variance ne représente que environ la moitié de la moyenne résulte du fait que le signal analysé est strictement positif.

Une deuxième approche est la reconstruction du signal original sur base du codage et le calcul de l'erreur entre les deux signaux. La figure 3.10 montre la reconstruction pour le même cycle qui a été analysé auparavant (fig 3.9).

La mesure d'erreur qui est utilisée, la MSE (3.5) (angl. mean square error), représente la moyenne de l'erreur quadratique.

$$MSE = \frac{1}{N}\sum(y_i - \hat{y}_i)^2 \tag{3.5}$$

Avec $\hat{y}_i$ la valeur de la reconstruction pour la position i et N le nombre de points.

En regardant la reconstruction réalisée en se basant sur le codage adapté, notez que les discontinuités et l'allure globale sont relativement bien reproduites.

Afin de comparer le résultat, la moyenne flottante (MF) (une méthode connue dans le domaine du filtrage), est utilisée. Le nombre de points utilisés pour la moyenne flottante correspond à la taille de la fenêtre qui est utilisée pour le codage adapté. L'erreur de cette approche est plus petite, ce qui veut dire

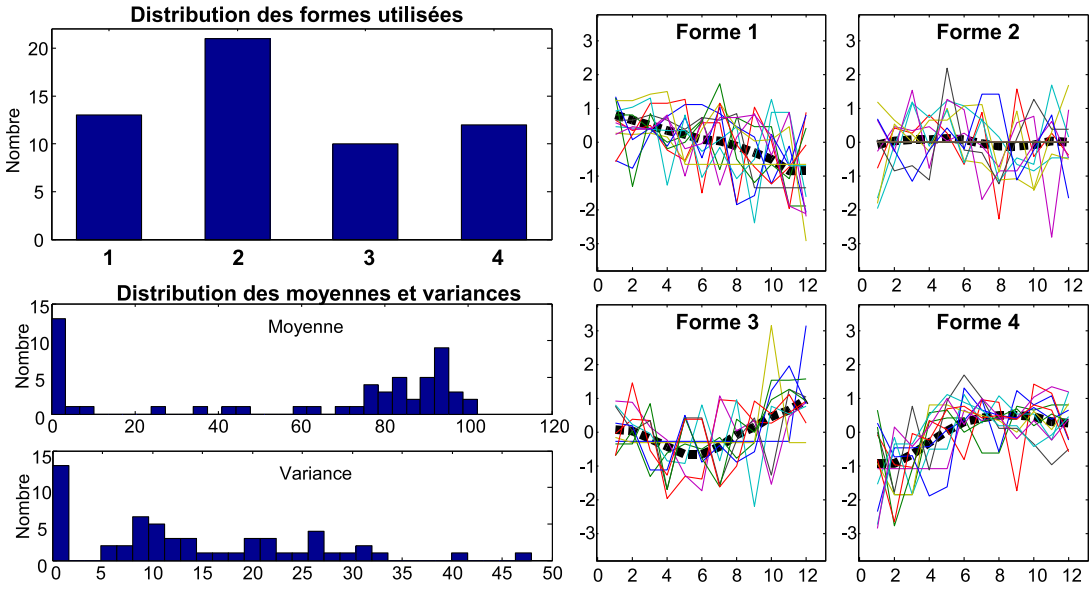


FIG. 3.9 – Analyse comment sont utilisées les formes d'un alphabet (A0)

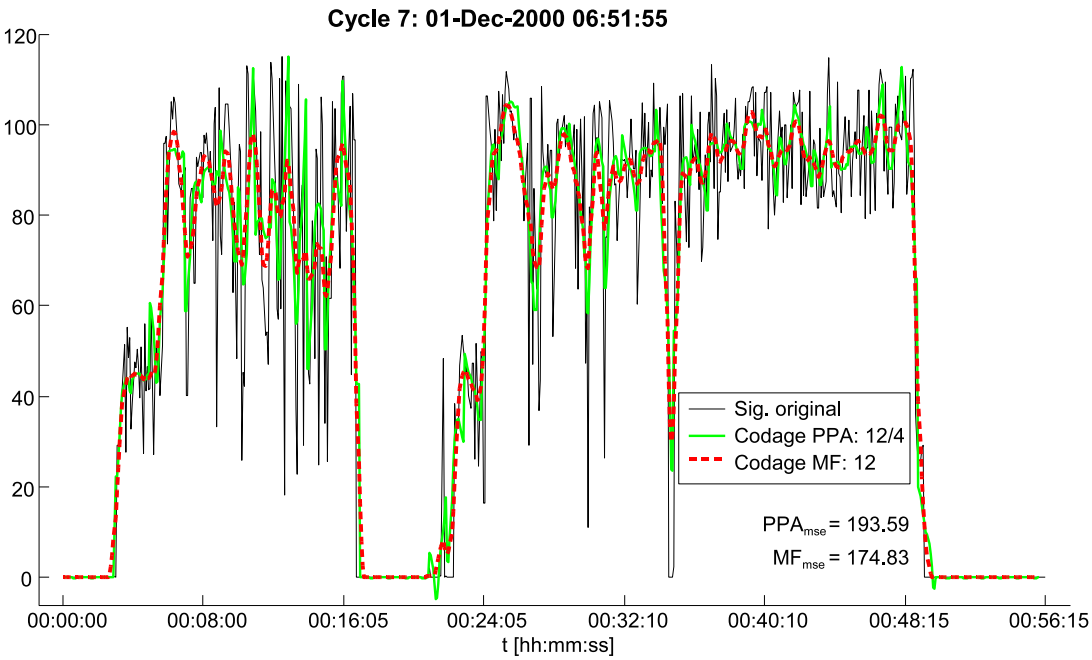


FIG. 3.10 – Reconstruction du signal et comparaison avec une approche de moyenne flottante.

que l'approximation qui est faite par le filtrage de la moyenne flottante est plus proche du signal original que celle basée sur le codage PPA.

N'oublions pas qu'un alphabet avec seulement quatre formes est utilisé, et que contrairement à l'approche de la moyenne flottante, les formes sont consécutives sans recouvrement. Analysons le comportement de l'erreur en augmentant le nombre de formes pour une taille de fenêtre fixe. La figure 3.11 montre ce comportement pour les alphabets de référence.

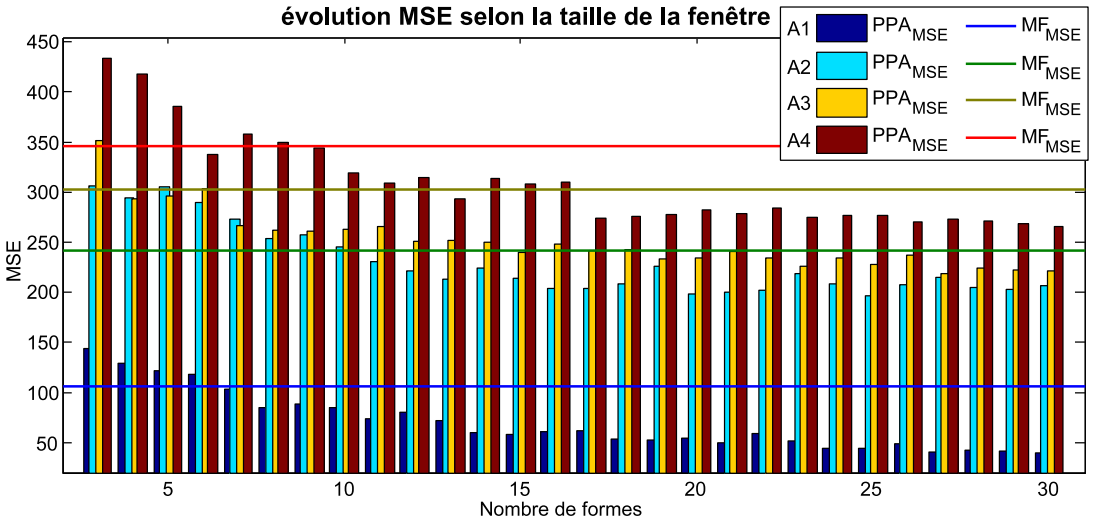


FIG. 3.11 – Analyse du comportement de l'erreur avec la taille de l'alphabet.

Cela montre qu'en augmentant le nombre de formes la reconstruction du signal s'améliore, et que la qualité dépasse celle de la moyenne flottante. L'allure du comportement de l'erreur montre aussi que le gain de qualité de reconstruction diminue avec le nombre de formes. Il ne sert donc à rien d'utiliser un très grand nombre de formes.

La figure 3.12 montre une partie de la reconstruction du cycle Nr. 7 (voir fig.3.10) avec des détails en utilisant deux des alphabets de référence (voir tab.3.3).

Il est clairement visible que le codage basé sur l'alphabet A2, avec une taille de fenêtre plus grande, réalise un filtrage plus important (qualité de reproduction plus faible) que le codage basé sur A1. Par contre les discontinuités sont relativement bien reconstruites dans les deux cas.

Il ne faut pas non plus oublier que le but principal de notre approche n'est pas de coder au mieux un signal, mais de réduire la taille des données qui sont à manipuler pour les phases suivantes de l'analyse sans perdre les informations pertinentes tels que les instants d'enclenchement du four par exemple. Le but principal de cette étude est de réaliser une catégorisation des cycles de production selon leur évolution temporelle. Regardé de ce point de vue, il est intéressant d'analyser le codage avec le résultat de la classification non-supervisée qui se base sur ce codage adapté. Cependant cette analyse aurait dépassé le cadre de cette thèse.

3.3 Classification de séries temporelles de tailles variables

La deuxième et dernière partie de l'analyse évolutive (voir 2. fig 1.10) consiste à regrouper des évolutions du processus qui se ressemblent dans des classes d'évolutions [55]. Dans la figure 3.13 cette partie a été isolée de l'approche globale afin de focaliser le sujet de ce chapitre.

Il s'agit en fait d'une classification non-supervisée de séries temporelles selon leur évolution. Après le codage, l'évolution du processus est représentée par les "mots" c.-à-d. des séquences de triplets. Ce codage est l'information en entrée pour la réalisation du classificateur.

La classification non-supervisée est une tâche classique du domaine de l'analyse de données et des extractions de connaissances. Il existe un grand nombre de méthodes pour la réalisation de cette tâche

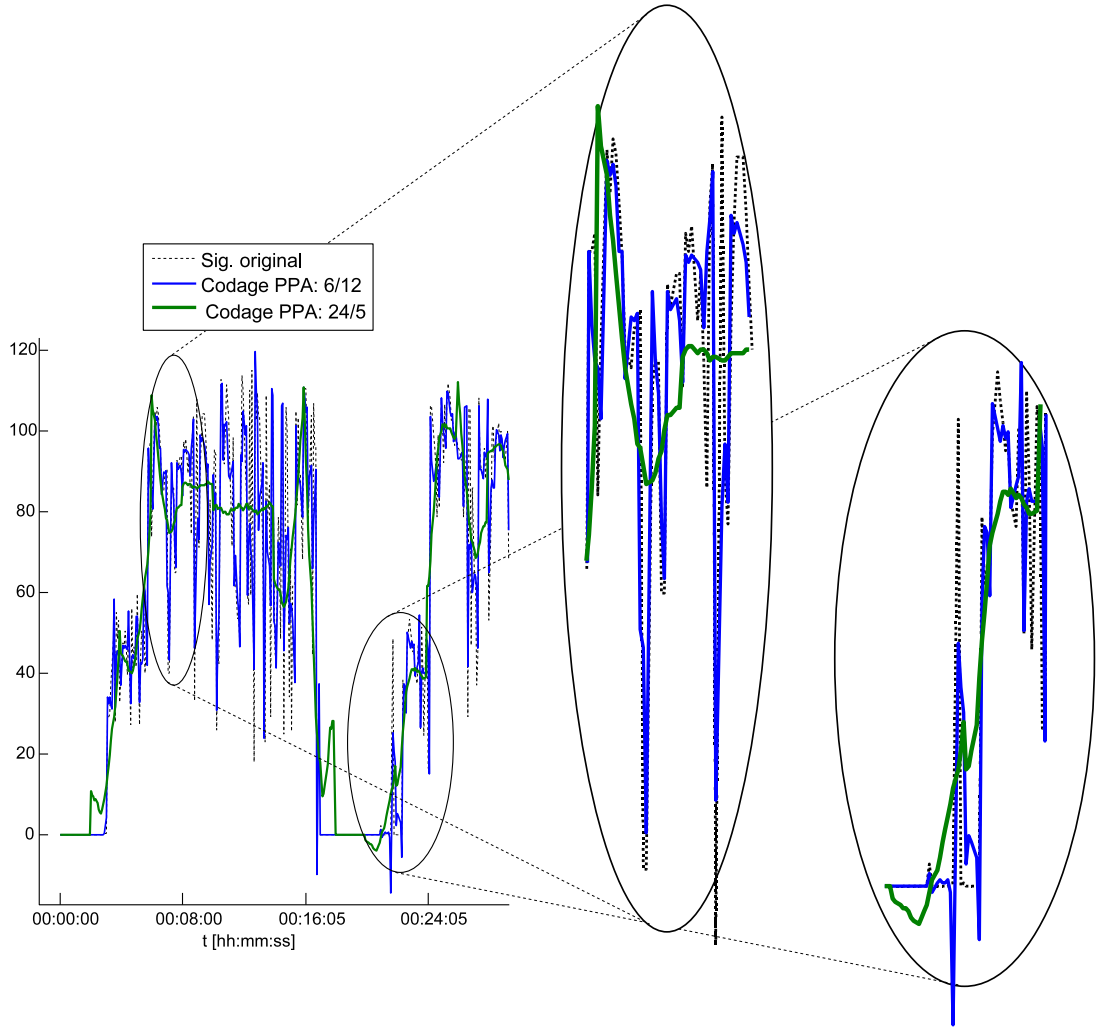


FIG. 3.12 – Partie du signal exemple reconstruit par A1 et A2

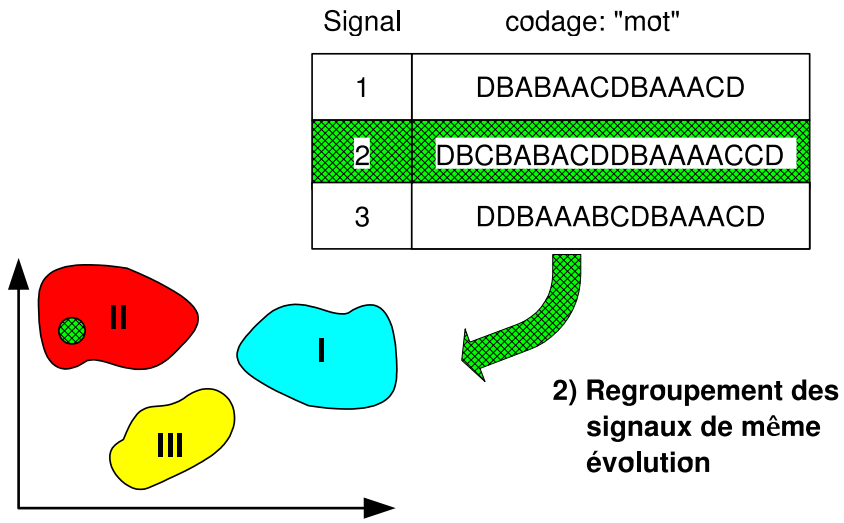


FIG. 3.13 – Deuxième partie de l'analyse évolutive : regroupement des charges selon leur évolution.

dans des cas typiques. Un aperçu d'une partie de ces méthodes a été donné dans le chapitre 2.2.2.

Dans le contexte du projet industriel, les séquences qu'il s'agit de classifier de façon non-supervisée, représentent des évolutions temporelles des cycles de production (voir chap. 1.2). Influencés par divers paramètres, il a été constaté que les cycles montrent une variation importante concernant leur durée. Une autre constatation est qu'un certain nombre d'événements revient dans chaque cycle de production. Mais le moment de leur apparition varie dans le temps à l'échelle du cycle. Les analyses qui soulignent cette constatation ne sont pas reprises dans ce document, mais le lecteur intéressé est renvoyé au rapport final du projet CECA [2].

Cette particularité se retrouve dans d'autres domaines comme par exemple les signaux cardio-vasculaires ou la consommation énergétique urbaine pour n'en citer que deux. Deux problématiques différencient l'application du cas typique :

1. La définition de similarité entre deux séquences temporelles représentant des évolutions n'est pas triviale. Ceci mène vers une analyse de la similarité et du calcul d'une mesure de dissimilarité à utiliser afin de pouvoir l'adapter au domaine analysé.
2. Les cycles qu'il s'agit de regrouper peuvent avoir des variations importantes dans leur durée. Ce qui veut dire que les séquences qui sont à classifier n'ont pas la même taille et qu'il faut en tenir compte.

Ces particularités sont détaillées dans ce chapitre et les outils pour les aborder vont être décrits.

3.3.1 Particularités et problématiques

Dans ce sous-chapitre les deux particularités évoquées auparavant, à savoir la similarité avec la définition d'une mesure de dissimilarité et les séquences de taille variable qu'il s'agit de classifier de façon non-supervisée, sont expliquées. La problématique qui en découle est discutée.

3.3.1.1 La similarité, un sujet non trivial

Quelle est la définition de la similarité ? Les dictionnaires utilisent communément les termes suivants avec des définitions relativement semblables :

Similaire (adj.) Se dit de choses qui peuvent, d'une certaine façon, être assimilées les unes aux autres. Syn. : analogue, semblable.

Similitude (n.f.) Ressemblance plus ou moins parfaite entre deux ou plusieurs choses. Syn. : analogie, identité, affinité.

Afin d'entamer le sujet, la figure 3.14 montre deux images qui font apparaître toute la problématique.

Ces deux images ont une forte ressemblance. Différentes propriétés sont très proches comme par exemple la palette des couleurs, les grandes lignes des formes représentées telles que les contours, les positions des yeux, du nez et de la bouche. Cependant il s'agit bien de deux sujets complètement différents.

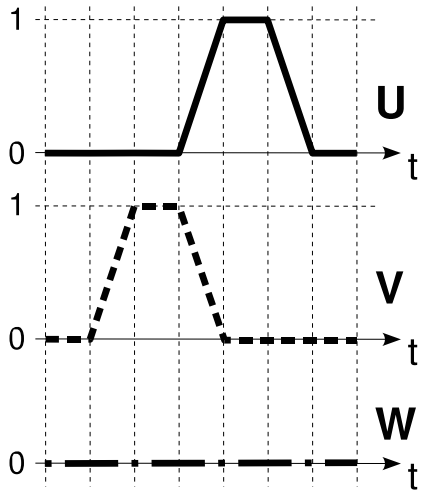
La similarité est difficile à définir, mais l'humain la réalise (se rend compte) quand il la voit [30]. En fait la réelle nature de la similarité est une question plutôt philosophique.

Dans la suite de cette analyse une approche plus pragmatique et mathématique sera entamée, sans pour autant perdre de vue l'aspect subjectif du sujet. Il est important de pouvoir adapter les méthodes à la problématique concrète du sujet analysé en se référant à des avis d'experts.

Définissons pour commencer trois séquences temporelles $U(t), V(t), W(t)$ relativement simples (voir fig. 3.15(a)). Les séquences ont toutes la même durée et elles sont échantillonnées avec le même intervalle d'échantillonnage fixe. Les deux premières séquences, $U(t)$ et $V(t)$, représentent un signal avec un créneau d'amplitude unitaire d'une durée de deux pas de temps, mais commençant à différents instants de la séquence. La dernière séquence, $W(t)$, reste nulle sur toute la durée.

Afin de pouvoir déduire, de façon automatique, lesquelles de ces trois séquences se ressemblent le plus, il est nécessaire d'être en mesure de pouvoir calculer une valeur pour la dissimilarité entre les séquences. La métrique qui est utilisée le plus souvent dans le contexte d'un calcul de dissimilarité se base sur la distance euclidienne. La figure 3.15 montre le résultat obtenu avec cette mesure sous forme de tableau (fig. 3.15(b)) et sous forme de regroupement hiérarchique (fig. 3.15(c)).

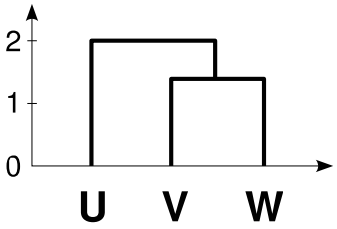
FIG. 3.14 – Deux images similaires ou non ?



(a)

	Distance euclidienne
$D(U,V)$	2
$D(U,W)$	$\sqrt{2}$
$D(V,W)$	$\sqrt{2}$

(b)



(c)

FIG. 3.15 – Similarité entre trois séquences simples :
(a) Les séquences : U, V, W ;
(b) Les distances euclidiennes entre chaque paire ;
(c) Regroupement par arbre hiérarchique.

Dans cet exemple simple qui utilise la distance euclidienne comme mesure de dissimilarité, le signal $W(t)$ est plus proche de $U(t)$ et de $V(t)$ que ne le sont ces derniers entre eux. Ce résultat peut aussi-bien être correct que faux. Sans aucune information supplémentaire sur l'origine des signaux ou sur le genre de similarité qui est recherché, ce résultat ne peut être discuté davantage, ce qui montre que le contexte joue un rôle fondamental.

Supposons le contexte suivant : les signaux représentent le signal d'un détecteur de présence infrarouge dans un bureau. L'échantillonnage est fait sur des plages de 3 heures commençant à minuit et représente le maximum du signal durant la période d'échantillonnage. Dans ce cas, $U(t)$ représente une journée durant laquelle une personne était présente entre 12 :00 et 18 :00, $V(t)$ une journée avec une présence entre 6 :00 et 12 :00 et $W(t)$ une journée ou personne n'était dans le bureau. Si maintenant le critère de l'analyse est l'occupation du bureau, le résultat obtenu avec la distance euclidienne est discutable, car il faudrait plutôt regrouper les signaux $U(t)$ et $V(t)$ qui ont dans ce cas une évolution plus proche, puisque durant les deux journées il y avait présence dans le bureau, même si c'était à différents moment de la journée.

Ces deux exemples, (fig. 3.14) et (fig. 3.15), montrent que le sujet de la similarité est fortement relié au domaine analysé et que la définition d'une mesure de dissimilarité adaptée à cette analyse est un facteur clef.

3.3.1.2 Taille variable des séquences

Dans l'exemple précédent (fig. 3.15) les séquences avaient toutes la même taille. Dans le domaine de l'analyse de séries temporelles cette propriété n'est pas toujours donnée. La figure 3.16 représente un exemple simple qui illustre la problématique. Elle montre la séquence $U(t)$, déjà introduite précédemment, avec une légère variation $U^{'}(t)$ qui ne se différencie de la première séquence que par un ajout de deux échantillons à valeur nulle.

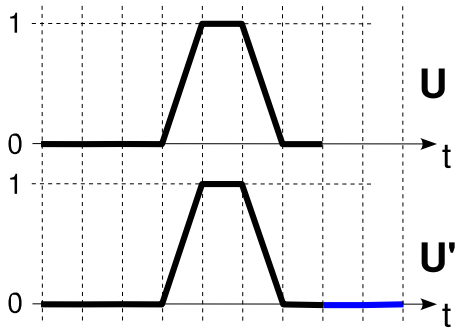


FIG. 3.16 – Similarité entre deux séquences simples de taille différente

Le calcul de ressemblance basé sur la distance euclidienne, comme utilisé précédemment, est impossible.

Une première possibilité pragmatique serait de faire une normalisation sur l'axe de temps afin d'uniformiser les cycles sur une même taille. Deux remarques peuvent être faites à ce sujet :

- la normalisation est une opération linéaire de compression sur l'axe de temps. Le résultat serait un décalage et une déformation du créneau dans le temps avec la problématique discutée auparavant.
- la normalisation sur l'axe de temps engendrerait un calcul d'interpolation afin de garantir un échantillonnage comparable. Une interpolation entraînerait une déformation du créneau.

3.3.2 Mesures de similarité

Dans le chapitre 3.3.1 la difficulté de définir la similarité a été discutée brièvement. La base de toute discussion sur le sujet est la définition d'une mesure de la similarité. Une telle mesure est directement reliée à la discussion des propriétés des mesures de distances. Ce chapitre traite du sujet de la mesure de

similarité respectivement de dissimilarité dans le contexte de trouver un moyen de regrouper les séquences temporelles selon leur évolution.

Commençons par la définition de la mesure de distance (dissimilitude) telle qu'elle est comprise ici :

Soit O_1 et O_2 deux objets de l'univers des objets possibles. La distance entre les deux est donnée par : $D(O_1, O_2)$.

Pour qu'une telle distance soit utilisable pour la comparaison, elle devrait avoir les propriétés suivantes : elle doit garantir la symétrie (3.6), elle doit garantir la similarité avec elle-même (3.7), elle doit être définie positive (3.8) et elle doit vérifier l'inégalité triangulaire (3.9).

$$\begin{aligned} D(A,B) &= D(B,A) && (3.6)\\ D(A,A) &= 0 && (3.7)\\ D(A,B) &> 0 \qquad \forall A \neq B && (3.8)\\ D(A,B) &\leq D(A,C) + D(B,C) && (3.9) \end{aligned}$$

Cependant parfois la propriété de l'inégalité triangulaire (3.9) contredit l'intuition humaine. Keogh dans [30] cite l'exemple suivant : le cheval (CH) et l'homme (H) sont très différents, mais ils partagent des caractéristiques avec le centaure (CE). Avec une certaine mesure de distance appropriée à ce problème et qui vérifie l'inégalité triangulaire, nous devrions obtenir la relation suivante.

$$D(CH, H) \leq D(CH, CE) + D(CE, H)$$

Ce qui aurait pour signification que le cheval et l'homme ne sont pas tellement différents. De ce fait, l'inégalité triangulaire est une propriété discutable dans le contexte de la similarité.

La mesure la plus utilisée dans le domaine d'analyses de séries temporelles est la distance euclidienne. Elle est un cas particulier de la métrique de Minkowski (3.10) que l'on obtient pour $p = 2$.

$$D(A,B) = \sqrt[p]{\sum_{i=1}^{N} |A_i - B_i|^p} \qquad (3.10)$$

Analysons la distance euclidienne dans le contexte de séries temporelles. Les deux séquences A et B sont, dans ce cas, des séries avec N échantillons. La distance euclidienne $D_{eucl}(A,B)$ représente en quelque sorte le cumul des distances locales $(A_i - B_i)^2$ pour tous les échantillons $i = 1 \mapsto N$. Cela réalise un alignement linéaire dans le temps (voir 3.17(a))

Cet alignement figé dans le temps est une contrainte forte et empêche une souplesse qui est nécessaire pour l'analyse de la similarité de séries temporelles. Reprenons les exemples $U(t)$, $V(t)$ et $W(t)$ (fig. 3.15(a)) et analysons-les plus en détail. Entre $U(t) - W(t)$ et respectivement $V(t) - W(t)$ le calcul de distance se limite à une différence en deux endroits sur la séquence, avec chacune une distance locale de valeur 1 à l'endroit du créneau. Pour la distance $U(t) - V(t)$ (fig. 3.17(a)), les deux créneaux se succèdent et il existe quatre endroits pour lesquels la distance locale a la valeur 1.

Afin de garantir une approche de similarité pour les évolutions temporelles, plus de souplesse pour l'alignement sur l'axe des temps serait préférable. La figure 3.17(b) montre une telle approche avec une compression de U (en couleur bleue) avant le créneau, et un étirement de U (en couleur verte) derrière le créneau, afin d'aligner les créneaux sur l'axe des temps (en couleur rouge). Dépendant du coût d'une telle transformation non-linéaire sur l'axe des temps, les séquences $U(t)$ et $V(t)$ se ressemblent plus ou moins. Pour une telle méthode une comparaison de séries de taille différente tel que U et U' ne représenterait pas d'obstacle.

Maintenant, comment et à quel endroit réaliser ces transformations non-linéaires, n'est pas une tâche triviale. Une méthode qui autorise de telles manipulations est connue sous le nom anglais de "Dynamic Time Warping" (DTW) et réalise un cheminement temporel dynamique tout en calculant une valeur de dissimilarité entre deux séquences. Dans la suite de ce chapitre les bases de cette mesure vont être développées. Elles sont suivies de quelques adaptations et des méthodes de calculs qui sont utilisées. Toute cette partie est illustrée par des exemples basés sur des séquences artificielles afin de simplifier la compréhension.

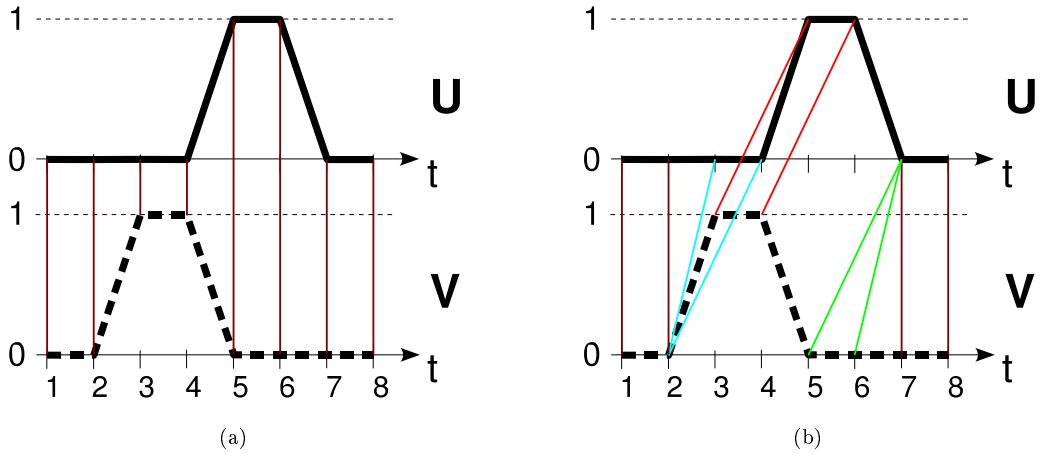


FIG. 3.17 – (a) Alignement linéaire sur l'axe de temps ;
(b) Alignement non-linéaire avec possibilité de compression ou extension sur l'axe de temps

3.3.2.1 Les bases de la distance DTW

Dans cette section les bases de la méthode du "Dynamic Time Warping" (DTW) sont détaillées. Cette méthode est issue du domaine de la programmation dynamique formulé par Bellman en 1957 [3]. Surtout dans le domaine de la parole la programmation dynamique a beaucoup été utilisée [4] ; [59] ; [52]. Le terme de "Dynamic Time Warping" vient probablement aussi du domaine de la parole avec Myers et Rabiner en 1981 [48]. La méthode DTW est utilisée dans diverses approches [61] et a été combinée avec d'autres méthodes comme les modèles de Markov cachés[57] ou les Ondelettes[38]. Plus tard cette méthode est aussi appliquée dans des contextes plus génériques de l'analyse de séries temporelles avec Berndt, Clifford en 1994 [5] et d'autres [62] ; [35] ; [35].

Cette section commence avec la définition de la méthode DTW, les propriétés et l'algorithme dans sa forme récursive. Cette première partie est suivie d'une explication détaillée sur base d'un exemple simple. Commençons par la définition mathématique de la méthode DTW. Dans la suite du document la valeur de dissimilarité obtenue par la méthode DTW sera appelée distance DTW, même si cette mesure ne garantit pas l'inégalité triangulaire qui est spécifiée dans la définition d'une distance.

Soit $S(t)$ une série temporelle multivariée de taille n définit par l'équation (3.11).

$$S = \{\vec{s}(t) | 1 \leq t \leq n\} \tag{3.11}$$

Avec $\vec{s}(i)$ le vecteur de caractéristiques locales à l'instant $t = i$.

Deux séries de ce type, S_1 et S_2 de tailles n_1 et n_2 quelconques, peuvent être comparées avec la méthode DTW si :

- leurs vecteurs de caractéristiques $\vec{s_1}(t)$ et $\vec{s_2}(t)$ ont le même nombre et les mêmes types de composantes,
- il existe une mesure de dissimilarité entre les vecteurs de caractéristiques (3.13).

La distance DTW entre ces deux séries est donnée par $D_{dtw}(S_1(n_1), S_2(n_2))$ en se référant à la définition (3.12) qui est une formulation récursive.

$$D_{dtw}(S_1(i), S_2(j)) = d(i,j) + \min \left\{ \begin{array}{l} D_{dtw}(S_1(i-1), S_2(j)) \\ D_{dtw}(S_1(i), S_2(j-1)) \\ D_{dtw}(S_1(i-1), S_2(j-1)) \end{array} \right\} \tag{3.12}$$

Avec la distance locale entre les vecteurs de caractéristiques :

$$d(i,j) = d(\vec{s_1}(i), \vec{s_2}(j)) \tag{3.13}$$

et avec les conditions de bord (3.14) et le critère d'arrêt de la récursion (3.15).

$$D_{dtw}(\emptyset, S_2(j)) = D_{dtw}(S_1(i), \emptyset) = \infty \quad (3.14)$$

$$D_{dtw}(\emptyset, \emptyset) = 0 \quad (3.15)$$

Afin d'aborder la distance DTW de façon plus intuitive et visuelle, un exemple simple est introduit pour illustrer les démarches. Deux séquences courtes et de taille différente sont utilisées. Afin de simplifier l'explication, le vecteur de caractéristiques est réduit à un scalaire de valeurs numériques, soit un nombre entier qui représente l'amplitude de la séquence à chaque instant (voir fig. 3.18).

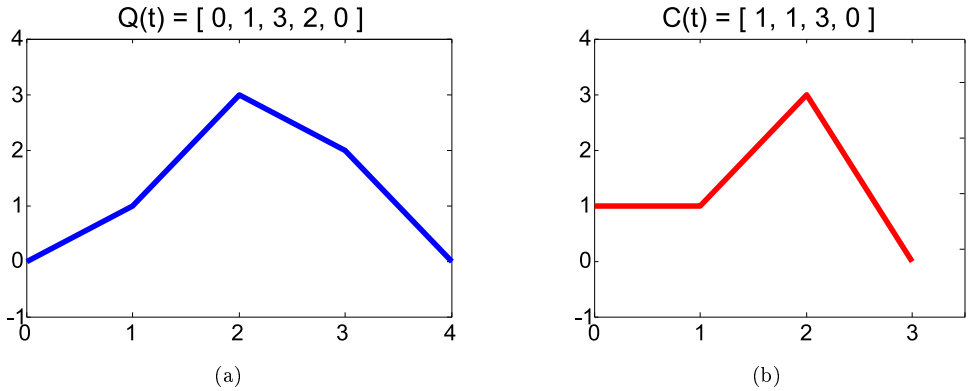


FIG. 3.18 – (a) $Q(t)$ une séquence à 5 points, (b) $C(t)$ une séquence à 4 points.

En premier lieu une matrice de la taille de $Q(t)$ sur la taille de $C(t)$ ([5x4]) est construite. Elle est remplie avec les distances locales $d(i, j)$ entre les scalaires des séquences pour toutes les positions de la matrice (local distance matrix - LDM). Dans le cas de cet exemple, la distance euclidienne peut être utilisée, ce qui revient à prendre la valeur absolue de la différence des scalaires (3.16).

$$d(i, j) = |q(i) - c(j)| \quad (3.16)$$

Une interprétation des distances locales dans l'espace des séquences est illustrée dans la figure 3.19(a). Par exemple $d(4, 1) = 1$ réalise une comparaison entre deux points des séquences relativement éloignée sur l'axe du temps.

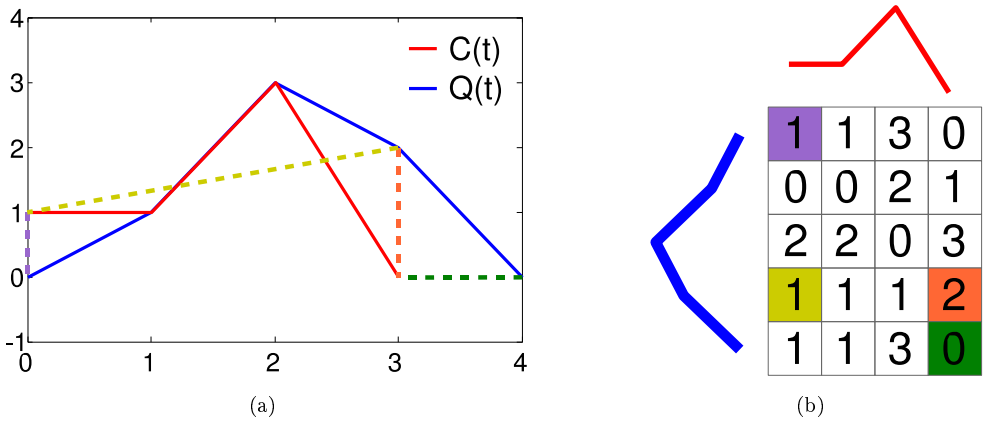


FIG. 3.19 – Calcul de la matrice des distances locales :
(a) Les deux séquences sur le même système d'axes avec quelques exemples (différentes couleurs) d'éléments comparés avec leurs positions dans la LDM ;
(b) La matrice des distances locales (LDM)

La méthode DTW consiste à trouver le minimum de la distance cumulée des cheminements possibles à travers cette matrice en garantissant les propriétés suivantes :

- Les conditions de bord sont : les premier et dernier éléments des séquences sont alignés. Le point (0,0) est le point de départ et (5,4) est le point d'arrivée de tous les chemins autorisés.
- La continuité du cheminement : chaque point des deux séquences doit être utilisé au moins une fois. Un saut dans le temps n'est pas autorisé.
- La monotonie du cheminement : il est impossible de retourner dans le temps, un arrêt provisoire est autorisé.

Les conditions de continuité et de monotonie sont garanties par la restriction à trois possibilités d'avancement pour le cheminement : horizontal ($j-1 \mapsto j$), vertical ($i-1 \mapsto i$) et en diagonale ($i-1 \mapsto i; j-1 \mapsto j$) (voir fig. 3.20). Ces conditions sont reprises dans l'équation (3.12).

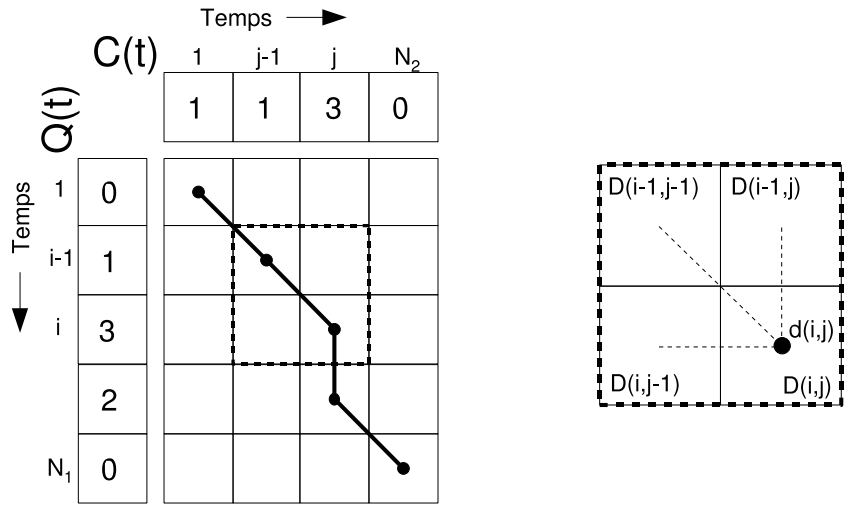


FIG. 3.20 – Le principe de la distance DTW

En se basant sur la matrice LDM il est facile de construire la matrice globale des distances cumulées (angl. global distance matrix (GDM)) de façon incrémentale en commençant au point de départ des deux séquences (0,0) et en avançant ligne par ligne ou colonne par colonne. La valeur finale au point (5,4) représente la distance DTW entre les deux séquences ($D_{dtw}(Q,C) = 2$).

La figure 3.21 montre les deux matrices, LDM et GDM pour cet exemple. Dans la GDM le chemin qui minimise la distance globale est tracé en vert. Une deuxième représentation de la GDM en 3D (fig. 3.21(d)) montre que le chemin passe dans la vallée de la surface qui est construite par les valeurs de la GDM.

À partir de ce chemin, l'alignement non-linéaire dans l'espace du temps, réalisé par le calcul, peut être représenté (voir fig. 3.21(c)).

Le chemin qui donne le minimum de la distance cumulée est trouvé par rétro-propagation après la construction de la matrice GDM. Deux possibilités existent :

- Sauvegarder pour chaque case de la matrice la direction qui a conduit au minimum. En commençant au point final (5,4) cette information conduit à la construction du chemin qui minimise la distance globale.
- Commencer au point final et avancer en cherchant pour chaque position la direction qui réalise la plus petite différence.

Ces deux possibilités ne donnent pas nécessairement le même résultat. Dans ce travail la première solution est retenue.

Un avancement du chemin en diagonale représente un avancement d'un pas de temps dans les deux séquences (alignement uniforme). Un avancement horizontal ou vertical du chemin représente un étirement respectivement une compression d'une des séquences.

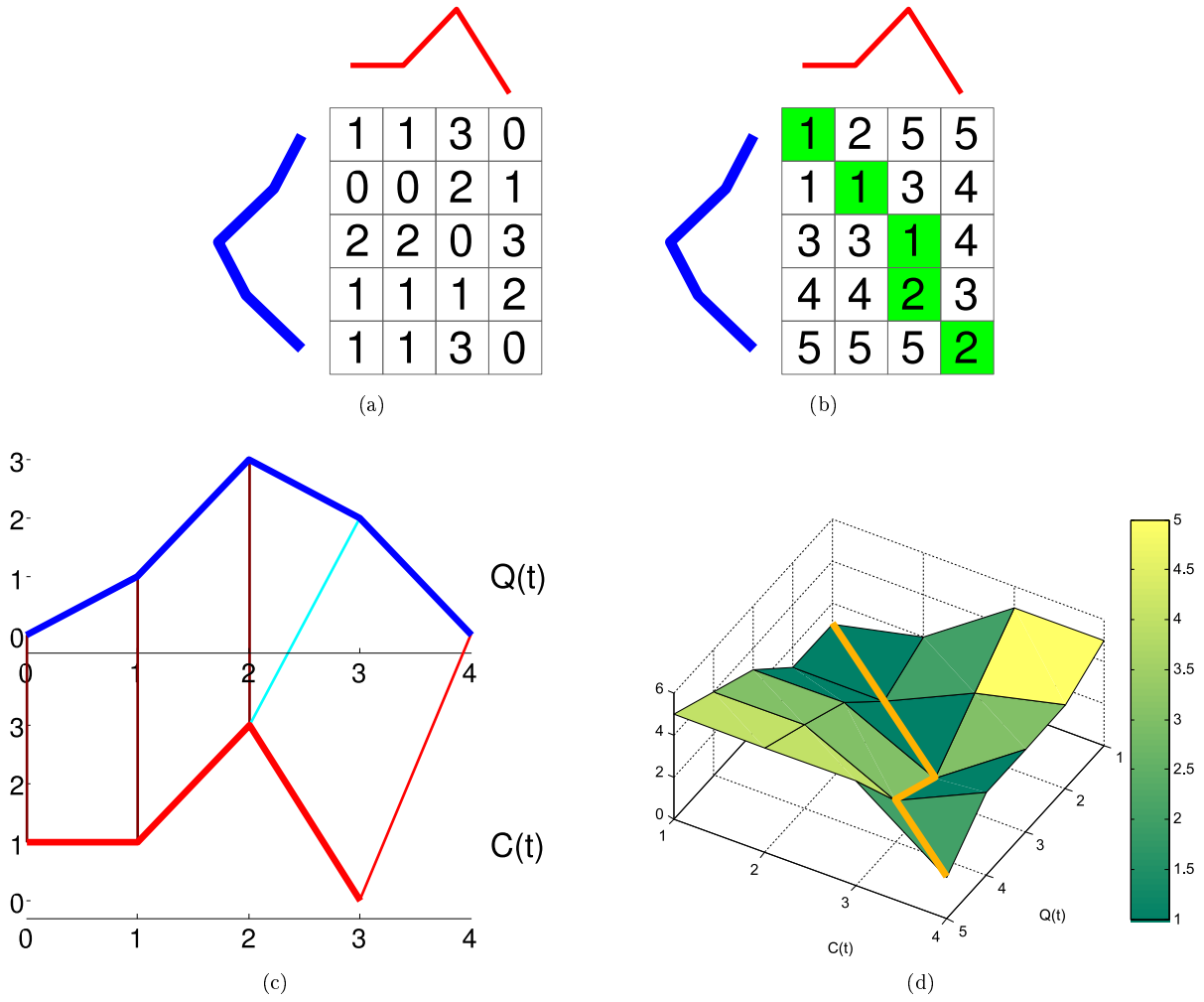


FIG. 3.21 – Calcul et alignement DTW pour l'exemple :
(a) Rappel de la matrice des distances locales ;
(b) GDM : la matrice globale des distances cumulées avec le chemin qui minimise cette distance ;
(c) L'alignement non linéaire des deux séquences ;
(d) Représentation 3D de la GDM avec le chemin parcouru.

Dans l'exemple un avancement vertical du point (3,3) au point (4,3) est réalisé. La figure 3.21(c) représente cet avancement par la compression des points $t = 3$ et $t = 4$ de la séquence Q sur un seul point $t = 3$ de C. Comme auparavant cette compression est représentée par la ligne en couleur bleue. Après la compression, le chemin avance de nouveau en diagonale, ce qui représente un alignement uniforme dans le temps. Dans la figure 3.21(c) cet alignement est en rouge clair pour indiquer qu'il s'agit d'un alignement uniforme, mais qu'il ne représente plus l'alignement initial qui est en rouge foncé.

Avec cette distance DTW il est donc possible de comparer des séquences de taille quelconque avec une souplesse dans l'alignement temporel. Pour revenir sur les trois séquences $U(t), V(t)$ et $W(t)$ (fig. 3.15(a)), il est possible de réaliser un regroupement qui cette fois-ci est basé sur la distance DTW. Le tableau 3.4 montre le résultat des distances croisées et les compare au résultat basé sur la distance euclidienne.

	Dist. eucl.	Dist. DTW
$D(U,V)$	2	2
$D(U,W)$	$\sqrt{2}$	4
$D(V,W)$	$\sqrt{2}$	4

TAB. 3.4 – Comparaison de la distance euclidienne et de la distance DTW sur les séquences $U(t), V(t)$ et $W(t)$

En utilisant la distance DTW, les séquences $U(t)$ et $V(t)$ sont regroupées en premier lieu, ce qui reflète davantage l'approche de similarité basée sur l'évolution temporelle qui est sujet de ce travail.

La définition et les bases de la distance DTW ont été expliquées et accompagnées d'un exemple simple. En vue d'une application de la méthode DTW à des problèmes réels comme : la classification des séquences de grandes tailles, des données bruitées, le temps de calcul et les données multivariées, pour ne nommer que les plus importants pour la classification non-supervisée des cycles de production du four à arc électrique, quelques analyses plus approfondies de la méthode sont nécessaires.

3.3.2.2 La distance DTW appliquée à la classification

Toute approche de classification se base sur une mesure de similarité voire de dissimilarité, ce qui est le plus souvent représenté par une mesure de distance entre les éléments analysés. Ce chapitre discute l'adéquation de la distance DTW dans ce domaine. Afin de simplifier la discussion et en même temps d'avancer vers l'application industrielle un corpus de "mots" simples, basé sur l'ensemble des lettres $\{A, B\}$, est construit (voir tab. 3.5).

	Mot	Remarque
1.	AAAAA	5 lettres identiques
2.	AAA**B**A	5 lettres avec un B à différentes positions
3.	AA**B**AA	
4.	A**B**AAA	
5.	AAAAAAAAAA	10 lettres identiques
6.	AAAAAA**BB**AA	10 lettres avec 2 B consécutifs à différentes positions
7.	AAAA**BB**AAAA	
8.	AA**BB**AAAAAA	
9.	AA**B**AAAA**B**AA	2 lettres B séparées, une au début et une à la fin
10.	A**B**ABA	
11.	AA**BB**AA**BB**AA	variante de 9. avec un dédoublement des lettres B

TAB. 3.5 – Les mots utilisés dans l'analyse de classification

Le corpus se compose de mots à deux tailles différentes (5 et 10 lettres) avec en majorité des lettres A. Afin d'analyser le comportement du calcul de dissimilarité, des lettres B sont ajoutées et déplacées de façon structurée à l'intérieur de la séquence à différents endroits.

Il reste à définir une distance locale pour ce type de séquence. L'approche la plus simple consiste à dire que la valeur de dissimilarité entre les deux lettres A et B est de "1" et à spécifier que celle entre deux lettres identiques est de "0".

La figure 3.22 montre le résultat obtenu par une classification par arbre hiérarchique en utilisant la distance DTW telle que définie au chapitre 3.3.2.1. Pour l'arbre hiérarchique, l'algorithme qui est utilisé ici représente un regroupement de deux ou plusieurs éléments par la plus petite distance entre eux.

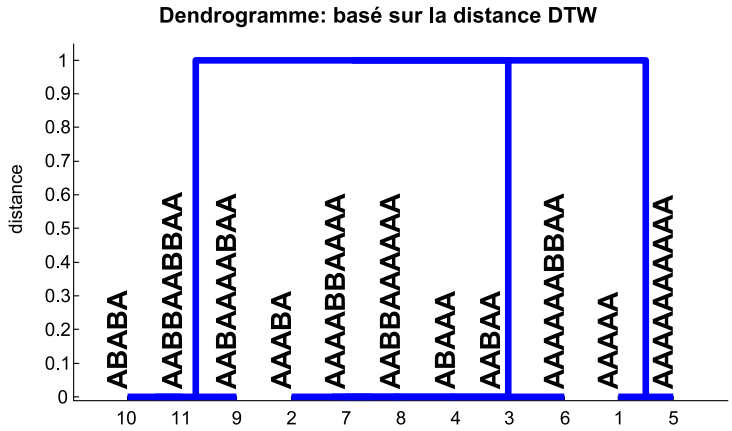


FIG. 3.22 – Classification hiérarchique basée sur la distance DTW

La méthode a bien regroupé les séquences sans lettre B. Les séquences avec une seule apparition de la lettre B sont regroupées. Même les séquences où deux B consécutifs n'apparaissent qu'une seule fois, ce qui ressemble fortement aux séquences avec un B, se retrouvent dans la même classe. Finalement une troisième classe est créée où sont regroupées les séquences où la lettre B apparaît à deux endroits dans la séquence, en incluant la séquence 11 (variante de la séquence 9) où la lettre B est dédoublée chaque fois.

En principe ce résultat semble satisfaisant, étant donné que pour le regroupement la taille de la séquence ne joue pas de rôle. Seule la forme importe pour la création des classes. Mais même si le regroupement est satisfaisant, l'organisation interne des classes donne à réfléchir. La distance entre tous les mots internes à une classe ont la valeur zéro. Cependant, même si les mots 2,3 et 4 se ressemblent, il y a une plus grande différence entre 2 et 4 qu'entre 2 et 3. Sans parler des mots 6 et 4 par exemple où la différence subjective est encore plus importante.

Il est évident, et cela a déjà été évoqué dans le chapitre 3.3.1, que la similarité est un sujet non trivial avec toujours une part de subjectivité. De ce fait, il serait intéressant pour l'utilisateur s'il pouvait influencer l'algorithme, afin de l'adapter à ses besoins. Un premier paramètre est la définition de la distance locale entre les vecteurs de caractéristiques. Mais cette mesure n'a pas d'influence directe sur le mécanisme global de la recherche du chemin minimisant la distance cumulée et donc sur l'allignement réalisé par le DTW.

Une approche intéressante a été proposée par Stuart N. Wrigley [61] appliquée dans le domaine de la reconnaissance de la parole. Elle se base sur le fait que l'algorithme DTW de base a une préférence pour le chemin diagonal. En effet, partant de (i, j) pour atteindre le point $(i+1, j+1)$, aller en diagonale ne représente qu'une seule opération. Pour utiliser un autre chemin selon les règles du DTW, deux opérations sont nécessaires, ce qui augmente la somme cumulée d'un facteur de deux, si les distances locales sont identiques. Stuart propose de ce fait de calibrer cette opération et d'ajouter une pénalité facultative pour les différentes directions. Basée sur la définition du DTW (3.12) cette modification est exprimée dans l'équation (3.17).

$$D_{dtw}(S_1(i), S_2(j)) = d(i,j) + \min \left\{ \begin{array}{rcl} \mathbf{ph} & + & D_{dtw}(S_1(i-1), S_2(j)) \\ \mathbf{pv} & + & D_{dtw}(S_1(i), S_2(j-1)) \\ \mathbf{2} & * & D_{dtw}(S_1(i-1), S_2(j-1)) \end{array} \right\} \tag{3.17}$$

Pour la paramétrisation des trois directions, il suffit de spécifier deux paramètres de pénalisation.

Cette modification donne à l'utilisateur la possibilité d'influencer le comportement global de la recherche du chemin minimal. Une pénalisation de l'étirement ou de la compression a une influence sur la valeur de la distance DTW entre deux séquences, et influe de ce fait directement sur la classification qui en est déduite. La figure 3.23 montre l'influence du paramètre pénalité sur le résultat de la classification en se basant sur le corpus des mots (tab. 3.5) et en spécifiant que $ph = pv$.

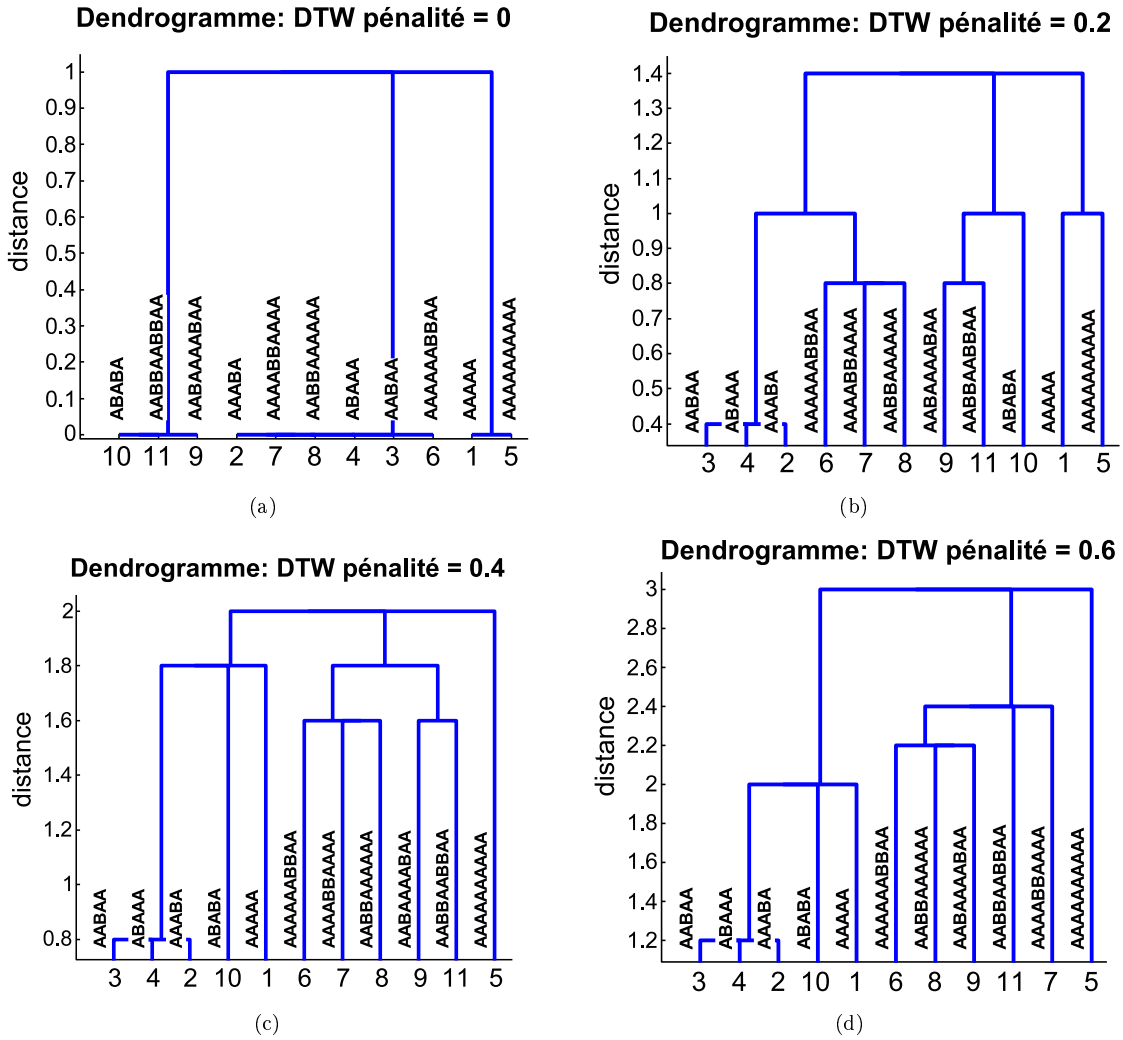


FIG. 3.23 – Influence de la pénalité directionnelle du DTW sur la classification (ici $ph = pv$) : (a) pas de pénalité $ph = pv = 0$; (b) faible pénalité $ph = pv = 0.2$; (c) pénalité moyenne $ph = pv = 0.4$; (d) forte pénalité ph = pv = 0.6.

Pour cette analyse les pénalités horizontale et verticale sont synchrones, et l'ajustement des trois directions est utilisé. Avec l'augmentation de la pénalité la distance DTW entre les mots augmente et le regroupement est modifié. Au début les mots similaires restent regroupés. La pénalité $ph = pv = 0.2$ représente le même regroupement si l'arbre est coupé au niveau de la distance de 1.2. Mais à l'intérieur de chaque classe des sous-regroupements se créent. Avec la pénalité, la longueur des mots prend de plus en plus d'importance. A la fin il y a la longueur des mots qui l'emporte sur la forme. Cela montre que ce paramètre influe bien sur le comportement du mécanisme global de la recherche du chemin minimal et modifie la distance DTW globale entre des séquences analysées.

Cette analyse soulève une question supplémentaire : la distance cumulée entre deux mots courts est par principe plus petite que celle entre deux mots longs, qui ont une dissimilarité semblable, parce que le

chemin est plus court. Afin de pouvoir comparer les distances entre de tels mots de différente taille une normalisation de la distance DTW s'impose.

Il y a plusieurs possibilités pour faire une telle normalisation. Ici deux variantes sont analysées :

Normaliser par la diagonale de la GDM : La matrice des distances cumulées (GDM) est divisée par la valeur $\sqrt{n_1^2 + n_2^2}$ où n_1, n_2 sont les tailles des séquences comparées. Cette méthode se base sur le fait que le chemin direct entre deux séquences est la diagonale. Si deux séquences se ressemblent le chemin minimal ne dévie que faiblement de la diagonale.

Normalisation par la séquence la plus longue : Ici la matrice GDM est divisée par la valeur $\max(n_1, n_2)$. La philosophie de cette approche est que, pour une séquence (longue), toute comparaison devrait être traitée de manière comparable.

Ces deux approches sont comparées dans la figure 3.24 en n'utilisant que quatre mots sélectionnés du corpus d'exemples (tab. 3.5). Puisque l'effet des différentes normalisations n'est visible que si la pénalité diffère de zéro, la pénalité est fixée à $ph = pv = 0.7$.

Il est visible que les deux normalisations ont un effet sur le regroupement réalisé. La normalisation utilisant la taille du plus long mot regroupe les mots avec un ou deux "B" et ceux sans la lettre "B" contrairement à la normalisation par la diagonale qui regroupe les mots de même taille. Le regroupement selon le contenu et non nécessairement par la taille correspond plutôt à l'approche de similarité telle qu'elle est comprise dans ce travail puisque les cycles de production peuvent avoir différentes durées et des évènements comme le chargement ou l'enclenchement de la lance à oxygène peuvent apparaître à des instants variables durant un cycle (voir chap. 1.2.3).

Ces deux extensions du calcul de la distance DTW, à savoir la pénalité des directions durant le cheminement et la normalisation de la distance cumulée par la taille des séquences analysées, ajoute des paramètres à la méthode. Ils peuvent d'un côté être utilisés afin d'adapter le calcul DTW à la subjectivité de l'utilisateur. De l'autre côté, avec le nombre de paramètres la complexité augmente, car ajuster les paramètres est généralement une tâche qui n'est pas triviale. Elle est souvent en relation avec une optimisation globale d'une méthode dans une application concrète.

3.3.2.3 Optimisation du calcul DTW

Le calcul de la distance DTW est relativement coûteux en temps de calcul. Tout d'abord une matrice des distances locales de taille $[n_1 \times n_2]$ doit être calculée. Dans un deuxième passage la même matrice est parcourue afin de minimiser la distance cumulée. Comparée au calcul de la distance euclidienne, qui est de l'ordre $O(n)$ pour des séquences de taille n, la distance DTW est en $O(n^2)$ pour les mêmes séquences. Ce point ne se fait pas remarquer dans des problèmes de la taille des exemples traités dans ce chapitre. Mais pour la classification non-supervisée des cycles de production du four à arc électrique ou pour d'autres problèmes de grandeur nature, ce point ne peut être négligé, car le temps de calcul pour la distance DTW augmente au carré avec la taille des séquences analysées.

Même si ce point n'est pas un sujet primaire de cette thèse, il est important de pointer vers plusieurs travaux qui existent sur le sujet avec en gros les approches suivantes :

- limiter les calculs sur une zone de la matrice complète [52], [5],
- utiliser des méthodes de réduction de données afin de raccourcir les séquences à manipuler (voir chap. 3.1.1),
- utiliser autant que possible un calcul estimatif rapide qui est basé sur des limites inférieures [62], [35], [31].

Le travail le plus récent est celui de Keogh [31]. Il montre entre autres qu'une indexation basée sur la distance DTW est possible, même si la distance DTW ne garantit pas l'inégalité triangulaire. Le fait de pouvoir indexer des séquences augmente, de façon importante, la rapidité d'une recherche de similarité entre une séquence donnée avec d'autres déjà indexées.

Dans le contexte de ce travail nous nous sommes limités à la méthode de réduction des données en utilisant le codage adapté aux données qui a été introduit dans le chapitre 3.2. En ce qui concerne la classification non-supervisée des séquences, une méthode basée sur des prototypes est utilisée afin d'éviter le calcul de toutes les combinaisons, comme c'est le cas avec la classification par arbre hiérarchique. Finalement, dans le projet, la phase de création du classificateur est une phase préliminaire pour laquelle

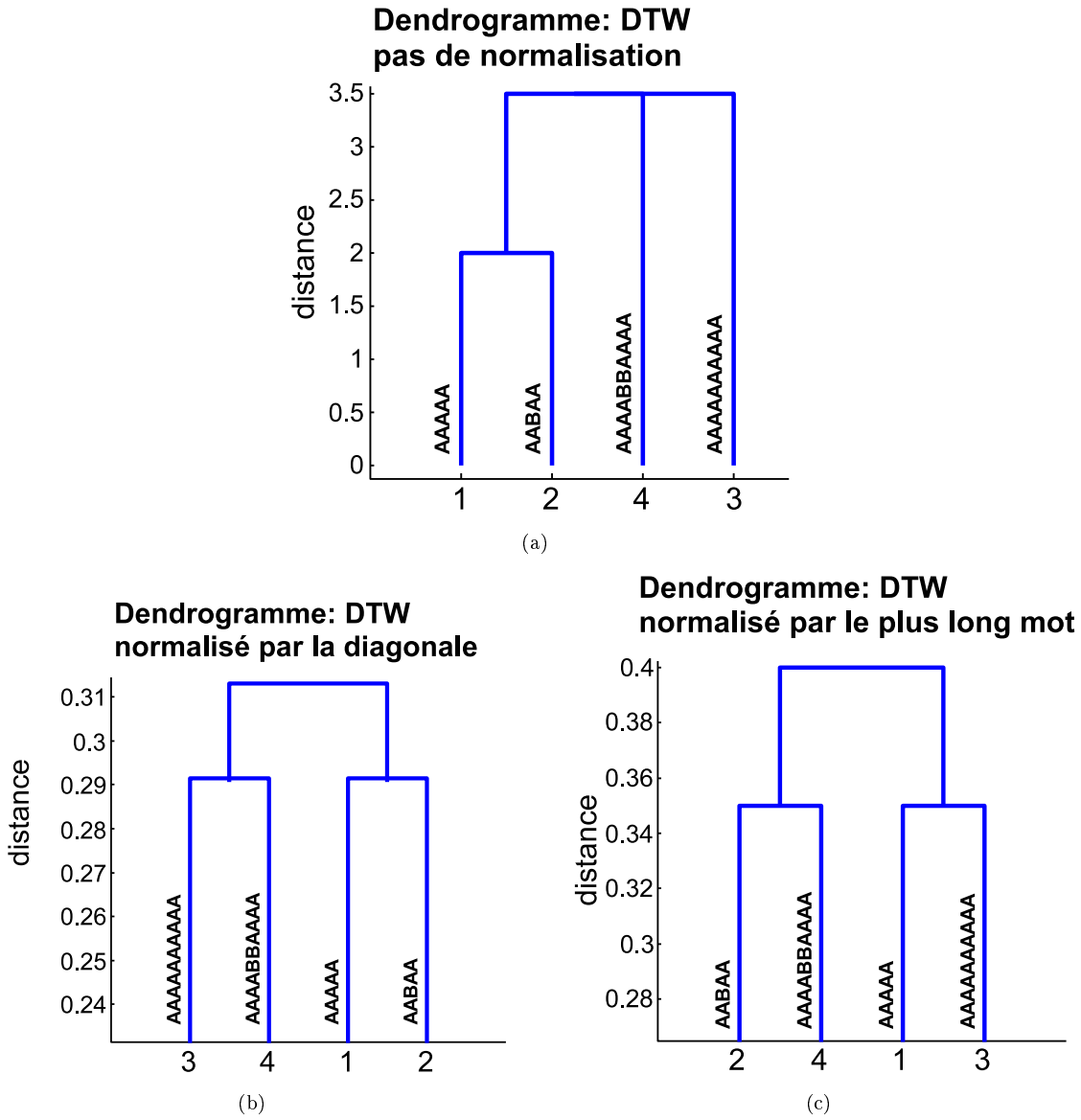


FIG. 3.24 – Méthode de normalisation de D_{DTW} par la taille des séquences : (a) Pas de normalisation afin d'avoir une référence; (b) DTW, normalisé sur la diagonale; (c) DTW, normalisé par la taille du plus long mot.

le temps de calcul n'a pas une importance aussi grande qu'il ne l'a dans la phase du contrôle en ligne. Dans la phase du contrôle une simple attribution d'une séquence à une classe est nécessaire, et cela est beaucoup moins coûteux.

3.3.2.4 La distance DTW appliquée aux données du four

Jusqu'à présent la distance DTW a été appliquée à des exemples simples et exemplaires, de petite taille avec des valeurs entières. Différentes propriétés de la distance ont été démontrées sur base de ces exemples. Quittons maintenant ce domaine synthétique et propre, et regardons comment les propriétés se comportent sur les données du four, qui sont des valeurs réelles et bruitées.

Dans le chapitre précédent le codage a été proposé comme mesure pour la réduction de la taille des données à manipuler. Le codage adapté, introduit au chapitre 3.2, est composé d'une suite de triplets avec un identifiant pour la lettre représentant la forme, et deux valeurs réelles pour la moyenne et la variance locale attribuée à la forme.

Dans la définition de la distance DTW, équations (3.12) - (3.15), les séquences S sont spécifiées comme une suite de vecteurs de caractéristiques. Ceci souligne la capacité d'une analyse multivariée, s'il est possible de définir une métrique pour la dissimilarité locale entre de tels vecteurs, ce qui en général devrait être possible.

Chaque triplet du codage adapté représente un vecteur de trois caractéristiques. Afin de pouvoir appliquer la distance DTW, il est nécessaire de définir une mesure de distance locale, c.-à-d. une mesure de dissimilarité entre deux de ces triplets.

Une définition simple et en analogie avec la distance locale utilisée dans les exemples serait la distance euclidienne entre deux triplets. En principe cela serait possible, car les caractéristiques sont toutes de nature numérique. Mais le plus important dans la définition de la distance locale est qu'elle reflète aussi la nature des données. Dans le cas présent la première caractéristique représente une forme, la deuxième l'amplitude à laquelle cette forme se trouvait dans le signal original, et la dernière l'étendue de la forme. La figure 3.25 représente deux vecteurs de caractéristiques tirés de deux cycles de production. Le codage représenté est basé sur l'alphabet A0 du codage avec 4 lettres $\{A, B, C, D\}$ représentées dans le vecteur de caractéristiques par leur index (identifiant) dans l'alphabet.

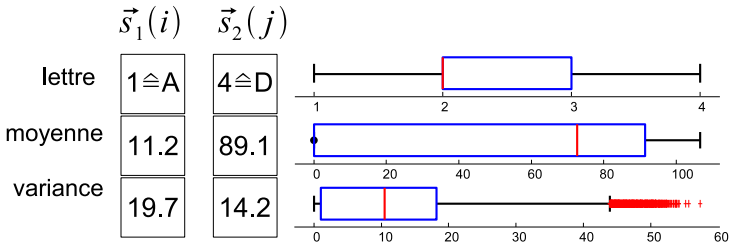


FIG. 3.25 – Exemple de deux vecteurs caractéristiques avec une information de distribution des composantes

A côté des deux vecteurs les diagrammes "boîte à moustache" (angl. Box and Whisker plot) représentent la distribution de chacune des trois caractéristiques pour un échantillon représentatif de 500 cycles codés (20%). Les diagrammes montrent que les trois caractéristiques ont des domaines de valeurs différentes : la ligne rouge représente la médiane, et la boîte indique la limite supérieure du premier et troisième quartile.

Une simple application de la distance euclidienne à ces vecteurs de caractéristiques serait fortement biaisée par la moyenne qui, pour le signal analysé de la puissance électrique, a les valeurs les plus importantes dans la plupart des cas.

Une solution est de faire une normalisation centrée réduite des éléments du vecteur sur base d'un échantillon représentatif de cycles codés.

Une question doit être soulevée dans ce contexte : est-ce que cela a un sens de calculer la différence entre les index des lettres de l'alphabet ? Pour y répondre il faut revoir la méthode utilisée pour la création des alphabets. La création d'un alphabet est basée sur une cartographie réalisée par une SOM

(voir chap. 3.2.2). L'avantage de cette méthode, et c'était une des raisons du choix de la méthode, est la relation topologique entre les éléments de la carte. Ce qui veut dire que la distance $D(A, B)$ entre la forme A et B qui sont voisins sur la carte est plus petite que $D(A, C)$ où A et C ne le sont pas. Un calcul de différence entre les index est donc légitime dans le contexte de donner une information de dissimilarité entre les différentes formes.

L'approche de la distance euclidienne basée sur des vecteurs de caractéristiques normalisés de façon centrée réduite a été la première réalisation qui a déjà donné des résultats encourageants.

Mais les analyses faites sur l'apprentissage et la paramétrisation de la SOM (annexe B) ont conduit à la définition (3.18) de la distance locale. L'équation utilise les notations du chapitre 3.2.3 pour le codage et celles du chapitre 3.3.2.1 pour la distance locale.

$$d(\vec{s_1}(i), \vec{s_2}(j)) = \sqrt{D_{eucl}(C_1(i), C_2(j))^2 + (\tilde{\mu}_1(i) - \tilde{\mu}_2(j))^2 + (\tilde{\sigma}_1(i) - \tilde{\sigma}_2(j))^2} \tag{3.18}$$

Les variables $\tilde{\sigma}, \tilde{\mu}$ indiquent les formes normalisées (3.3) des éléments "moyennes" et "variances" du triplet de codage.

$D_{eucl}(C_1(i), C_2(j))$ représente la distance euclidienne entre les formes correspondant aux index dans les vecteurs caractéristiques.

Cette définition est un compromis entre temps de calcul et précision. En fait la distance locale réelle, basée sur une métrique euclidienne, serait obtenue en reconstruisant la partie du signal pour les deux triplets en dénormalisant chaque triplet (3.4) et en calculant la distance euclidienne entre ces parties locales, ou mieux encore, en calculant une valeur de dissimilarité locale sur base d'une distance DTW afin de garder une souplesse dans l'alignement même au niveau du codage. Réaliser un tel calcul pour chaque distance locale serait très coûteux en temps de calcul.

Dans la définition (3.18) la distance euclidienne entre les lettres de l'alphabet peut être calculée au préalable et accédée par simple indexation durant le calcul de la distance locale (lookup table). Il est possible de faire remarquer que l'équation (3.18) réalise une pondération uniforme des éléments normalisés du triplet. Cela représente un choix implicite d'une paramétrisation qui n'est pas analysé selon son adéquation en ce qui concerne la tâche de comparaison des cycles de production. Une telle analyse pourrait fournir davantage de possibilités afin d'ajuster la méthode à la subjectivité de l'utilisateur.

En ce qui concerne le calcul de la distance cumulée, la formule modifiée du DTW (3.17) avec pénalisation est utilisée. Les paramètres de pénalisation ph, pv ont la même valeur parce que, dans ce contexte, un cheminement symétrique est souhaité. En ce qui concerne la valeur de la pénalisation, des analyses ont montré qu'il est plus intuitif pour l'utilisateur d'orienter la pénalisation aux valeurs des distances locales et que cette approche donne de bons résultats. Un exemple souligne cette affirmation.

Dans l'exemple des arbres hiérarchiques (voir chap. 3.3.2.2) des pénalités entre "0" et "0.6" ont été analysées. Dans cet exemple la distance locale, qui est représentée par la distance entre les lettres A, B, a été définie à la valeur 1. C'est pourquoi des valeurs de pénalité $< 1/3$ qui sont ajoutées à la distance cumulée, ont déjà une influence sur la distance DTW.

Pour les données du four, l'exemple de la figure 3.25 peut être utilisé. En se basant sur l'alphabet A0 de quatre lettres ayant une taille de 12 pas de temps pour la forme, et en se basant sur les valeurs normalisées, la distance locale a la valeur suivante :

$$d(i, j) = \sqrt{1.0054 + (-1.0648 - 0.8627)^2 + (0.9139 - 0.3598)^2} = 5.0277$$

Une pénalité entre $[0, 1]$ sur cette valeur a moins d'influence. Donner une autre valeur fixe, comme par exemple 3 ou 6, est difficile à gérer, parce que les valeurs de la distance locale dépendent de l'alphabet utilisé, même si les valeurs de moyenne et de variance ainsi que les formes des alphabets sont normalisées.

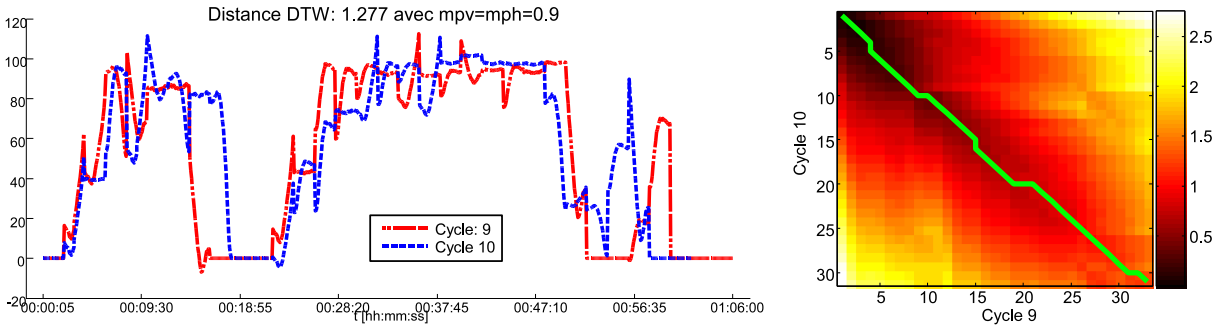
Une méthode qui donne de bons résultats et qui s'adapte automatiquement aux séquences analysées est de prendre un multiple de la médiane des valeurs de la matrice des distances locales (3.19) et (3.20).

$$\text{ph} = \mathbf{mph} * \text{median(LDM)} \tag{3.19}$$

$$\text{pv} = \mathbf{mpv} * \text{median(LDM)} \tag{3.20}$$

Avec ph et pv les variables de l'équation (3.17) et **mph** et **mpv** les nouveaux paramètres pour l'utilisateur.

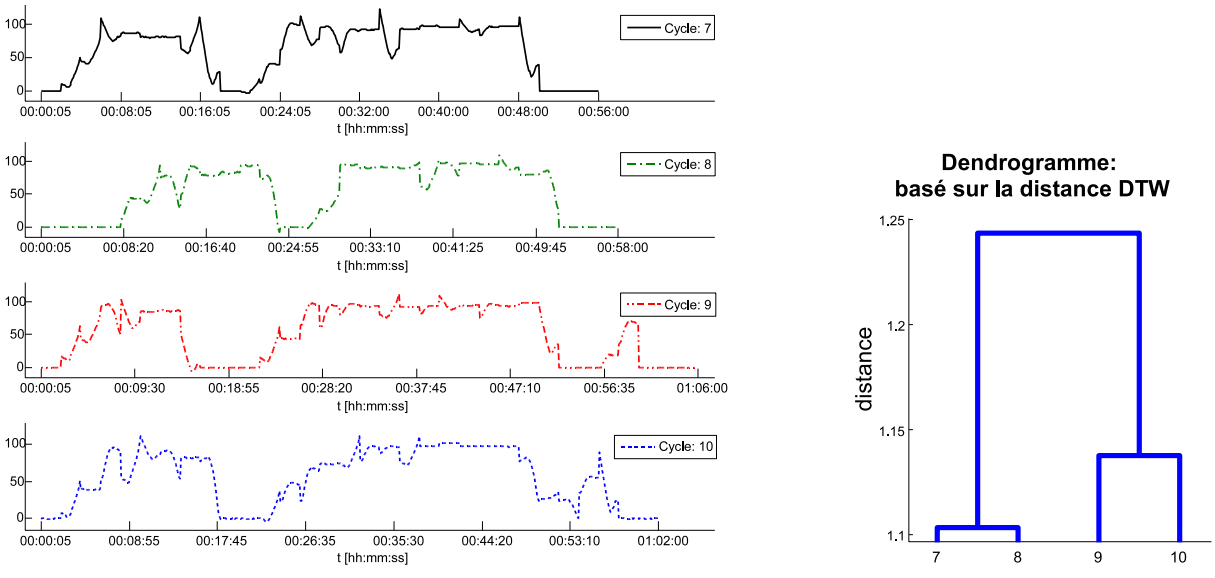
La figure 3.26 montre le résultat d'un calcul de distance DTW entre deux cycles de production (cycle 9 et 10) en se basant sur le codage réalisé avec l'alphabet A2.



(a) (b)

FIG. 3.26 – Distance DTW entre deux cycles de production : (a) reconstruction des signaux à partir du codage ; (b) la matrice globale des distances cumulées avec le chemin minimal.

Après l'adaptation de la distance DTW aux données du four il est possible de continuer avec la tâche de réaliser un classificateur pour les cycles de productions, qui se base sur l'évolution temporelle. Avec ce calcul de distance une classification par arbre hiérarchique est réalisée pour quatre cycles basés sur le codage avec A2. La figure 3.27(a) montre la reconstruction des quatre cycles.



(a) (b)

FIG. 3.27 – Classification hiérarchique de quatre cycles de production basée sur la distance DTW : (a) reconstruction des quatre signaux à partir du codage A2 ; (b) Dendrogramme du résultat de la classification basée sur la distance DTW.

Basé sur la distance DTW entre tous ces cycles la figure 3.27(b) montre la classification hiérarchique réalisée en forme de dendrogramme. Pour le calcul des distances DTW une pénalité symétrique de $mpv = mph = 0.9$ est utilisée. Le résultat de cette classification est satisfaisant, car un expert du domaine aurait regroupé les cycles de la même manière.

Cependant la classification par arbre hiérarchique n'est pas utilisable pour la tâche d'extraction de classes d'évolutions du processus, puisque le nombre de cycles qui doit être utilisé afin d'avoir un corpus

représentatif est trop important pour une telle méthode. La section suivante va se consacrer à une méthode de classification qui se base sur des prototypes.

3.3.3 Classification non-supervisée basée sur des prototypes

La méthode de classification basée sur les arbres hiérarchiques, qui a été utilisée dans les exemples de la section précédente 3.3.2.2 ne peut pas être utilisée sur un grand nombre d'échantillons, car le temps de calcul de cette méthode augmente de façon polynomiale avec le nombre d'échantillons puisqu'une combinatoire complète entre les échantillons est nécessaire. Le temps de calcul élevé de la distance DTW aggrave encore la situation.

Une deuxième méthode qui n'est pas seulement bien connue dans le domaine du connexionnisme, en rapport avec la classification, se base sur des prototypes et utilise un apprentissage basé sur la compétition (voir fig 3.28). Un aperçu des principes de la méthode et de quelques variantes a été donné au chapitre 2.2.2. Ce genre de méthodes a l'avantage d'augmenter sa complexité de façon linéaire avec le nombre d'échantillons pour un nombre fixe de prototypes. Un inconvénient par rapport aux arbres hiérarchiques est que la hiérarchie complète n'est pas connue après l'apprentissage.

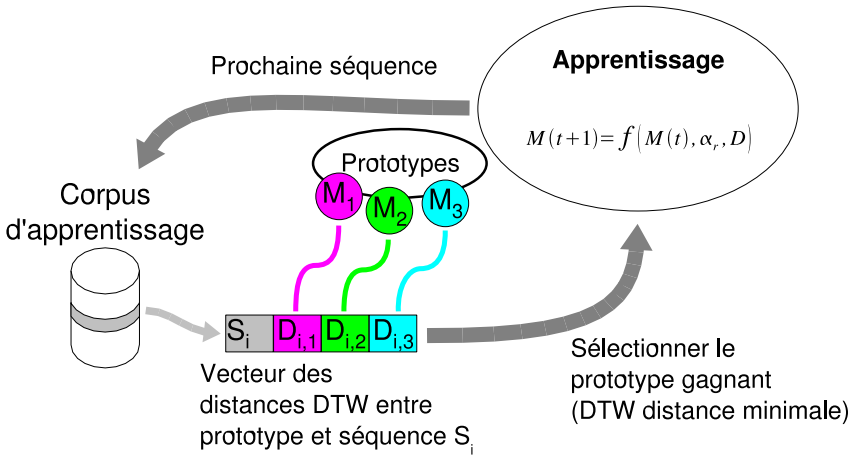


FIG. 3.28 – Principe de la classification non-supervisée basée sur des prototypes et un apprentissage compétitif

C'est ce genre d'approches qui sera utilisé afin de construire de façon non-supervisée un classificateur qui regroupe les cycles de production selon leur évolution.

La suite de cette section donne les détails de l'approche qui est utilisée et explique les outils nécessaires pour l'apprentissage compétitif dans le contexte des évolutions temporelles. Comme dans les sections précédentes, les explications sont soutenues par des exemples simples. Finalement la section va se terminer avec l'application de la méthode aux données de l'installation industrielle.

3.3.3.1 La méthode basée sur un apprentissage compétitif.

Dans le contexte de ce travail, une innovation est d'avoir proposé une extension permettant de regrouper des séquences temporelles de taille quelconque. Nous présentons cette extension ci-dessous.

Afin de limiter la complexité et de se focaliser sur le sujet de la similarité, une méthode relativement simple, comparée au variantes listées au chapitre 2.2.2, est utilisée. Il s'agit d'une méthode proche de la méthode des k-means ou encore de l'agrégation autour d'un centre mobile. Comme toutes les approches qui se basent sur des prototypes avec un apprentissage compétitif, il s'agit d'une approche itérative composée des phases suivantes :

1. initialisation des prototypes,

$$\mathcal{M} = \{M_1, M_2, \cdots, M_n\} \tag{3.21}$$

2. sélection aléatoire d'une séquence S_i du corpus d'apprentissage,

$$\mathcal{S} = \{S_1, S_2, \cdots, S_t\} \tag{3.22}$$

   avec S_i tel que spécifié par l'équation (3.11)
3. recherche du prototype gagnant (BMU, angl. Best Mapping Unit)(min. de D_{dtw}) en se basant sur la distance DTW selon les équations (3.17), (3.13)-(3.15), (3.18) et (3.19)-(3.20),

$$M_b^{S_i} = \text{bmu}(S_i) = \arg\min_{j \in \mathcal{M}} (D_{dtw}(S_i, M_j)) \tag{3.23}$$

4. itération de 2 - 3 pour tout le corpus d'apprentissage,
5. apprentissage c.-à-d. adaptation des prototypes M_j vers les séquences qui lui sont attribuées $\mathcal{S}_j(t)$,

$$M_j(t+1) = f\left(M_j(t), \alpha_r(t), \mathcal{S}_j(t)\right) \tag{3.24}$$

6. itération des points 2 - 5 pour un nombre fixe d'époch.

Les phases 3 et 5 de cet algorithme sont les plus intéressantes dans le contexte de ce travail, car c'est dans la recherche du BMU et dans l'apprentissage des prototypes que la similarité joue un rôle.

Afin de trouver le prototype gagnant (BMU) (3.23), c'est à dire le prototype qui ressemble le plus à la séquence sélectionnée, la distance DTW est utilisée. Elle a été introduite et discutée dans la section 3.3.2. Un vecteur de distances DTW est calculé. Ce vecteur contient la distance DTW entre la séquence S_i analysée et tous les prototypes $M_j \in \mathcal{M}$. Le prototype gagnant (BMU) est celui qui réalise la plus petite distance DTW.

Dans la cinquième phase toutes les séquences attribuées aux prototypes $M_j(t)$ c.-à-d. les $\mathcal{S}_j(t)$ ainsi que la distance DTW respective réalisée sont utilisées pour l'apprentissage des prototypes. L'apprentissage se fait après que tout le corpus d'apprentissage ait été analysé et les séquences soient attribuées au prototype le plus proche. Cette approche est aussi connue sous le nom de "mode batch".

La phase d'apprentissage des prototypes soulève un nouveau problème. Comment adapter les prototypes aux séquences qui lui sont attribuées afin d'obtenir un prototype représentatif pour des séquences similaires, tout en assurant une convergence de l'approche vers une classification où les classes sont compactes et bien éloignées les unes des autres ?

Dans la figure 3.28 l'adaptation d'un prototype est décrite selon une loi globale d'apprentissage (3.24). A une epoch donnée t la modification du prototype $M(t+1)$ est une fonction du prototype actuel $M(t)$, d'un taux d'apprentissage $\alpha_r(t)$ et des séquences qui lui sont attribuées $\mathcal{S}_j(t)$.

Pour la définition de la loi d'apprentissage il est possible de se baser sur une forme classique du domaine de l'apprentissage non-supervisé qui se base sur la règle de Hebb [22]. L'idée principale était de reprendre l'apprentissage qui est utilisé pour les SOM [37] (3.25), qui est bien connu et dont il existe des analyses sur les propriétés de l'apprentissage, comme par exemple la convergence de la méthode.

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i(x)}(n)(x - w_j(n)) \tag{3.25}$$

Avec w_j le vecteur des poids, η le taux d'apprentissage, x le vecteur d'entrées et $h_{j,i(x)}$ la fonction de voisinage du neurone gagnant $i(x)$ pour le neurone j.

Afin de mieux comprendre la particularité du contexte de ce travail et l'argumentation de la suite de ce chapitre, analysons quelques instants cette loi d'apprentissage et posons-la dans son contexte. Les cartes auto-organisatrices (SOM) (fig. 3.29) ont déjà été discutées (chap. 2.2.2). Ici la relation entre le vecteur des poids de chaque neurone de la carte et le vecteur d'entrée est mise en question. Chaque neurone est connecté à chaque élément du vecteur d'entrée et possède de ce fait une représentation dans l'espace d'entrée. Le but de l'apprentissage est de positionner les neurones dans l'espace d'entrée afin qu'ils représentent la topologie des entrées qui sont dans le corpus d'apprentissage. Cette relation entre l'espace d'entrée et la représentation des neurones (vecteur des poids) est la base de bon nombre de méthodes classiques basées sur des prototypes utilisant un apprentissage non-supervisé.

Analysons le contexte de ce travail dans cette optique. La séquence est représentée par une suite de vecteurs de caractéristiques. Afin de simplifier la réflexion, supposons que ce vecteur soit réduit à un

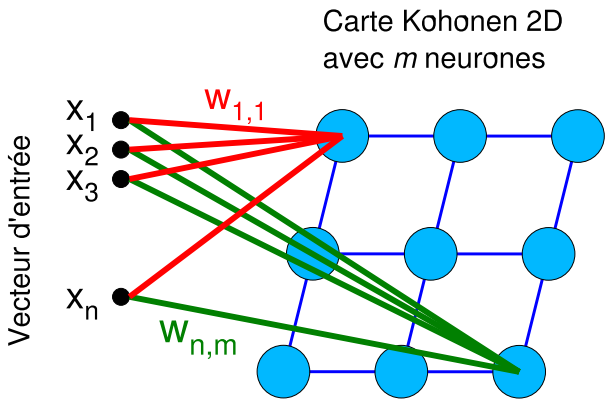


FIG. 3.29 – Carte auto-organisatrice de Kohonen (SOM) 2D avec le vecteur d'entrée x et les vecteurs des poids w_i

scalaire, comme c'était le cas des exemples qui précèdent ce chapitre. La séquence devient une suite de scalaires, un vecteur, qui représente le vecteur d'entrée du classificateur. La taille de ce vecteur donne la dimension de l'espace d'entrée. Une séquence et un prototype peuvent avoir des tailles différentes ce qui veut dire qu'ils représentent des vecteurs d'entrées dans différents espaces $S_i \in \mathcal{R}^n$ et $M_j \in \mathcal{R}^m$ avec $n \neq m$. Notez en plus que les prototypes $M_i \; \forall i \neq j$ ne vont généralement pas avoir la même taille.

La loi d'apprentissage (3.25) de la SOM qui utilise une différence vectorielle entre x le vecteur d'entrée et w_j le vecteur des poids du neurone j n'est pas utilisable tel quel.

En regardant plus en détail la méthode de calcul du DTW (chap. 3.3.2.1), il est remarqué que la méthode ne fournit pas seulement une valeur de dissimilarité entre deux séquences, elle produit aussi un cheminement dans la matrice des distances cumulées qui conduit à la distance DTW. Dans le contexte de la classification non-supervisée, ce cheminement est utilisable afin de réaliser une mise à échelle non-linéaire dans le temps entre la séquence et le prototype. Ce mécanisme sera appelé dans la suite du document le "mappage DTW" et il est expliqué dans le chapitre 3.3.3.2. Considérons ici seulement que après le mappage DTW la séquence et le prototype ont la même taille, ils sont donc dans le même espace, et les opérations de différence vectorielle ou autres sont possibles.

Dans le contexte de l'apprentissage des prototypes, il est important de dire que le mappage DTW peut se faire dans les deux directions : la séquence peut être mappée vers le prototype ou inversement, le prototype peut être mappé vers la séquence. Cette particularité sera analysée dans la section du mappage DTW.

Les propriétés de ce mappage et de l'impact d'une tel distortion non-linéaire sur l'algorithme de la classification n'est pas connue. Afin de garder l'approche aussi simple que possible, la partie du voisinage de l'apprentissage n'est pas considérée dans le cadre de ce travail, ce qui facilite l'analyse des mécanismes en relation avec le calcul DTW. Afin d'enlever le voisinage de (3.25), la fonction de voisinage suivante peut être définie.

$$h_{j,i(x)}(n) = 1 \tag{3.26}$$
$$h_{k,i(x)}(n) = 0 \qquad \forall k \neq j \tag{3.27}$$

Après reformulation, adaptation à la nomenclature utilisée pour les prototypes et l'utilisation de la méthode batch, une loi d'apprentissage relativement simple (3.28) est obtenue.

$$M_j(t+1) = (1 - \alpha_r(t))M_j(t) + \alpha_r(t)\bar{S}_{M_j}(t) \tag{3.28}$$

Avec $\bar{S}_{M_j}(t)$ la séquence moyenne des séquences $\mathcal{S}_j(t)$ qui sont attribuées à l'instant t au prototype M_j. Cette séquence moyenne est obtenue avec l'aide du mappage DTW.

Cette loi d'apprentissage est comparable à celle utilisée par la méthode souvent utilisée et bien documentée des "centres mobiles" ou des "k-means", sauf qu'ici un seul taux d'apprentissage est utilisé pour

tous les prototypes. Après analyse de cette méthode d'apprentissage, l'incorporation du voisinage afin de mener vers une méthode SOM utilisant le DTW comme mesure de dissimilarité est faisable sans trop de modifications. Cependant cette démarche n'a pas été réalisée dans le cadre de ce travail.

3.3.3.2 Mappage DTW

Reprenons l'exemple simple utilisé durant l'introduction des bases du calcul DTW (fig. 3.21), et analysons plus en détail le chemin qui est parcouru dans la matrice des distances cumulées.

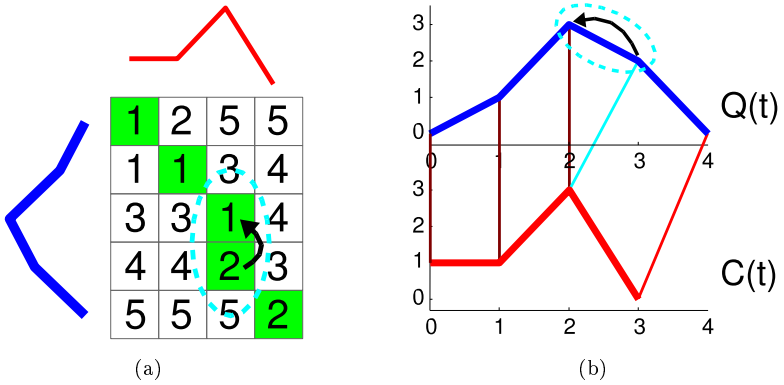


FIG. 3.30 – Relation entre cheminement et alignement DTW :
(a) GDM avec le chemin qui minimise la distance cumulée ;
(b) L'alignement non linéaire des deux séquences.

La figure 3.30 montre la GDM et l'alignement qui est fait par le calcul DTW. Les points 2 et 3 de $Q(t)$ sont comparés à un seul point (2) de $C(t)$. En comprimant ces deux points de $Q(t)$ sur un seul, la taille de la séquence résultante $Q'(t)$ est identique à celle de $C(t)$ (voir 3.31(a)). Par cette opération une mise à échelle de deux séquences est réalisable en appliquant, de façon non-linéaire, une compression à l'endroit où l'impact (le coût) pour la similarité est minimal. Notez que l'opération peut aussi se faire dans le sens inverse. Pour l'exemple cela voudrait dire mettre la séquence $C(t)$ à l'échelle de $Q(t)$ (voir 3.31(b)). Pour ce faire un étirement du point 2 de $C(t)$ sur deux points est demandé. Les lignes en pointillés représentent à chaque fois la séquence mappée dans le repère de l'autre séquence.

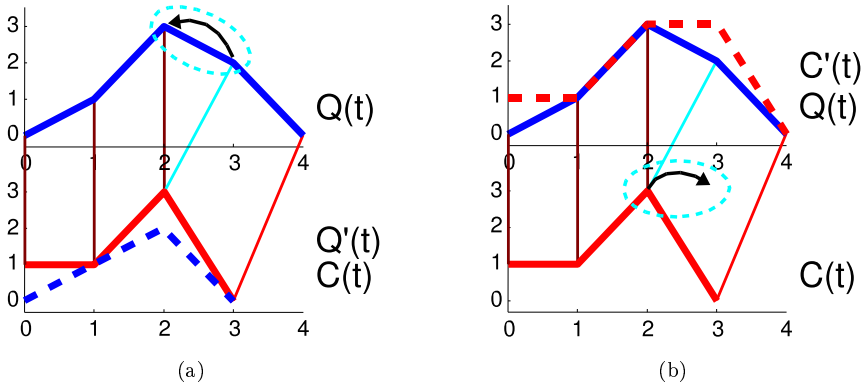


FIG. 3.31 – Mappage DTW : Une mise à échelle bi-directionnelle :
(a) mappage de $Q(t)$ à l'échelle de $C(t)$;
(b) mappage de $C(t)$ à l'échelle de $Q(t)$.

D'un côté, pour réaliser une compression, des points d'une séquence sont éliminés, ce qui représente une perte d'information, même si elle se fait à un endroit de la séquence où l'impact est minimisé. De l'autre

côté, pour l'étirement, des points sont ajoutés à la séquence. Dans ce cas il y a création d'information. Plusieurs possibilités de réalisation existent pour les deux actions. Dans les exemples de cette section la compression est réalisée par un simple enlèvement des points de la séquence. Pour l'étirement un dédoublement du point est utilisé. Ce sujet est discuté plus en détail dans la section 3.3.3.3.

Une remarque s'impose pour éviter la confusion durant la suite des discussions : s'il est question de compression ou d'étirement, il faut toujours une séquence de référence. La compression de deux points sur un seul pour une séquence $Q(t)$, expliquée auparavant, peut aussi être vue comme un étirement d'un point sur deux par l'autre séquence $C(t)$.

Un exemple légèrement plus complexe (fig. 3.32) montre que les deux opérations (compression et étirement) apparaissent généralement en même temps. Ceci devient encore plus visible dans la partie application aux données du four.

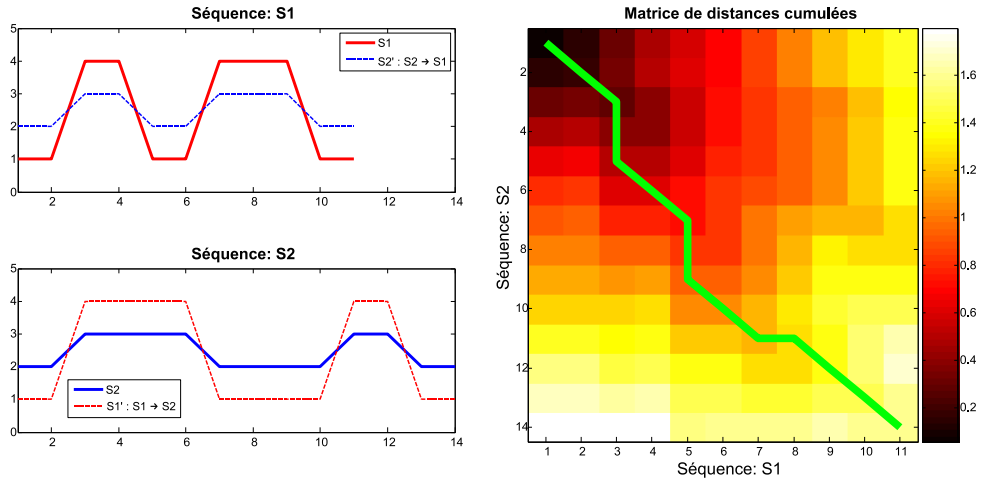


FIG. 3.32 – Exemple de mappage avec compression et étirement

Cette section du mappage finit avec un exemple qui démontre l'effet qu'a la pénalité du cheminement sur le mappage DTW. L'exemple de la figure 3.32 utilise une pénalité de $mph = mpv = 0.6$. Le mappage réalisé pour le même exemple avec une pénalité de $mph = mpv = 1$ est illustré dans la figure 3.33.

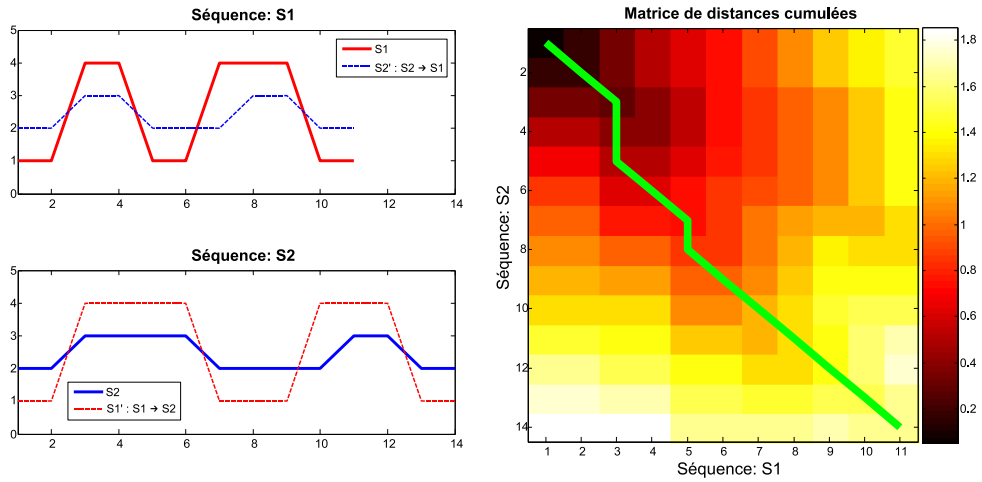


FIG. 3.33 – Influence de la pénalité sur le mappage DTW

Avec une pénalité plus importante, un chemin plus direct dans la matrice des distances cumulées en résulte. Au niveau des signaux après le mappage cela se manifeste par un plus faible alignement du

deuxième créneau des deux séquences.

Cette valeur joue donc bien le rôle d'un paramètre afin d'adapter le comportement du mappage aux besoins subjectifs du domaine analysé.

3.3.3.3 Apprentissage basé sur le mappage DTW

Le mappage DTW qui a été introduit dans la section 3.3.3.2 est une méthode avec laquelle il est possible de réaliser une mise à échelle de deux séquences. L'équation (3.28) décrit la loi d'apprentissage utilisée pour la modification incrémentale des prototypes afin d'obtenir des prototypes représentatifs pour les classes de séquences. Dans cette relation $\bar{S}_{M_j}(t)$ représente une séquence moyenne des séquences attribuées au prototype M_j à l'itération t.

Afin de pouvoir implémenter cette loi d'apprentissage, $\bar{S}_{M_j}(t)$ doit avoir la même taille que $M_j(t)$. En utilisant le mappage DTW il est possible de produire :

– les séquences mappées

$$S'_{k,j} = \mathrm{MAP}_{dtw}(S_k, M_j) \quad \forall S_k \in \mathcal{S}_j(t) \tag{3.29}$$

– la séquence moyenne

$$\bar{S}_{M_j}(t) = \mathrm{mean}(S'_{k,j}) \tag{3.30}$$

Dépendant du codage utilisé pour les séquences, la moyenne doit être spécifiée explicitement. Avec l'alphabet $\{A, B\}$ par exemple seules les deux formes sont possibles, et la moyenne doit être ajustée à l'une ou l'autre. Pour le codage adapté, utilisé pour les données du four, la situation est moins simple encore et sera discutée plus loin dans cette section.

Afin d'analyser et d'illustrer le comportement de la méthode, le corpus des mots de l'exemple utilisé pour la classification hiérarchique (tab. 3.5) est réutilisé. En vue d'approcher avec cet exemple les séquences réelles de l'installation industrielle, une forme de quatre points est attribuée à chaque lettre de l'alphabet {A,B} (fig. 3.34(c)). En ajoutant une moyenne et une variance au vecteur des caractéristiques, un codage tel qu'il est utilisé pour les séquences de l'installation industrielle est réalisé (fig. 3.34(a)).

La figure 3.34 représente deux des séquences qui seront utilisées dans cet exemple. Le code des séquences S_2 et S_{11} est repris par les deux tableaux 3.34(a) et 3.34(b). Dans le code de S_{11} trois valeurs sont soulignées, parce qu'elles déviennent légèrement des valeurs de moyenne ou de la variance utilisées pour les autres séquences. La motivation de cette variation est une meilleure démonstration des propriétés de la méthode.

Afin de montrer la forme réelle des deux séquences, elles sont décodées et illustrées dans la figure 3.34(d). Cet exemple dispose donc d'un corpus de onze séquences (fig. 3.35) ayant un codage similaire au codage adapté (chap. 3.2).

Selon l'algorithme (voir chap. 3.3.3.1), la première phase consiste à initialiser des prototypes. Le but est de pouvoir comparer le résultat obtenu ici avec celui qui était obtenu pour le corpus de base en utilisant la classification hiérarchique (fig.3.22, 3.23(b)). En regardant cette classification hiérarchique sur le niveau global, trois classes étaient obtenues. Cet exemple va donc utiliser trois prototypes afin d'apprendre trois regroupements. Ils sont initialisés en se basant sur un savoir préalable :

– Deux tailles de séquences sont comprises dans le corpus : un codage de 5 formes, et un codage de 10 formes.
– Les séquences qui existent dans le corpus sont ou bien des séquences uniformes ou bien elles contiennent un ou deux créneaux.

Afin d'avoir une différence de taille avec les séquences et une différence de taille entre les prototypes, deux prototypes avec une taille de sept formes et un autre avec une taille de huit formes sont utilisés. Concernant l'évolution, un prototype uniforme (n'utilisant que des formes du type $\{A\}$), un prototype avec un créneau central utilisant la forme $\{B\}$ et un prototype avec deux créneaux symétriques sont réalisés. La figure 3.36 montre les prototypes initiaux.

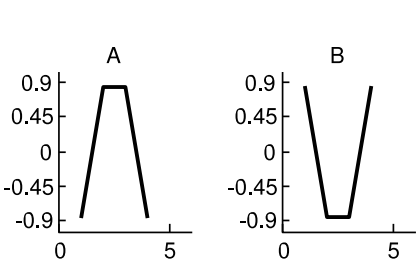
Notez que pour les prototypes, ni la position (moyenne) ni l'étendue (variance) des formes, telles qu'elles existent dans les séquences, ne sont reprises. Le but de cet exemple est d'analyser la modification des prototypes avec l'avancement de l'apprentissage.

C(i)	A	A	A	B	A
$\mu(i)$	0.5	0.5	0.5	-0.5	0.5
$\sigma(i)$	0.577	0.577	0.577	0.577	0.577

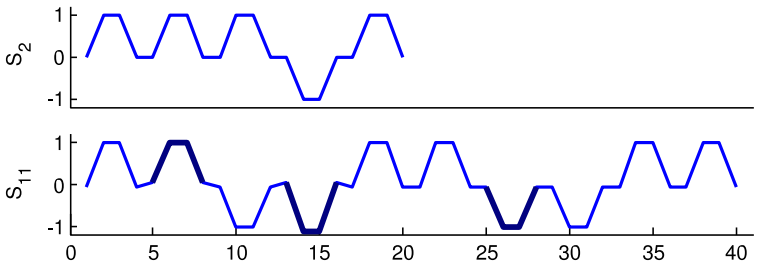
(a)

C(i)	A	A	B	B	A	A	B	B	A	A
$\mu(i)$	0.5	**0.55**	-0.5	-0.5	0.5	0.5	-0.5	-0.5	0.5	0.5
$\sigma(i)$	0.577	0.577	0.577	**0.635**	0.577	0.577	**0.52**	0.577	0.577	0.577

(b)



(c)



(d)

FIG. 3.34 – Un exemple du corpus avec les formes primitives de l'alphabet :
(a) Code de la séquence Nr. "2" du corpus ;
(b) Code de la séquence Nr. "11" du corpus ;
(c) L'alphabet des formes primitives utilisé pour cet exemple ;
(d) Deux séquences reconstruites sur base du codage et de l'alphabet.

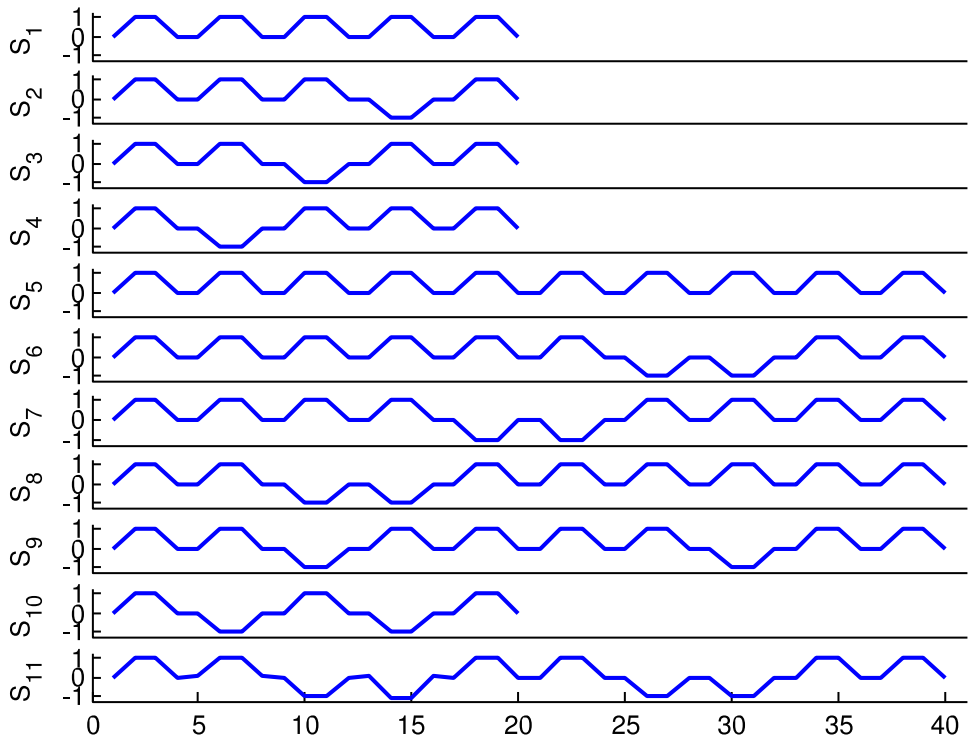


FIG. 3.35 – Corpus des séquences utilisées dans l'exemple

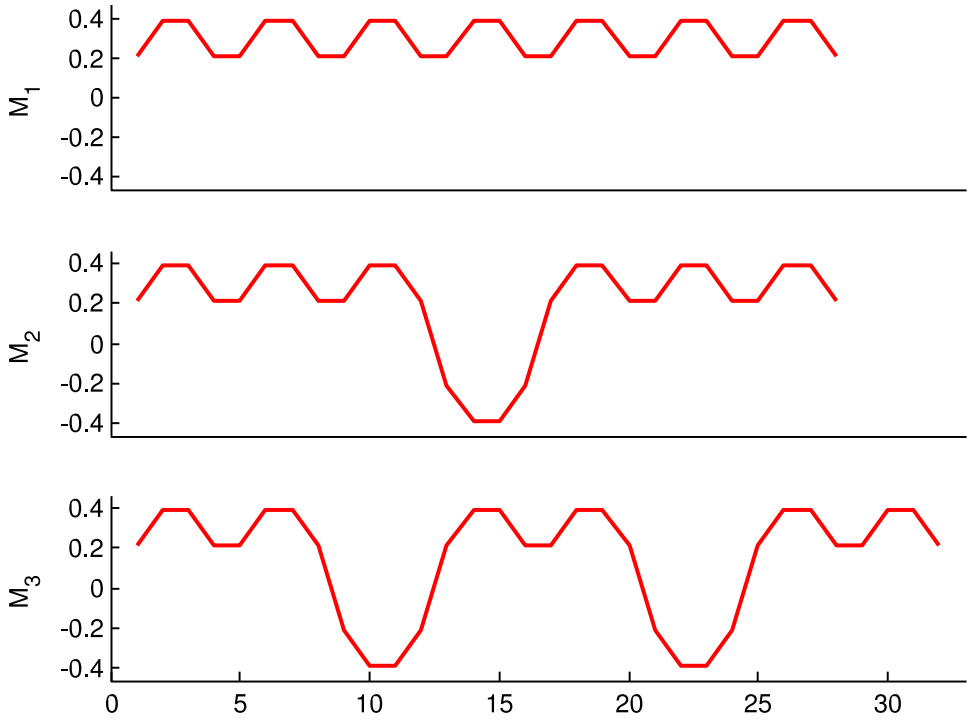


FIG. 3.36 – Prototypes initiaux utilisés dans l'exemple

Remarquez que déjà pour cette première attribution (voir tab. 3.6), le même regroupement que celui réalisé par la classification hiérarchique (fig.3.23(b)) est réalisé. Notez que le but primaire de cette analyse est l'adaptation des prototypes et non le résultat de la classification qui en découle. Les prototypes étaient initialisés de manière à favoriser l'obtention de ce résultat.

La prochaine étape dans l'algorithme regroupe la sélection aléatoire d'une séquence et la recherche du BMU en utilisant la distance DTW. En ce qui concerne le calcul de la distance DTW, la méthode telle qu'elle est décrite dans la section 3.3.2.4 est reprise, puisque le codage utilisé est semblable à celui des données de l'installation.

Il est important de noter que tout calcul doit se faire sur la base du codage afin de réduire les temps de calcul. Revenir sur la forme décodée des séquences réduirait l'effet de compression du codage relatif au temps de calcul.

Pour cet exemple une pénalité $mph = mpv = 1$ est utilisée (3.19, 3.20) avec la forme modifiée de la distance DTW (3.17). Pour la distance locale la forme (3.18) est utilisée.

Après avoir parcouru tout le corpus des séquences, chaque séquence est attribuée à un prototype et la distance DTW ainsi que le cheminement pour chaque séquence sont connus. Le tableau 3.6 montre l'attribution des séquences pour les prototypes initiaux $\mathcal{S}_j(0) \quad \forall j = 1 \cdots 3$ avec la distance DTW entre séquence et prototype gagnant.

$M_1(0)$		$M_2(0)$						$M_3(0)$		
S_1 ;	S_5	S_2 ;	S_3 ;	S_4 ;	S_6 ;	S_7 ;	S_8	S_9 ;	S_{10} ;	S_{11}
0.78 ;	0.81	0.78 ;	0.78 ;	0.78 ;	0.89 ;	0.81 ;	0.89	0.76 ;	0.82 ;	0.77

TAB. 3.6 – Attribution des séquences aux prototypes initiaux avec leur distance DTW avec le prototype

La prochaine phase de l'algorithme est la phase d'apprentissage. Dans cette phase les prototypes sont modifiés afin de mieux représenter les séquences qui leur sont attribuées. Dans la loi d'apprentissage (3.28) le taux d'apprentissage $\alpha_r(t)$ est une fonction du temps c.-à-d. du cycle d'apprentissage. Pour cet exemple le taux d'apprentissage décroit de façon linéaire de 0.5 vers 0.01 avec le temps. 20 cycles d'apprentissage

(epochs) sont réalisés pour cet exemple.

Avant de pouvoir calculer les nouveaux prototypes, les séquences attribuées à chaque prototype sont mappées vers ce prototype. Pour cet exemple, la méthode décrite auparavant (chap. 3.3.3.2), avec enlèvement et dédoublement pour la compression respectivement l'extension, est utilisée. La figure 3.37 montre les séquences mappées $S'_{k,3}$ qui sont attribuées au prototype M_3 ainsi que le prototype. Notez que à l'origine les séquences S_9 et S_{11} sont des séquences à 10 triplets et que S_{10} est une séquence à 5 triplets, qui sont toutes mappées dans l'espace du prototype M_3 qui a une taille de 8 triplets.

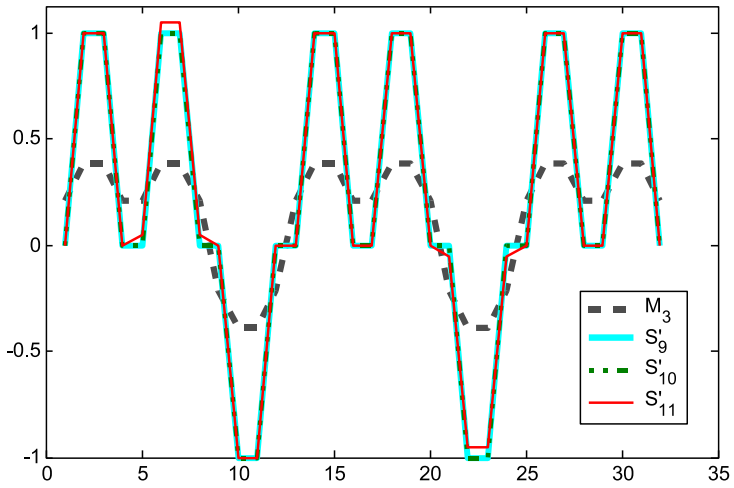


FIG. 3.37 – Séquences mappées, attribuées au prototype M_3 à l'instant $t = 0$

La séquence S'_{11} est la seule des exemples qui diverge légèrement des autres séquences mappées, parce qu'elle a de petites variations de la moyenne et de la variance à trois endroits comme décrit auparavant. A part cela le mappage réalise un alignement au prototype parfait.

Une moyenne de ces séquences mappées est utilisée, afin de calculer le prototype qui sera utilisé pour la prochaine itération. En ce qui concerne le calcul de la moyenne des séquences, il faut analyser le codage utilisé. Une réflexion semblable à celle utilisée dans le contexte de la distance locale (3.18) a mené vers l'approche suivante, qui est un compromis entre exactitude et temps de calcul. Pour chaque triplet $\vec{s}_{k,j}(i)$ des séquences mappées $S'_{k,j}$ la moyenne est calculée comme :

$$\text{mean}(\vec{s}_{k,j}(i)) = \begin{bmatrix} \text{round}(\text{mean}(C_{k,j}(i))) \\ \text{mean}(\mu_{k,j}(i)) \\ \text{mean}(\sigma_{k,j}(i)) \end{bmatrix} \tag{3.31}$$

Prendre l'arrondi de la moyenne des identifiants des formes de l'alphabet est explicable à cause du fait qu'il y a une topologie entre les formes qui vient de la carte SOM utilisée pour la création de l'alphabet. Puisque dans l'exemple uniquement deux formes existent, la même approche est possible et sera utilisée ici.

Cependant pour le calcul de la dissimilarité locale appliqué aux données de l'installation, qui sera analysé dans le chapitre 3.3.3.4, une distance euclidienne entre les formes primitives est utilisée. Une forme moyenne en utilisant les formes $C_{k,j}(i)$ est alors calculée. Afin d'obtenir de nouveau un identifiant pour le codage, une classification sur base de l'alphabet, comme durant le codage, est utilisée.

La loi d'apprentissage (3.28) est aussi une moyenne pondérée entre la séquence moyenne et le prototype précédent, ce qui est facile à implémenter par la même approche (3.31).

La figure 3.38 montre la moyenne $\bar{S}_{M_3}(0)$, le prototype initial $M_3(0)$ avec le nouveau prototype $M_3(1)$ après la première epoch. Dans la moyenne, la perturbation provenant de la séquence S_{11} est encore légèrement présente. Cette perturbation est aussi reprise par le prototype modifié. La même approche est réalisée pour les deux autres prototypes. Ces nouveaux prototypes $M_j(1)$ seront utilisés pour le calcul de l'attribution des séquences pour la deuxième itération.

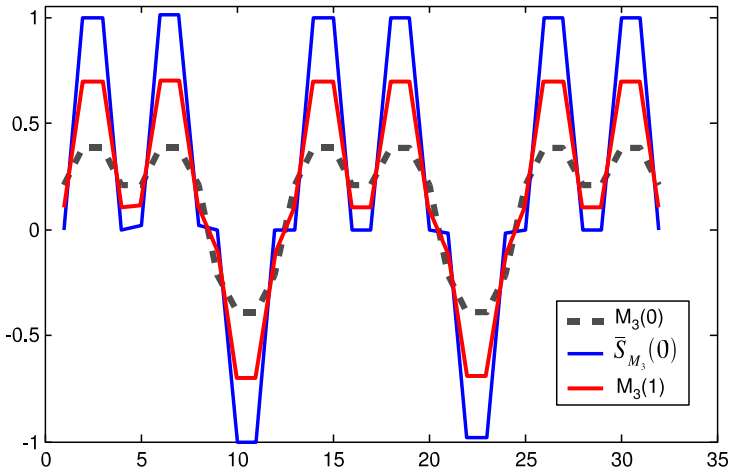


FIG. 3.38 – Exemple d'une séquence moyenne et de la modification du prototype

Pour cet exemple seulement 20 cycles d'apprentissage sont utilisés. Chaque cycle se déroule comme expliqué auparavant. Afin de pouvoir analyser l'apprentissage, la moyenne de l'erreur entre les séquences et le prototype attribué est calculée à chaque étape. La figure 3.39(a) montre le comportement de cette erreur au long de l'apprentissage. Le deuxième diagramme de la figure montre l'évolution de la variance des distances entre les séquences et leurs prototypes respectifs. Le tableau 3.39(b) montre l'attribution des séquences aux classes après la phase d'apprentissage. Ce tableau reprend aussi la distance DTW après l'apprentissage. L'attribution n'a pas changé depuis le début de l'apprentissage. Cela est explicable par le fait que pour cet exemple simple les prototypes étaient initialisés de façon à représenter les regroupements finaux.

Afin de montrer de quelle façon les prototypes ont été modifiés par l'apprentissage, la figure 3.40 montre les trois prototypes initiaux et finaux.

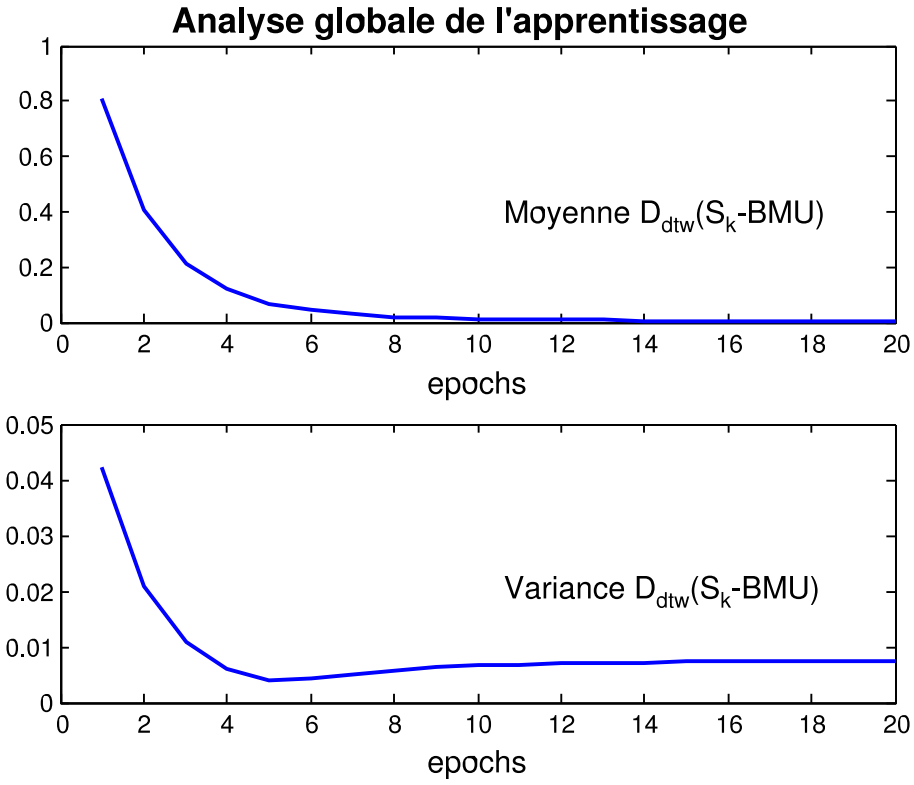
Cet exemple montre que la méthode est bien capable d'apprendre de façon non-supervisée le regroupement de séquences en se basant sur une métrique de similarité. Elle réalise donc bien une classification non-supervisée qui fournit en même temps des représentants des classes, car les prototypes sont similaires aux séquences de chaque classe. Le regroupement qui est réalisé correspond à la classification hiérarchique qui a été réalisée dans le chapitre 3.3.2.2, avec l'avantage de l'apprentissage concurrentiel et non-supervisé, qui se base sur des prototypes et l'avantage du temps de calcul, qui n'augmente que de façon linéaire avec le nombre de séquences dans le corpus d'apprentissage.

Cet exemple soulève cependant aussi quelques questions :

- La modification des prototypes qui est utilisée ici ne change pas la taille du prototype. Est-ce qu'il ne serait pas intéressant de pouvoir adapter de façon non-supervisée la taille des prototypes, afin qu'elle s'adapte au besoin. Car il n'est pas évident de bien initialiser les prototypes en forme et en taille.
- De plus, fixer le nombre de prototypes qui est à utiliser afin d'obtenir un classificateur performant n'est pas une tâche triviale.
- L'approche de mappage qui consiste à couper ou dédoubler des points de la séquence pourrait être modifiée afin de garder une trace de la partie enlevée ou de lisser la partie ajoutée suivant l'entourage.

Ces points n'ont pas pu être abordés concrètement dans le cadre de cette étude, mais sont des sujets importants pour des études futures.

Étant donné que dans cet exemple le codage est similaire au codage adapté qui est utilisé pour les séquences des cycles de productions, la méthode est directement utilisable sur les données de l'installation industrielle.



(a)

$M_1(20)$		$M_2(20)$						$M_3(20)$		
S_1 ;	S_5	S_2 ;	S_3 ;	S_4 ;	S_6 ;	S_7 ;	S_8	S_9 ;	S_{10} ;	S_{11}
1.45 ;	1.49	1.45 ;	1.45 ;	1.45 ;	1.65 ;	1.49 ;	1.65	5.76 ;	12.69 ;	25.98

(b)

FIG. 3.39 – Le résultat de l'apprentissage des prototypes :
(a) Évolution de l'erreur moyenne durant l'apprentissage ;
(b) L'attribution des séquences aux prototypes après l'apprentissage (classification). La distance DTW est $x * 10^{-3}$

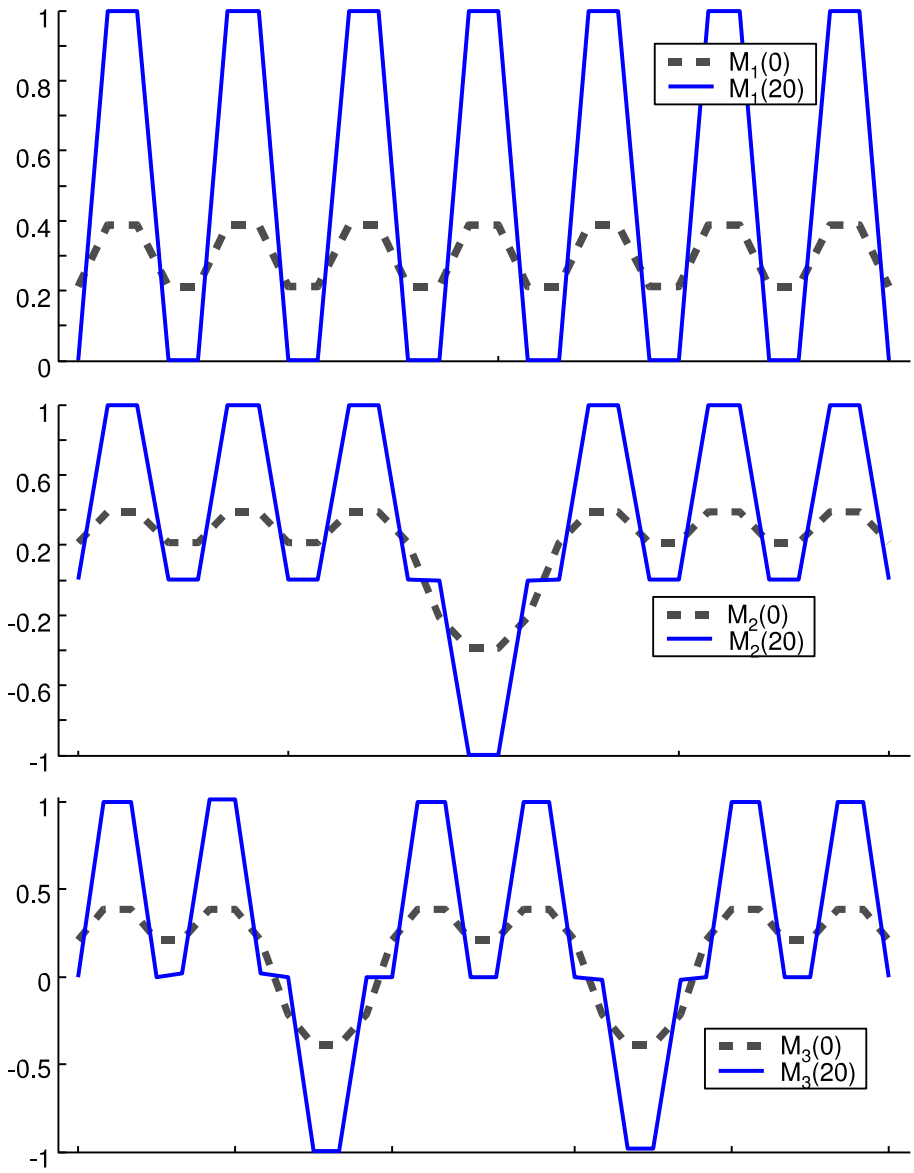


FIG. 3.40 – Comparaison des prototypes initiaux aux prototypes finaux.

3.3.3.4 Classification non-supervisée appliquée au processus industriel

La méthode de classification non-supervisée a donné un résultat satisfaisant sur un corpus de test, il est de ce fait relativement probable que la méthode soit capable de produire un résultat pour les données du four, car en plus les codages utilisés sont similaires. Mais comment peut-on être sûr que le résultat qui sera produit aura une redevance réaliste ?

Afin de pouvoir valider la classification, qui dans ce cas se base sur une mesure de similarité, l'idée était de demander les avis à des groupes d'experts humains, qui connaissent plus ou moins bien l'installation et le processus. Faire analyser par des experts le résultat d'une classification non-supervisée, établi par la méthode proposée, ne nous semblait pas approprié, pour des raisons d'objectivité. Les groupes d'experts ne doivent pas être influencés par un résultat qui leur est proposé afin d'éviter une validation biaisée. C'est pourquoi, une autre approche, qui nous a semblé plus objective, a été employée.

Chaque groupe d'experts a manuellement regroupé des cycles d'évolution. Pour avoir des conditions relativement équivalentes sans trop user du temps des experts, un corpus de références comprenant 20 cycles de productions sélectionnés aléatoirement a été construit. Afin de représenter l'évolution du processus de façon comparable à celle employée pour le classificateur artificiel, le signal de la puissance électrique de chaque cycle a été imprimé sur des cartes (une carte représentant un cycle) et un numéro de référence allant de 1 à 20 leur a été attribué.

Basé sur la réprésentation graphique, deux groupes d'experts ont été invités à regrouper les cycles dans quatre catégories. Le premier groupe était un groupe de deux personnes de ProfilARBED Recherche qui connaissent bien le processus et qui ont travaillé sur le projet CECA. Le deuxième groupe se composait de deux personnes du CRP Henri Tudor impliquées dans le projet.

Finalement la même tâche a été attribuée à la méthode de classification non-supervisée basée sur la distance DTW (UdtwC, angl. Unsupervised DTW-based Clustering). Pour cet exemple l'alphabet A2 a été utilisé. Cet alphabet représente un compromis entre la qualité de reconstruction des signaux basés sur le codage et le temps de calcul pour la distance DTW qui augmente avec le nombre de triplets dans le codage. L'initialisation des prototypes d'évolutions a été faite en utilisant quatre cycles d'évolution du corpus de référence (1,2,4,20). Ils ont été sélectionnés de façon à assurer une certaine variance dans les évolutions des prototypes. Le corpus d'apprentissage était représenté par 150 cycles utilisables de la base de données et comprenant les 20 cycles de référence.

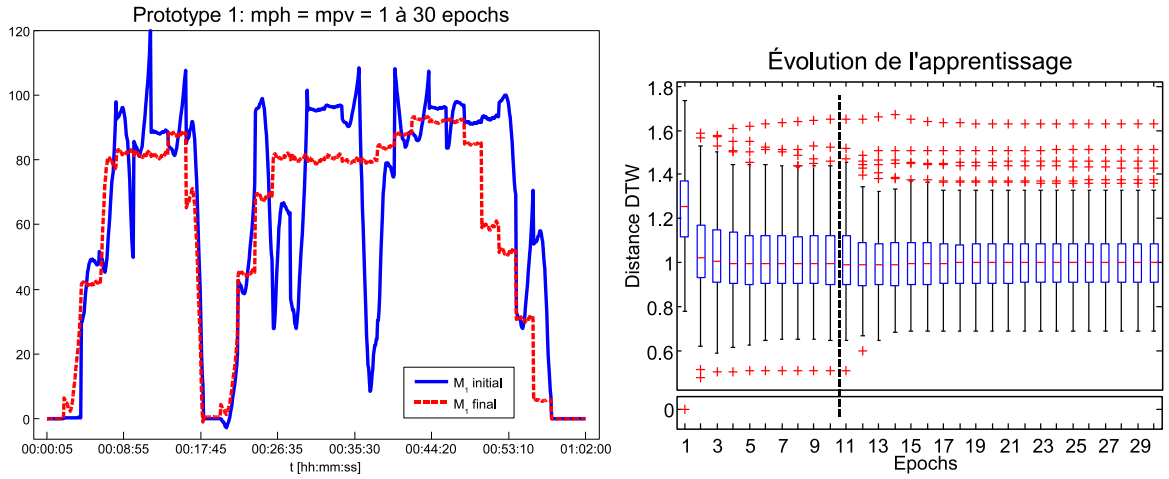
L'apprentissage a été réalisé en deux phases :

- une première phase avec un taux d'apprentissage élevé (allant de 0.9 à 0.2 de façon exponentielle) sur 10 epochs. Durant cette phase, la moitié (les cycles les plus proches du prototype) des cycles attribués à un prototype a été utilisée pour l'adaptation du prototype. Une pénalité symétrique (mph = mpv = 1) a été utilisée. Cette phase est utilisée afin de positionner les prototypes sans être influencé par des cycles outlier.
- une deuxième phase avec un taux plus faible (allant de 0.6 à 0.05 de façon exponentielle) sur 20 epochs. Durant cette phase, tous les cycles attribués à un prototype sont utilisés pour l'apprentissage. La pénalité n'a pas été changée (mph = mpv = 1).

La figure 3.41(a) montre le prototype M1 dans sa forme initiale et après les deux phases d'apprentissage. La phase de fusion du deuxième panier est fortement modifiée après l'apprentissage. Localement, d'importantes modifications sont visibles. D'autres parties, comme la fusion du premier panier (premier créneau) et la partie de chargement du deuxième panier ainsi que le début de la deuxième phase de fusion, ne montrent que de légères modifications. Par contre, la coupure de courant durant la fusion du deuxième panier est presque complètement enlevée.

La figure 3.41(b) représente l'évolution de l'apprentissage sur les 30 epochs. Pour chaque epoch un diagramme "boîte à moustache" montre la distribution des distances DTW entre les signaux et les prototypes attribués respectifs (distances intra classes toutes classes confondues). Le trait vertical entre l'epoch 10 et 11 indique graphiquement le changement entre les deux phases de l'apprentissage.

Le trait rouge à l'intérieur de chaque boîte indique la médiane de la distribution. La diminution de cette valeur avec l'avancement de l'apprentissage illustre que les prototypes sont modifiés de manière à mieux représenter le corpus d'apprentissage. La taille des moustaches (traits noirs verticaux partant de chaque boîte, limités par un petit trait horizontal) est un indicateur de la dispersion des classes. Avec l'avancement de l'apprentissage cette dispersion est réduite, et cela spécialement durant la deuxième



(a) (b)

FIG. 3.41 – Résultat de l'apprentissage des prototypes sur les données du four :
(a) Exemple d'adaptation d'un prototype sur base du prototype M1 après 30 epochs d'apprentissage ;
(b) Évolution des distances DTW intra-classes, toutes classes confondues, représentée par des diagrammes "boîte à moustache" pour chaque epoch de l'apprentissage.

phase ce qui s'explique par le fait que, durant cette phase, tous les cycles attribués à un prototype sont considérés pour l'adaptation de ces derniers.

Les croix rouges à l'extérieur des limites des moustaches sont considérées comme outliers de la distribution des distances DTW. Pour la première epoch le diagramme montre une croix à la distance zéro, ce qui s'explique avec la sélection des prototypes étant membres du corpus d'apprentissage.

Le résultat de cette expérience est représenté dans un tableau comparatif (tab. 3.7) des regroupements, réalisés par les différents experts et par la méthode non-supervisée UdtwC basée sur la distance DTW et le codage par l'alphabet A2 des signaux. Afin de calculer le regroupement réalisé par cette méthode, une classification sur base des quatre prototypes obtenus auparavant est effectuée pour les 20 cycles de référence, en se basant sur la distance DTW. La classification UdtwC est représentée dans le tableau pour deux valeurs du paramètre mphv ($mph = mpv$) afin de montrer son influence sur le résultat de la classification.

Group	Classe 1	Classe 2	Classe 3	Classe 4
PARE	1 **6** **13** **20**	7 **9** **12** **15** **16**	2 **8** **10** **14** 17 **18** **19**	3 4 5 11
CRPHT	**6** **13** **20**	2 3 **9** 11 **12** **15** **16** 17	4 5 **8** **10** **14** **18** **19**	1 7
UdtwC mphv=0.1	**6** **13** 16 **20**	2 7 **9**	1 3 **8** 11 **14** 15 17 **18** **19**	4 5 10 12
UdtwC mphv=1.4	**13** **20**	3 4 5 **9** 10 **12**	2 **8** 11 **14** 16 **18** **19**	1 6 7 15 17

TAB. 3.7 – Comparaison des résultats obtenus par les quatre différentes catégorisations

Les cycles en **gras** sont classés de la même façon par les deux groupes d'experts et sont par après aussi mis en évidence pour les modèles UdtwC. Les cycles soulignés sont regroupés de façon identique pour le groupe ProfilARBED Recherche et un des modèles UdtwC, et finalement les numéraux encadrés sont identiques pour le groupe CRP Henri Tudor et un des modèles UdtwC.

Pour la "Classe 1" la situation semble relativement claire pour les cycles 13 et 20 et notamment le cycle 6, même si pour UdtwC avec $mphv = 1.4$ ce cycle ne se retrouve plus dans cette classe. Aussi pour

la "Classe 3" un certain nombre de cycles s'y retrouvent dans tout les cas (8, 14, 18 et 19). Pour la "Classe 2" une similarité est observable au moins entre les deux groupes d'experts. La méthode UdtwC ne réussit pas à regrouper ces cycles. Pour la "Classe 4" aucune correspondance entre les deux groupes d'experts n'est visible. Remarquer cependant que pour $mphv = 0.1$ davantage de similarités avec le groupe PARE est visible, et que avec $mphv = 1.4$ la correspondance avec le groupe CRPHT est supérieure. Cela souligne le fait que ce paramètre pourra être utilisé afin d'adapter la méthode à la subjectivité de l'expert humain.

Dans ce contexte il faut remarquer que un seul test a été réalisé, et que pour la méthode UdtwC l'initialisation des prototypes pour l'apprentissage joue un rôle important. Plusieurs essais avec différentes initialisations auraient éventuellement pu améliorer le résultat. Mais n'oublions pas que sélectionner 20 cycles parmi 2471 ne correspond pas vraiment à une bonne représentativité. Il est donc fortement possible que les 150 cycles utilisés pour l'apprentissage ne reflètent pas la distribution de ces 20 cycles de référence.

De plus, il a fallu de nombreuses discussions, entre les deux personnes du groupe CRP Henri Tudor, avant de décider quels cycles vont être mis ensemble afin d'aboutir à quatre catégories.

Le résultat reflète la problématique de la subjectivité en relation avec la similarité, ce qui a déjà été soulevé antérieurement. De plus, les experts de ProfilARBED Recherche ont fait remarquer qu'il n'est pas évident d'analyser un cycle en se basant seulement sur la puissance électrique. Il y a des phénomènes qui ne sont pas directement visibles dans le signal de la puissance électrique, mais qui ont une influence sur le comportement du processus. Ils sont représentés par d'autres mesures. Les experts ont supposé qu'un résultat moins ambigu pourrait être espéré, si ces informations étaient ajoutées à la représentation des cycles de production.

Avec la classification non-supervisée des cycles de production basée sur la similarité des évolutions, la phase de l'analyse (fig. 1.10) est terminée. Pour la suite de cette étude, un certain nombre de prototypes d'évolutions utilisant différents alphabets est nécessaire.

3.3.3.5 Création de prototypes d'évolution pour le processus industriel

Afin de produire des prototypes d'évolution pour le processus industriel, la classification UdtwC, qui était employée pour la création des classes du corpus de références, est appliquée à un ensemble représentatif de cycles de production, qui est codé dans les quatre alphabets de référence (voir tab.3.3).

Un ensemble de 16 prototypes est construit. La production de ce pool de prototypes a été réalisée en cinq phases :

- Dans la phase d'initialisation les quatre alphabets sont traités indépendants les uns des autres. La procédure est la même pour chaque alphabet. La boucle d'apprentissage commence par une sélection aléatoire de neuf cycles de production de la base de données. Sur base d'un corpus d'apprentissage réduit (50 cycles de productions) un apprentissage court (10 epochs) est réalisé avec un taux d'apprentissage allant de 0.8 à 0.3, et une pénalité symétrique de 0.9.
  La qualité du clustering est mesurée sur base de l'erreur moyenne du corpus après l'apprentissage. La boucle est répétée avec une nouvelle sélection aléatoire de neuf cycles, jusqu'à ce que, pendant la durée de 10 boucles, plus aucune amélioration n'a été constatée. Les quatre meilleures réalisations sont gardées pour chaque alphabet.
- La deuxième phase consiste à analyser et à sélectionner les huit meilleurs prototypes pour chaque alphabets. Les critères de sélections sont : le taux d'utilisation des prototypes pour le corpus d'apprentissage et la similarité entre les prototypes, en utilisant la distance DTW et la classification hiérarchique (fig. 3.27). Il a été essayé que les prototypes sélectionnés soient représentatifs et qu'en même temps ils couvrent une grande diversité de comportements du processus.
- Durant la troisième phase les prototypes sélectionnés subissent un deuxième apprentissage avec un corpus plus important (300 cycles) et une durée d'apprentissage plus longue (20 epochs). Le taux d'apprentissage va de 0.7 à 0.3, et la pénalité reste inchangée à 0.9. De nouveau chaque alphabet est traité indépendamment des autres.
- Après cet apprentissage, une dernière sélection des prototypes est faite, en ne gardant que quatre prototypes par alphabet, ce qui fait en tout un corpus de 16 prototypes. Les critères de sélection sont les mêmes que précédemment.
- Dans la dernière phase un troisième apprentissage est réalisé, en utilisant un corpus d'apprentissage de 250 cycles et une durée de 30 epochs. Le taux d'apprentissage va de 0.4 à 0.01, et une pénalité

symétrique de 1.1 est utilisée. Ici les prototypes des quatre alphabets sont appris ensemble, ce qui fait qu'une certaine situation de concurrence entre les alphabets est réalisée. Il faut noter, que durant cette phase les prototypes basés sur l'alphabet A1 ne sont pas sélectionnés comme BMU au même taux que les prototypes issus des autres alphabets.

La figure 3.42 montre les 16 prototypes qui vont être utilisés dans la suite de l'étude.

3.4 Conclusion du chapitre

Dans ce chapitre l'analyse de séries temporelles est abordée dans le contexte d'une approche précise, à savoir l'analyse des évolutions temporelles du processus industriel présenté dans le chapitre 1. La finalité de l'analyse est la production d'un classificateur qui est utilisable pour la deuxième étape de l'approche globale, le contrôle prédictif basé sur une aide à la décision.

Après un aperçu sur les spécificités du traitement de séries temporelles, un nouveau codage, appelé codage adapté, a été présenté. Celui-ci se base sur une analyse des formes primitives qui composent le signal analysé. Une approche connexionniste non-supervisée (SOM) est utilisée pour extraire ces formes primitives d'un corpus représentatif du signal en question. Par cette méthode cinq alphabets ont été produits, dont le premier avec le but d'être simple, afin de pouvoir l'utiliser pour l'explication de la méthode et la discussion des résultats. Les quatre autres alphabets, qui représentent différentes tailles et nombre de formes, sont utilisés pour le codage du corpus entier de cycles dans la base de données du projet. Le codage a été comparé à trois autres méthodes, à savoir la moyenne flottante (qui est une approche de filtrage), l'approximation par sections constantes (PAA) et l'approximation par sections linéaires (PLA)(qui sont des méthodes de codages).

Sans avoir l'ambition d'être exhaustif il a été montré, sur base d'exemples, que le codage adapté contient dans son code les deux méthodes (PAA, PLA) et que la qualité de l'approximation réalisée par le codage adapté dépasse celle de ces deux méthodes.

La deuxième partie du chapitre traite la question du regroupement des cycles de production selon leur évolution temporelle, en se basant sur le codage adapté. Dans cette partie les sujets principaux sont la similarité et les méthodes exploitables qui peuvent regrouper des cycles de production dont l'évolution est semblable. En bénéficiant des propriétés de la méthode DTW (Dynamic Time Warping) il a été montré qu'une classification non-supervisée, basée sur un apprentissage concurrentiel, est réalisable. Les propriétés de la méthode DTW sont : l'alignement non-linéaire sur l'axe du temps, la possibilité de paramétriser la distance DTW (afin de correspondre à la subjectivité individuelle selon l'application) et le mappage DTW (par lequel il est possible de transférer des séquences d'un espace temporel vers un autre). Une méthode d'apprentissage concurrentiel simple a été employée pour assurer la maîtrise de l'ensemble. Il est cependant facilement possible d'utiliser un autre algorithme qui va probablement améliorer l'ensemble de cette classification.

Basée sur un environnement artificiel, cette classification, appelée UdtwC (Unsupervised DTW-based Clustering), a été analysée, et une validation qualitative a été présentée.

Basée sur une expérience utilisant un corpus de référence de cycles de production, la méthode UdtwC a été comparée à deux résultats de classifications manuelles réalisées indépendamment par deux groupes d'experts. Le résultat de cette comparaison satisfait nos espérances.

Finalement un corpus de 16 prototypes d'évolution, qui va être utilisé dans la suite de l'étude, a été produit.

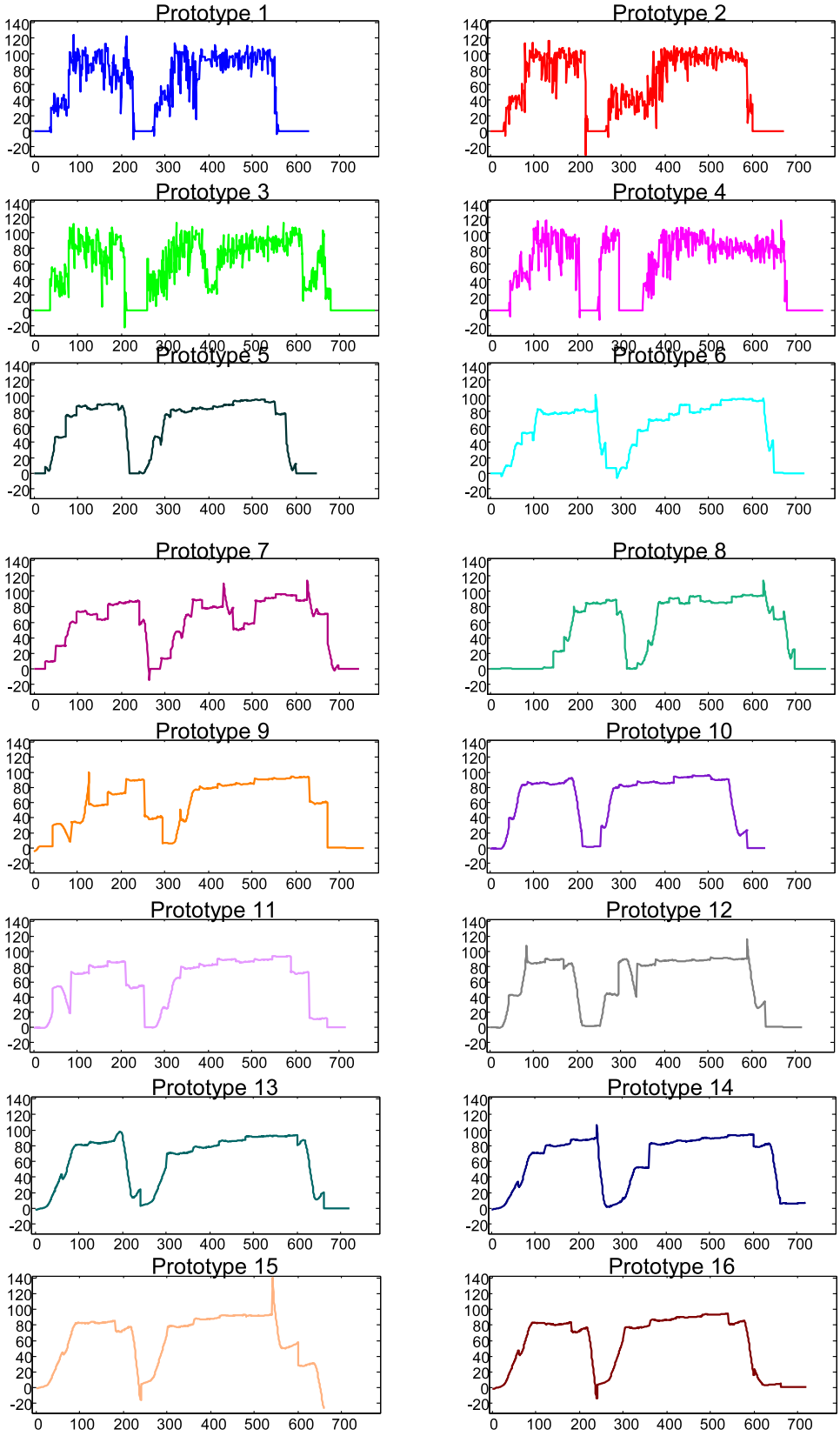


FIG. 3.42 – Les 16 prototypes d'évolutions

Chapitre 4

Un contrôle prédictif basé sur une aide à la décision

Sommaire

Un des buts du travail est d'analyser les possibilités afin d'utiliser les connaissances extraites par des méthodes de fouille de données temporelles, dans l'optique d'améliorer la conduite de l'installation et par ce biais d'aider l'industriel dans la démarche de réduction du coûts de production. Une hypothèse posée dans ce travail et soutenue par les experts consiste à dire qu'il existe une relation entre le comportement temporel de l'installation et sa consommation en énergie. Ce qui revient à dire qu'il serait possible de garantir une faible consommation, si l'on arrivait à manipuler le processus de telle manière que son évolution corresponde au mieux à une évolution qui est connue pour sa faible consommation.

La plupart des approches de contrôle se basent sur un modèle plus ou moins concret du processus. Sans aller dans les détails, ces approches essayent en gros de trouver les paramètres du modèle qui minimisent une certaine fonction de coût, en se basant sur des données provenant de l'installation. Une fois ces paramètres trouvés, le processus est contrôlé selon ce modèle.

Dans le projet CECA un des partenaires travaille sur une telle méthode avec un modèle du processus. Ce modèle prend en compte des phénomènes mécaniques, chimiques, physiques et thermodynamiques. La complexité de ce modèle est relativement importante et de ce fait il est difficile de le paramétriser. Souvent les installations industrielles ne fournissent pas les données nécessaires pour pouvoir fixer correctement les paramètres d'un tel modèle. Des données fortement bruitées voire même erronnées n'améliorent pas le processus de paramétrisation.

L'approche de contrôle proposée ici a brièvement été introduite dans le chapitre 1.4 et la figure 4.1 reprend le concept.

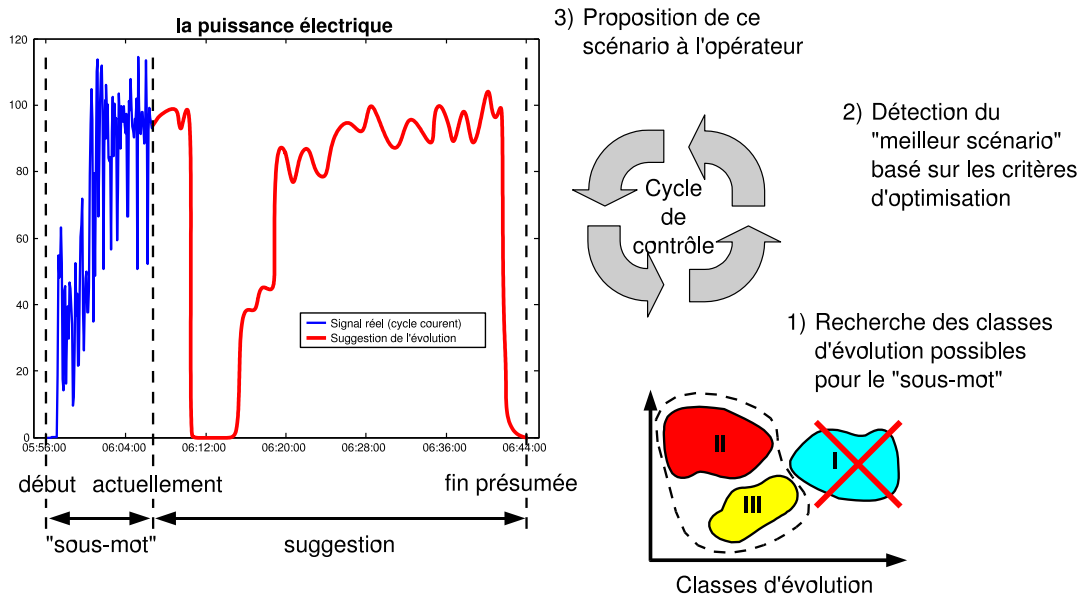


FIG. 4.1 – Le contrôle prédictif en ligne

Cette approche se base sur une analyse du comportement temporel et la relation entre ce comportement et les consommations. L'analyse du comportement temporel du processus est en premier lieu découplée de la consommation. Cette analyse a été décrite dans le chapitre précédent (chap. 3) et a produit de façon non-supervisée des classes d'évolutions représentées par des prototypes de comportement du processus analysé (voir fig. 3.42). Dans ce chapitre des méthodes de couplage seront proposées, avec lesquelles il sera possible de considérer des critères d'optimisation, afin de pouvoir proposer à l'industriel des scénarios avec un potentiel d'amélioration de la consommation globale.

A chaque instant de l'évolution d'un cycle de production une ou plusieurs propositions peuvent être fournies à l'opérateur. Ces propositions peuvent être vues comme une aide à la décision pour l'opérateur qui, en manipulant les paramètres du processus, ferme la boucle de contrôle.

Ce chapitre va aborder les différentes phases de cette approche de contrôle prédictif illustrées dans la figure 4.1 [53]. La première phase consiste dans la recherche des évolutions types similaires, en se basant

sur un début de séquence, c.-à-d. la partie connue d'un cycle de production en cours. Cette phase est suivie de l'élaboration de méthodes pour la sélection du "meilleur scénario". Ces méthodes sont basées sur la relation qui existe entre les évolutions types et les critères qu'il s'agit d'optimiser. Puisque une réelle implémentation des méthodes sur l'installation industrielle n'était pas possible dans le cadre du projet CECA, cette analyse se limite à analyser des possibilités théoriques d'une telle approche. La dernière phase consiste à établir plusieurs propositions alternatives qui pourraient être données à l'opérateur afin de le guider dans ses décisions. Le chapitre finit avec une discussion de cette méthode dans le contexte du contrôle prédictif, où en réalité l'opérateur humain interviendrait pour fermer la boucle.

Comme dans le chapitre précédent les démarches sont accompagnées d'exemples artificiels, afin de mieux illustrer les méthodes et les résultats obtenus, avant d'appliquer les méthodes aux données du processus industriel.

4.1 Classification d'évolutions en ligne

La première phase du contrôle prédictif consiste à rechercher dans les évolutions types celles qui ressemblent le plus au début de séquence qui décrit l'évolution du cycle en cours. Cela correspond, dans le domaine d'analyses de séries temporelles, à la localisation de sous-séquences. Le but de la localisation est de trouver dans une séquence donnée S la ou les parties P_i de cette séquence qui ressemblent le plus à une sous-séquence Q. Dans le contexte de ce travail, la similarité joue de nouveau un rôle important dans cette localisation. Ce qui veut dire que ni la taille ni la forme des parties P_j ne sont nécessairement identiques à Q. Une certaine souplesse dans les deux propriétés taille et forme est nécessaire.

Pour cela, la métrique pour mesurer la dissimilarité utilisée sera la distance DTW qui a été introduite dans le chapitre 3.3.2.

Une propriété spécifique au contexte de ce travail est que le début de la partie P_j recherchée doit s'aligner avec le début de la séquence S, ce qui simplifie considérablement la tâche. Ceci vient du fait que la séquence S représente une évolution type, c.-à-d. un cycle de production complet, et que la sous-séquence Q décrit le début d'un cycle de production.

Afin de pouvoir profiter de la propriété de réduction de données offertes par le codage adapté, tous les calculs doivent se faire sur les séquences (respectivement sous-séquences) codées. Le signal provenant de l'installation est donc en premier lieu codé au fur et à mesure de l'avancement du processus, en utilisant la méthode présentée dans la section 3.2.

Dans la suite de cette section, la problématique et la méthode de base pour la localisation sont expliquées. Sur base d'un corpus de séquences artificielles, les propriétés de cette méthode sont analysées.

4.1.1 Localisation de sous-séquences basées sur la similitude

Dans la partie de classification non-supervisée (chap. 3.3.3) la distance DTW est utilisée comme mesure de dissimilarité. L'utilisation de cette mesure pour la localisation est analysée en raison des expériences positives en relation avec la similarité, et afin d'éviter le plus possible de devoir maintenir et maîtriser trop de méthodes différentes. Les détails du calcul DTW sont donnés dans le chapitre 3.3.2.

Une des conditions de base de la méthode DTW est que les deux séquences sont comparées dans leur totalité. En d'autres mots, aussi bien le début (3.14) que la fin (3.15) des séquences doivent s'aligner (voir aussi forme récursive du DTW (3.12)).

Dans le contexte de la localisation, cette condition exige un regard plus détaillé de la méthode, car le but de la localisation de sous-séquences est spécialement de trouver une partie à l'intérieur d'une séquence qui est similaire à la totalité d'une autre séquence (sous-séquence) (chap. 3.1.3). Afin d'illustrer la problématique dans le contexte de cette étude, deux séquences test relativement simples sont données et analysées.

Soit $Q(t)$ la sous-séquence (4.1) de taille $M = 11$ et $S(t)$ la séquence (4.2) de taille $N = 29$.

$$Q(t) = \{0,0,0,0,0,1,1,1,0,0,0\} \tag{4.1}$$

$$S(t) = \{0,0,1,1,1,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0\} \tag{4.2}$$

Leur représentation graphique (fig. 4.2) montre qu'entre les deux séquences il existe une différence de taille et de forme importante. Revenons au contexte de l'étude et comparons la sous-séquence $Q(t)$ avec le début de la séquence $S(t)$. En prenant en compte l'aspect de similarité, la différence n'est pas très importante, elle se limite aux trois pas de temps pendant lesquels le palier à valeur nulle au début de la sous-séquence $Q(t)$ est plus long que celui de $S(t)$. Le créneau au début de $Q(t)$ a la même taille que celui du début de $S(t)$. Remarquez que tous les créneaux des $S(t)$ ont la même taille, ce qui va influencer le calcul de similarité.

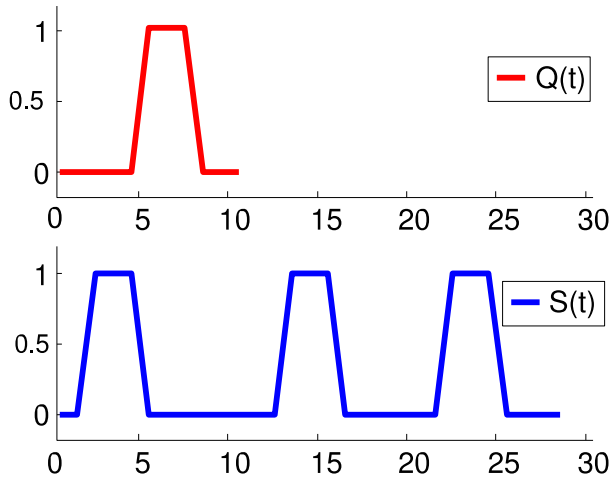


FIG. 4.2 – Les deux séquences test $S(t)$ et $Q(t)$ pour analyser la localisation DTW

Appliquons maintenant le calcul DTW entre les deux séquences et analysons le résultat de l'alignement qui est réalisé (fig. 4.3).

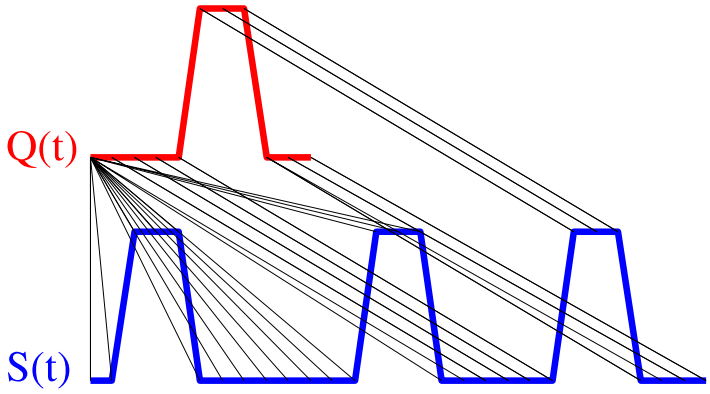


FIG. 4.3 – Résultat de la localisation en utilisant la méthode DTW de base en représentant l'alignement réalisé

Rappelons que le calcul DTW cherche l'alignement qui minimise la distance globale cumulée entre les deux séquences en alignant chaque fois le début et la fin. En prenant la séquence $S(t)$ comme référence, la méthode doit de ce fait étirer la sous-séquence $Q(t)$ à la taille de $S(t)$ en ajoutant des points à $Q(t)$ aux endroits les moins coûteux pour l'ensemble. Si $Q(t)$ est pris comme séquence de référence pour la description, ce qui peut paraître étrange, mais ce qui peut simplifier la compréhension, c'est $S(t)$ qui est comprimée à la taille de $Q(t)$ en coupant des parties de $S(t)$ aux endroits les moins coûteux pour l'ensemble.

Les paramètres utilisés pour le DTW sont les suivants : une métrique euclidienne pour de la distance

locale, une normalisation par la diagonale est réalisée, et une pénalité symétrique $mph = mpv = 0.7$.

En l'occurrence, dans l'exemple, la première partie de $S(t)$ est comprimée sur un point. Considérant la similarité et la comparaison manuelle, plusieurs alignements qui donnent le même résultat sont possibles, puisque les créneaux de $S(t)$ ont tous la même taille et sont donc similaires à $Q(t)$. L'implémentation de l'algorithme DTW qui est utilisée fait en sorte que l'alignement est construit par rétroaction en venant des points finaux des deux séquences. La figure 4.4 montre la matrice des distances cumulées (GDM) pour ce calcul avec le chemin qui est utilisé. La même GDM montre aussi d'autres vallées qui suffiraient au cheminement de la similarité.

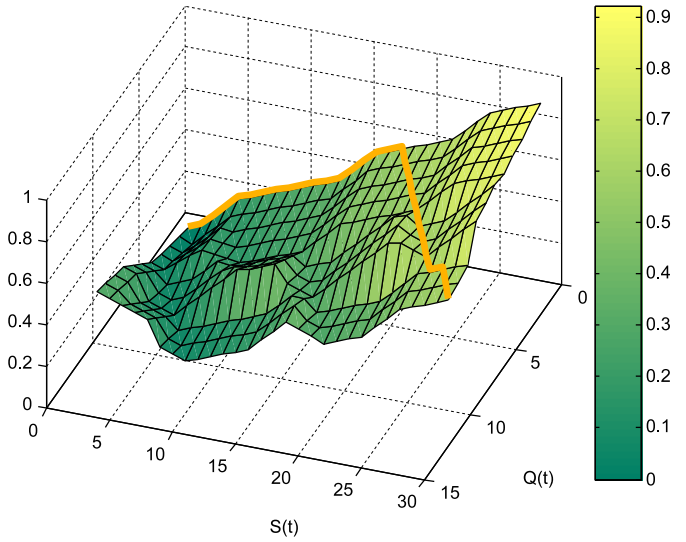


FIG. 4.4 – La matrice de distance globale GDM en illustration 3D

Puisque la méthode DTW n'est donc pas directement utilisable pour la localisation, une première approche plutôt pragmatique serait de comparer la sous-séquence $Q(t)$ avec toutes les parties $P_j(t)$ possibles de la séquence $S(t)$ et de sélectionner celle qui produit la distance minimale. Cette approche est très coûteuse, puisque déjà le calcul DTW en soi nécessite d'importantes ressources.

Dans le contexte de ce travail où une comparaison est faite entre le début d'un cycle de production et les évolutions prototypes, il est possible de se limiter aux parties qui commencent au début de la séquence. Une localisation globale partout dans la séquence n'est donc pas nécessaire, ce qui limite le nombre de calculs DTW à réaliser à $N - 1$ avec N étant la taille de la séquence $S(t)$.

Mais, analysons plus en détail les calculs du DTW. Après la construction de la matrice des distances locales (LDM), la matrice des distances cumulées est construite en calculant itérativement pour toute position i, j de la matrice la distance cumulée pour ce point $D_{dtw}(S(i), Q(j))$ (3.12). Cette formulation n'admet que trois directions afin d'arriver au point i, j : les chemins diagonal, horizontal et vertical. Cette contrainte de base de la méthode DTW assure que pour chaque point i, j seule la sous-matrice $[0, 0 - i, j]$ est prise en compte. Cela veut dire que pour chaque position i, j de la matrice, il est possible d'établir le cheminement optimal afin d'arriver à ce point en utilisant la rétro-propagation des choix locaux du cheminement.

Considérons le contexte de la localisation telle qu'elle est comprise dans ce travail, où chaque partie $P_i(t)$ de $S(t)$ doit débuter au point initial de $S(t)$ et où la sous-séquence $Q(t)$ est considérée dans son ensemble. Il suffit de ce fait de calculer la distance DTW entre les deux séquences $S(t)$ et $Q(t)$. La dernière colonne de la GDM pour $j = M$, avec M étant la taille de $Q(t)$, contient donc les distances DTW de toutes les parties $P_i(t)$ commençant à la racine de $S(t)$ (fig. 4.5).

Il faut noter que la normalisation de la distance DTW (chap. 3.3.2.2), qui est faite pour la position $[N, M]$, fausserait le résultat, puisque dépendant de la méthode de normalisation, la valeur de normalisation est différente pour chaque $P_i(t)$. La valeur des distances DTW dans la figure 4.5 en tient compte, car la normalisation est adaptée à chaque $P_i(t)$.

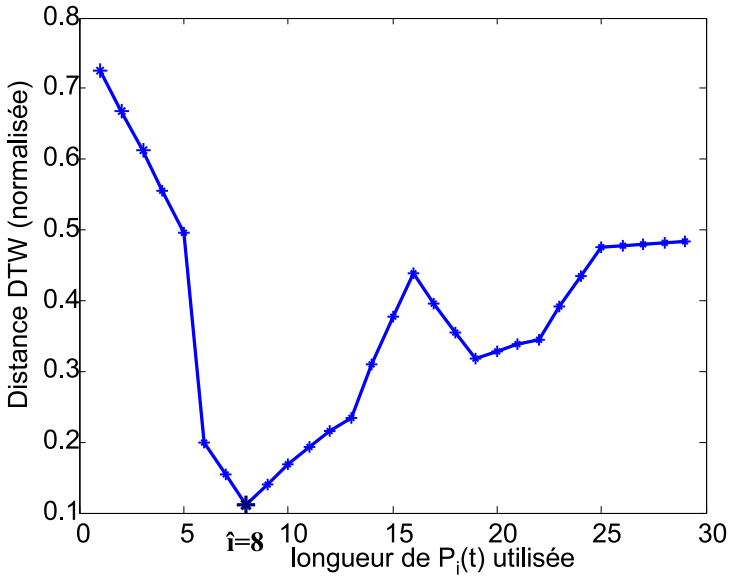
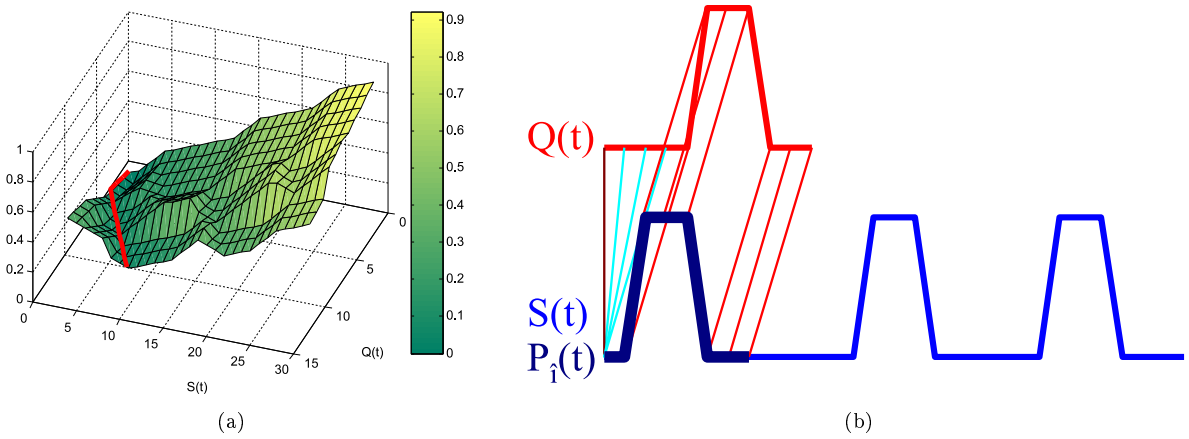


FIG. 4.5 – Dernière colonne de GDM ($j = M$) : $D_{dtw}(P_i(t), Q(t))\forall i = 1 \cdots N$

Le minimum de cette courbe ($i = \hat{\imath}$) désigne la partie $P_{\hat{\imath}}(t)$ de la séquence $S(t)$ qui minimise la distance DTW et de ce fait représente la partie commençant à la racine de $S(t)$ la plus semblable à la sous-séquence $Q(t)$.

En partant de cette position $[\hat{\imath}, M]$ pour la construction du cheminement, le chemin optimal qui spécifie l'alignement entre la sous-séquence et la partie la plus similaire peut être construit (fig. 4.6).



(a) (b)

FIG. 4.6 – Résultat $P_{\hat{\imath}}(t)$ de la recherche de la partie la plus similaire à $Q(t)$ qui commence à la racine de la séquence $S(t)$: (a) Cheminement de la localisation dans la GDM (3D) ; (b) alignement de la localisation

La méthode présentée ici, qui sera appelée la "localisation DTW-LB[13]", retrouve donc bien la partie d'une séquence, commençant à la racine, qui est similaire, au sens de la distance DTW, à une sous-séquence donnée. De plus la méthode ne nécessite presque aucun calcul supplémentaire à celui du calcul DTW entre la sous-séquence et la séquence complète. Afin d'analyser la méthode plus en détail, la section suivante va introduire un environnement de test avec des séquences artificielles.

[13] DTW bornée à gauche ; Angl. : Dynamic Time Warping Left Bounded

4.1.2 L'environnement artificiel de test

Un environnement artificiel en vue d'analyser une méthode a d'un côté l'avantage que cet environnement est maîtrisé, ce qui est important pour la validation d'une méthode. D'un autre côté, il est important que cet environnement reflète le plus possible la situation réelle, afin de valider la méthode dans le bon contexte. C'est la raison pour laquelle l'environnement de test qui est construit utilise le même type de codage qui est utilisé pour les données du processus industriel (une suite de triplets comprenant une forme issue d'un alphabet, une moyenne et une variance). La méthode est donc comparable à celle utilisée précédemment (chap. 3.3.3.3) sauf que cette fois-ci, un certain nombre de paramètres, comme la taille de séquences, les moyennes et variances des triplets, sont fixés de façon aléatoire.

Deux corpus seront construits : un premier qui représente les évolutions types (prototypes) et un deuxième qui représente les cycles de production. Puisqu'il est question de séquences artificielles qui ne se basent pas sur un processus physique réel, la terminologie utilisée dans ce chapitre sera : les "séquences types" pour les prototypes, et les "séquences d'évolution" pour les cycles de production.

La construction des séquences d'évolution est faite en utilisant des parties des séquences types afin d'assurer une relation entre elles. Le but du test est d'analyser la méthode de localisation DTW-LB dans sa capacité à détecter les séquences types espérées.

Le but de cet environnement de test n'est pas de produire une validation quantitative (statistique) de la méthode, mais d'analyser le comportement qualitatif dans des situations bien précises.

4.1.2.1 L'alphabet artificiel utilisé

Afin de garder aussi simple que possible l'environnement de test, l'alphabet ne comporte que trois formes linéaires simples (fig. 4.7).

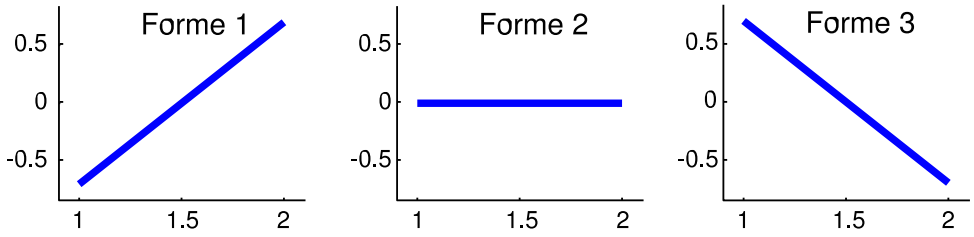


FIG. 4.7 – Les trois formes qui composent l'alphabet de l'environnement de test

La suite des formes vérifie une certaine topologie, c.-à-d. une relation hiérarchique basée sur la distance entre les formes, qui existe aussi dans les alphabets qui sont extraits par une SOM des données de l'installation $d(F1, F2) \leq d(F1, F3) \geq d(F2, F3)$. Les formes sont normalisées, ce qui veut dire que la moyenne est nulle et que la variance a la valeur "un" (sauf pour la forme 2).

Notez que les formes ont une taille (longueur) de deux, ce qui représente deux points sur l'axe des temps. Une séquence codée par 3 triplets par exemple représente dans la forme décodée une suite de six points sur l'axe des temps. Ceci est important pour la suite des analyses.

4.1.2.2 Les séquences types

La construction des séquences types repose sur les paramètres aléatoires suivants :

1. la longueur de la séquence type : c.-à-d. le nombre de triplets utilisés en limitant la variation entre 15 et 24.
2. l'ordre des formes utilisées.
3. La moyenne pour chaque triplet en limitant les valeurs à l'intervalle [-5,5].
4. la variance pour chaque triplet dans l'intervalle [-3,3] (forme 2, toujours nulle).

L'environnement artificiel sera composé de trois séquences types $\mathcal{M} = \{M_1, M_2, M_3\}$. Leur taille, qui était définie de façon aléatoire, est respectivement 24, 20 et 21 triplets. La figure 4.8 montre leurs représentations en forme décodée avec une échelle des temps commune.

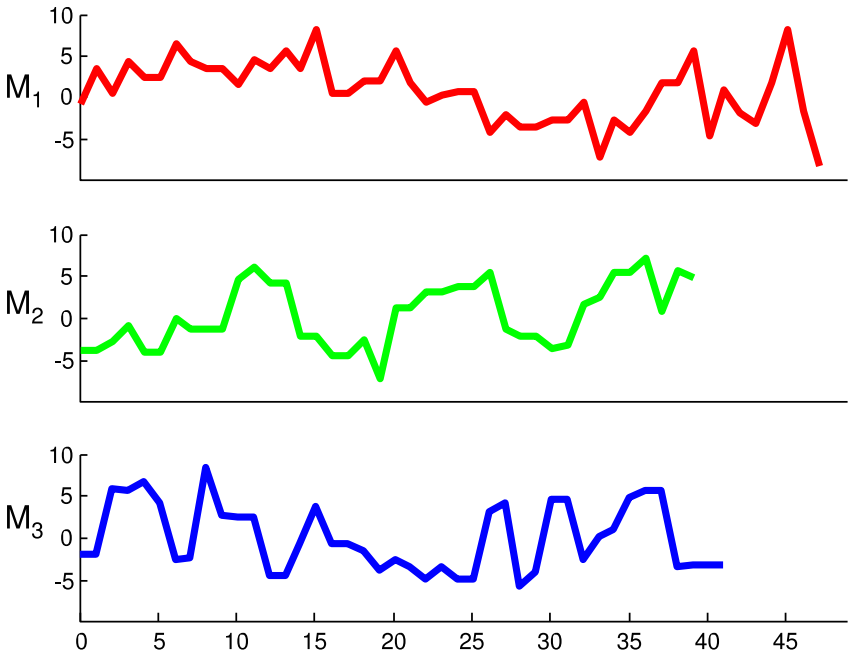


FIG. 4.8 – Les trois séquences types de l'environnement test

Les couleurs utilisées, rouge pour M_1, vert pour M_2 et bleu pour M_3, sont utilisées tout au long des analyses dans les différents diagrammes pour référer aux séquences types.

4.1.2.3 Les séquences d'évolutions

Les séquences d'évolutions représentent différents comportements temporels d'un processus $\mathcal{T} = \{T_1(t); T_2(t); \cdots T_s(t)\}$. Comme déjà évoqué auparavant, la construction des séquences d'évolutions se base sur les séquences types afin de construire de façon artificielle une relation bien définie entre les séquences d'évolutions et les séquences types. Ceci est important en vue d'analyser la méthode de localisation DTW-LB dans le contexte de la détermination des séquences types les plus proches durant l'avancement de l'évolution.

Trois méthodes pour la création de séquences d'évolutions ont été mises au point. Elles sont présentées ici, et le raisonnement qui est derrière chaque méthode est expliqué. De chaque méthode un certain nombre de séquences d'évolutions a été produit pour construire un corpus de séquences d'évolutions. Comme aucune validation quantitative n'est faite, le nombre de séquences pour chaque groupe est fixé à quatre exemplaires. Le but est de s'assurer que la méthode représente bien les propriétés de similarité et de localisation dans des contextes spécifiques.

1. Un groupe de séquences d'évolutions est obtenu par déformation d'une seule séquence type. La déformation consiste à étirer ou comprimer des parties de la séquence type en utilisant le dédoublement ou enlèvement de triplets.

   Le but de ce type de séquences dans le corpus est d'analyser l'influence que peut avoir le mécanisme du mappage DTW sur la localisation DTW-LB.

2. Un autre groupe de séquences d'évolutions est obtenu en mettant bout à bout des parties de différentes séquences types.

   Ici le but est d'analyser la capacité de détection de la séquence type dépendant du contenu de l'évolution. Un changement de la source (séquence type) durant l'évolution doit être détecté. Cela simule un changement que peut avoir le comportement du processus industriel à cause d'un changement de l'environnement industriel ou d'un choix stratégique de l'opérateur.

3. Un dernier groupe de séquences d'évolutions est obtenu en mettant une séquence type à l'échelle d'une autre en utilisant le mappage DTW.

   Le but cette fois-ci est d'analyser l'influence de la similarité entre séquences types (la proximité de classes de séquences) sur la localisation DTW-LB et en même temps d'analyser l'importance de la taille des séquences types sur cette localisation.

La notion d'aléatoire est aussi utilisée dans la production des séquences d'évolutions selon ces trois méthodes, dans le choix des parties qui sont modifiées et pour le nombre de modifications.

Il est inutile, et cela n'apporte pas d'information, de montrer tout le corpus de séquences d'évolutions qui est utilisé. Pour la démonstration et la discussion de la méthode un exemple représentatif est utilisé. Finalement les autres résultats seront discutés en bloc.

Afin de familiariser le lecteur avec la notation, un exemple du deuxième groupe de séquences d'évolution est montré.

$$T_6(t) = \{M_1(1-3); M_2(4-12); M_3(11-21)\} \tag{4.3}$$

L'exemple $T_6(t)$ est obtenu en prenant les trois premiers éléments de M_1, suivi des éléments 4 à 12 de M_2, et enfin les éléments 11 à 21 de M_3 et comprend en tout donc 23 triplets.

La représentation décodée de cette séquence d'évolution (fig. 4.9) représente les parties issues des différentes évolutions types dans leur couleur respective. Les éléments noirs soulignent la connexion entre les parties.

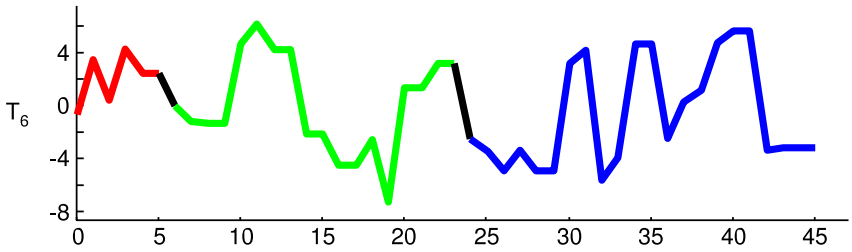


FIG. 4.9 – Un exemple de séquence d'évolution : $T_6(t)$

Rappelons que le but de ce groupe de séquences d'évolutions est d'analyser si la méthode de localisation DTW-LB est capable de retrouver les séquences types qui sont associées.

L'ensemble de ces éléments, c.-à-d. l'alphabet, les trois séquences types et le corpus de séquences d'évolutions forme l'environnement artificiel. Cet environnement est construit en utilisant des paramètres aléatoires. La reproductibilité des analyses est cependant garantie en fixant cet environnement après sa production.

4.1.3 Simulation de la recherche de prototypes dans le contexte de l'évolution temporelle

Cette section décrit la démarche utilisée afin de simuler une évolution temporelle en se basant sur l'environnement artificiel (chap. 4.1.2). Elle analyse la méthode de localisation DTW-LB (chap. 4.1.1) dans sa capacité de rechercher la séquence type la plus similaire pour un stade d'évolution donné, ce qui représente la première phase du contrôle prédictif d'aide à la décision (fig. 4.1). En même temps des outils pour la proposition de scénarios sont analysés.

Différentes représentations graphiques seront utilisées pour l'analyse de la localisation DTW-LB dans le contexte de l'évolution temporelle. Ces types de représentations sont utilisés dans la suite du document pour l'analyse et l'application de cette méthode au processus industriel. De plus, elles sont utilisées dans le contexte de la proposition de scénarios.

4.1.3.1 Simuler l'évolution temporelle

Afin de simuler une évolution temporelle en se basant sur le corpus des séquences d'évolutions $\mathcal{T}$ chaque séquence d'évolution $T_i(t)$ est représentée par une succession de stades d'évolutions $T_{i,p}(t)$ qui commencent tous à la racine $t = 0$, allant jusqu'à $t = p$ avec $p = 1 \cdots N_i$, pour N_i la taille de la séquence d'évolution $T_i(t)$.

Chaque $T_i(t)$ produit une évolution temporelle, qui est composée de N_i stades d'évolutions représentant des sous-séquences de l'évolution complète. $T_{i,2}(t)$ par exemple comprend les deux premiers triplets de $T_i(t)$, $T_{i,3}(t)$ les trois premiers triplets etc.

Prenons comme exemple la séquence $S(t)$ (eq. (4.2)) et posons $T_1(t) = S(t)$. L'évolution temporelle qui en découle est représentée par l'équation (4.4)

$$\begin{aligned} T_{1,1}(t) &= \{0\} \\ T_{1,2}(t) &= \{0,0\} \\ T_{1,3}(t) &= \{0,0,1\} \\ &\vdots \\ T_{1,28}(t) &= \{0,0,1,1,1,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,1,0,0,0\} \\ T_{1,29}(t) &= \{0,0,1,1,1,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0\} \end{aligned} \tag{4.4}$$

4.1.3.2 Recherche des localisations les plus proches

Le but est de trouver la séquence type, ou plutôt la partie d'une des séquences types qui ressemble le plus à un stade d'évolution donné. C'est pourquoi à chaque stade de l'évolution $T_{i,p}(t)$ d'une séquence d'évolution $T_i(t)$, la localisation DTW-LB est calculée pour toutes les séquences types $M_j(t) \in \mathcal{M}$.

Dans le contexte de l'environnement artificiel pour un stade d'évolution donné $T_{i,p}(t)$, trois courbes de distance DTW du type (fig. 4.5) en résultent. La séquence type qui réalise la distance DTW minimale globale sur l'ensemble des courbes représente celle qui est la plus similaire du point de vue localisation DTW-LB. L'index de ce minimum ($\hat{k} = k$) spécifie la taille de la partie de la séquence type $M_j(t)$ qui est considérée. Cette partie est notée $M_{j,\hat{k}}$.

Il serait possible de se contenter de ce résultat. Mais, puisque la localisation traite aussi le sujet subjectif qu'est la similarité, il est plus prudent de ne pas se contenter du premier minimum, mais d'utiliser les q premiers minima. Notez que ces minima peuvent provenir d'une seule séquence type, comme ils peuvent être distribués sur les séquences types. Cela dépend uniquement des valeurs de distance DTW-LB qui sont réalisées pour les différentes séquences types.

Dans la suite de cette section, seuls les trois premiers minima sont analysés ($q = 3$).

Afin de montrer un exemple de cette méthode, la séquence d'évolution $T_6(t)$ (fig. 4.9) est utilisée dans le huitième stade d'évolution $T_{6,8}(t)$ (4.5).

$$T_{6,8}(t) = \{M_1(1-3); M_2(4-8)\} \tag{4.5}$$

La figure 4.10 montre les trois courbes de distances DTW pour toutes les parties des séquences types M_1, M_2 et M_3.

Quelques détails des trois premiers minima qui sont réalisés sont repris dans le tableau 4.1 :

séquence type considérée	index de localisation $\hat{k}$	D_{dtw}	Référence
M_1	5	2.231	$M_{1,5}$
M_2	8	2.319	$M_{2,8}$
M_1	6	2.328	$M_{1,6}$

TAB. 4.1 – Détails sur les trois premiers minima de la localisation DTW-LB

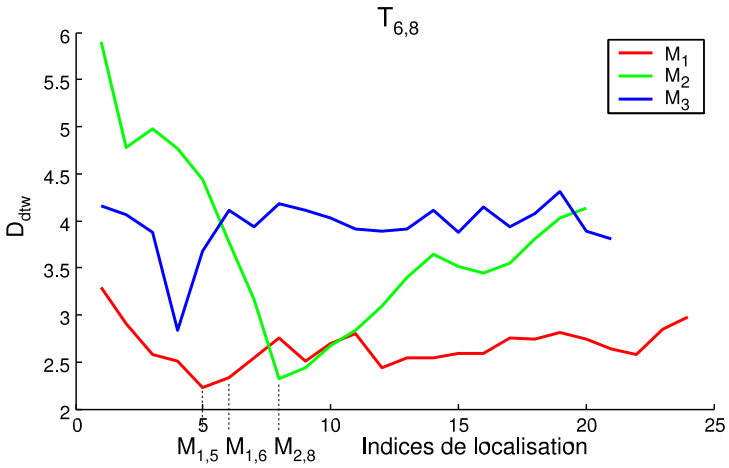


FIG. 4.10 – Courbes de distances DTW entre $T_{6,8}(t)$ et toutes les parties M_1, M_2, M_3

Ceci veut dire que pour le premier minimum par exemple, la séquence type M_1 est sélectionnée et que le minimum se trouve à l'indice cinq de la courbe ($\hat{k}_1 = 5$). La localisation DTW-LB utilise donc les cinq premiers triplets de M_1 ($M_{1,5}$) et la distance DTW réalisée est de 2.231. Pour le deuxième minimum, c'est M_2 la séquence type gagnante, avec huit triplets($\hat{k}_2 = 8$), la même taille que le stade d'évolution $T_{6,8}(t)$. Finalement pour le troisième minimum, de nouveau la séquence type M_1 est sélectionnée, mais avec six triplets et une distance plus élevée cette fois-ci.

Afin d'analyser plus en détail le résultat de ces trois minima, la figure 4.11 montre la mise à échelle des trois premiers minima avec le stade d'évolution $T_{6,8}(t)$, c.-à-d. la localisation DTW-LB avec $T_{6,8}(t)$ pour les parties des séquences types $M_{1,5}$, $M_{2,8}$, et $M_{1,6}$ en utilisant le mappage DTW vers l'espace $T_{6,8}(t)$ à chaque fois. L'information de mappage pour ces parties peut directement être extraite du résultat du calcul DTW sans calcul supplémentaire, en traçant le cheminement en rétro-propagation, en partant au point $[p, \hat{k}]$ de la GDM.

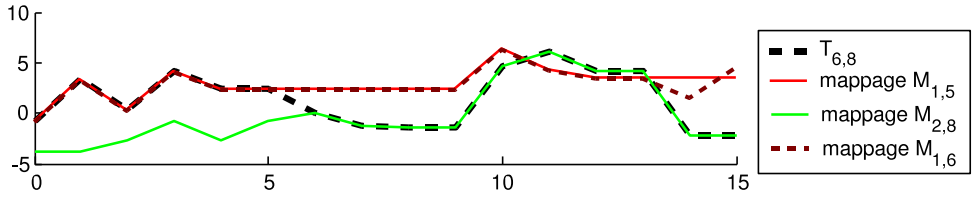


FIG. 4.11 – Localisation DTW-LB pour le stade d'évolution $T_{6,8}(t)$ avec les séquences types M_1, M_2, M_3

Rappelons que $T_{6,8}(t)$ représente les huit premiers triplets de $T_6(t)$, et que de ce fait il comprend les trois premiers triplets de M_1 et les triplets 4 à 8 de M_2 (4.5). Ce fait est bien reflété par la localisation DTW-LB, puisque $M_{1,5}$ et $M_{1,6}$ couvrent bien les trois premiers triplets de $T_{6,8}(t)$ ($t = 0 \cdots 5$) et $M_{2,8}$ couvre la partie des triplets quatre à huit ($t = 6 \cdots 15$). L'alignement réalisé par la localisation DTW-LB reprend donc bien les parties respectives.

En utilisant cette méthode il est possible de trouver pour chaque stade d'évolution les q meilleures localisations DTW-LB et d'en déterminer les parties de séquences types qui ressemblent le plus à ces stades d'évolutions. Comme la partie $M_{j,\hat{k}}$ est similaire au stade d'évolution analysé $T_{i,p}(t)$ et que $M_j(t)$ représente une séquence type, cette information peut être utilisée afin de prévoir la suite de l'évolution. Il suffit de retirer la partie utilisée pour la localisation DTW-LB et de proposer le reste de la séquence type comme prédiction.

Analysons le premier minimum de l'exemple $T_{6,8}(t)$. $M_{1,5}$ est la partie utilisée pour la localisation DTW-LB. Le reste de cette séquence type est $M_1(t)$ pour $t = 6 \cdots 24$ qui sera noté $M_{1,5+}(t)$. L'avantage

d'un environnement artificiel est le fait que le reste de la séquence d'évolution $T_6(t)$ est connu et qu'il est possible de comparer la prédiction à l'évolution réelle. La figure 4.12 montre $T_{6,8}(t)$, $M_{1,5+}(t)$, la prédiction, c.-à-d. le reste de la séquence type et le reste de la séquence d'évolution $T_{6,8+}(t)$.

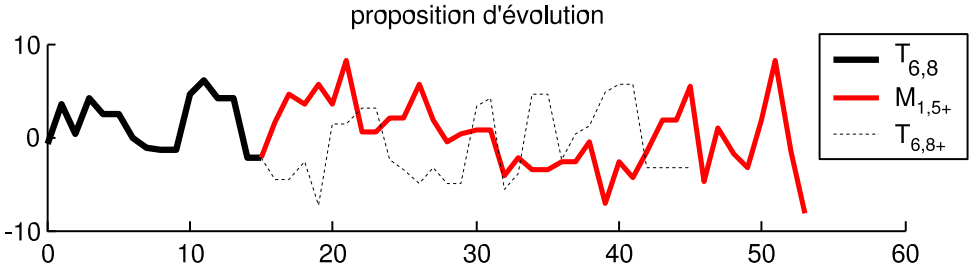


FIG. 4.12 – Première prédiction au stade d'évolution $T_{6,8}(t)$

Le résultat semble désastreux à première vue. Mais analysons plus en détail ce résultat. $T_{6,8}(t)$ comporte trois triplets de M_1 et quatre de M_2. Le reste de la séquence d'évolution $T_{6,8+}(t)$ ne contient plus aucun élément de M_1. Il reste seulement des éléments venant de M_2 et M_3. Pour la localisation, seule la première partie de la séquence d'évolution est utilisée. Le huitième stade d'évolution, qui est analysé ici, est l'instant pour lequel pour la première fois les éléments venant de M_2 prédominent ceux venant de M_1. Cela est reflété par le fait que le deuxième minimum est représenté par $M_{2,8}$ et donc la séquence type M_2.

La figure 4.13 représente la prédiction $M_{2,8+}$ faite pour ce deuxième minimum.

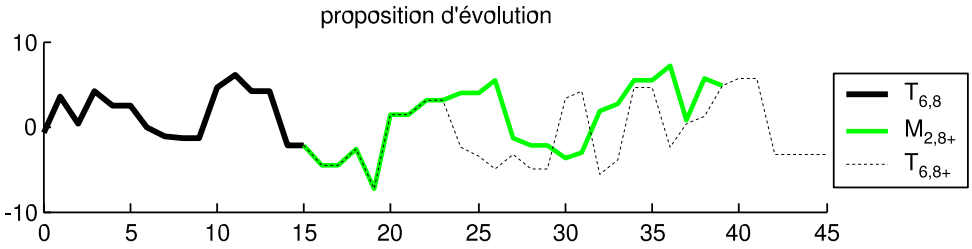


FIG. 4.13 – Deuxième prédiction au stade d'évolution $T_{6,8}(t)$

Les premiers éléments de cette deuxième prédiction couvrent bien la suite de la séquence d'évolution jusqu'à l'instant $t = 23$ c.-à-d. le $12^{\text{ème}}$ triplet de T_6. À ce moment il y a le deuxième changement dans la séquence d'évolution. Les éléments qui suivent sont $M_3(11 - 21)$, ce qui ne correspond plus à la proposition faite par $M_{2,8+}$.

Cet exemple montre bien que la localisation DTW-LB se base sur la similarité, et qu'il est important d'en tenir compte dans l'analyse des résultats qui sont produits par cette approche.

Il ne faut pas oublier que pour cet environnement artificiel de séquences d'évolution, il existe une relation entre les séquences d'évolution et les séquences types (chap. 4.1.2.3) ce qui est bien détecté par la localisation DTW-LB (fig. 4.11). Mais il n'existe aucune relation entre les trois séquences types (chap. 4.1.2.2). C'est la raison pour laquelle les prédictions faites pour cet environnement ne peuvent pas représenter de vraies prédictions. Dans le cadre du processus industriel les séquences types décrivent différents comportements du processus, mais la base est toujours la même installation physique, et il existe donc de manière implicite une relation entre ces séquences types.

Jusqu'ici un stade d'évolution spécifique a été analysé, et le résultat obtenu est explicable. Afin d'analyser l'évolution de la localisation DTW-LB avec l'avancement du stade d'évolution et de pouvoir représenter les résultats dans une forme compacte, des représentations graphiques ont été élaborées [53]. Elles sont présentées dans la prochaine partie de ce chapitre.

4.1.3.3 Graphiques d'analyse pour l'évolution de la localisation

Chaque séquence d'évolution $T_i(t)$ produit N_i stades d'évolution pour lesquels les q meilleures localisations DTW-LB sont calculées. Il n'est pas pratique d'analyser en détail toutes les évolutions pour tout le corpus de séquences d'évolutions en se basant sur les représentations utilisées dans la section 4.1.3.2. Il était nécessaire d'élaborer des graphiques standards afin de mieux pouvoir analyser les différents résultats.

Deux types principaux de graphiques peuvent être distingués :

Le graphique global : il donne un aperçu global sur l'évolution de la localisation DTW-LB pour l'ensemble des stades d'évolutions d'une séquence d'évolution donnée (fig. 4.14).

Le graphique local : il montre des détails de la localisation pour un stage d'évolution donné et un minimum spécifique (fig. 4.15).

Afin de donner un exemple d'un graphique global, de nouveau la séquence d'évolution $T_6(t)$, déjà bien connue, est utilisée. Le graphique global (fig 4.14) est composé de deux sous-graphiques :

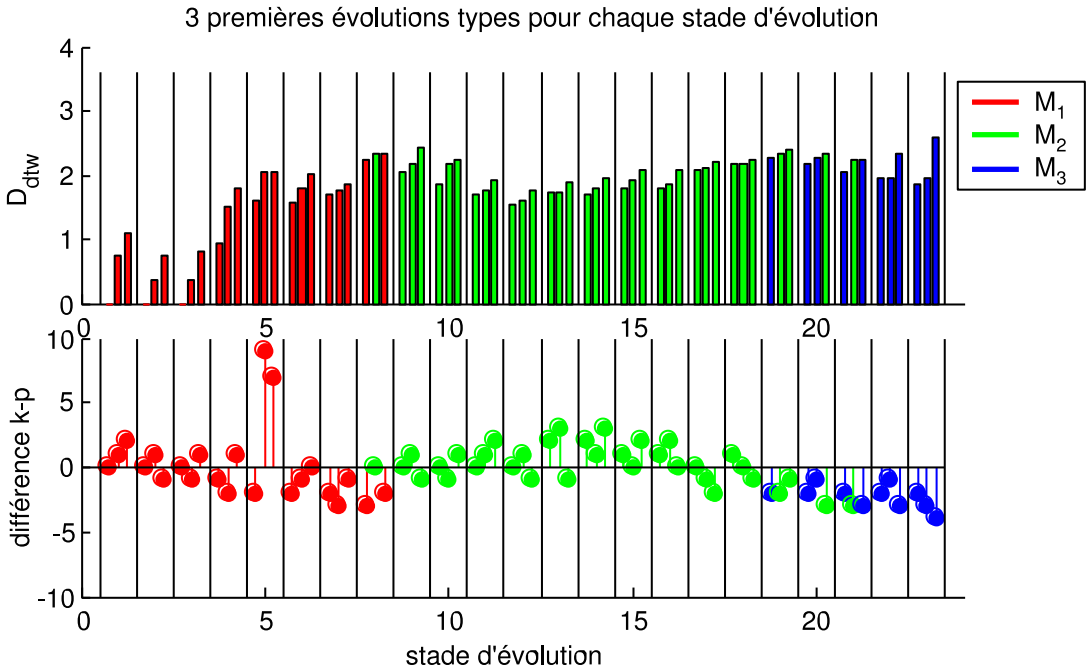


FIG. 4.14 – Graphique global de la séquence d'évolution $T_6(t)$

1. En haut les distances DTW réalisées pour les $q = 3$ meilleures localisations DTW-LB de chaque stade d'évolution sont représentées en diagramme bloc. Les traits verticaux séparent les stades d'évolutions. La couleur du bloc fait référence à la séquence type utilisée.

   Cela donne un rapide aperçu global sur les séquences types qui sont utilisées et la qualité de similarité des localisations DTW-LB pour toute l'évolution temporelle.

   Pour l'exemple donné ce graphique montre que la suite des séquences types détectées reprend bien les éléments qui composent la séquence d'évolution

   $T_6(t) = \{M_1(1-3); M_2(4-12); M_3(11-21)\}$.

   La détection est seulement décalée dans le temps. Mais aux instants de changement $p = 3$ et $p = 12$ un minimum de la distance DTW peut être constaté.

2. En bas la différence entre le nombre d'éléments $\hat{k}$ utilisés pour la localisation DTW-LB et la taille p du stade d'évolution est représentée pour chaque stade d'évolution.

   Si le point se trouve en-dessous de la ligne de base, la taille de la partie de la séquence type utilisée pour la localisation DTW-LB est plus petite que la taille du stade d'évolution. La partie

de la séquence type est donc étirée afin de représenter au mieux le stade d'évolution. Si le point se trouve au-dessus, c'est l'inverse, et la partie de l'évolution type est comprimée pour la localisation DTW-LB.

Ce graphique donne un aperçu sur les phénomènes de compression et d'étirement globaux avec le stade d'évolution. Cela est d'un côté un indicateur utilisable afin d'ajuster les paramètres de pénalité du calcul DTW. De l'autre côté, c'est un indicateur pour l'adéquation des parties de séquences types qui sont utilisées.

Pour donner un exemple, le stade d'évolution 5 a un comportement extraordinaire pour les deuxièmes et troisièmes minima, puisqu'une importante compression de dix respectivement neuf triplets est nécessaire pour la localisation. Cela indique que le coût de la compression n'est pas assez élevé (trop faible pénalité), ou bien les séquences types sont très éloignées les unes des autres. Dans ce cas c'est plutôt la pénalité, qui était 0.5 pour les directions horizontale et verticale, qui est en cause.

Le deuxième type de graphiques regroupe les illustrations qui étaient utilisées durant l'analyse de la localisation du stade d'évolution exemplaire $T_{6,8}(t)$. Il représente toutes les informations pour un stade d'évolution et un minimum spécifique. Comme exemple cette fois-ci $T_{6,21}$ le stade d'évolution 21 de la séquence d'évolution est représenté.

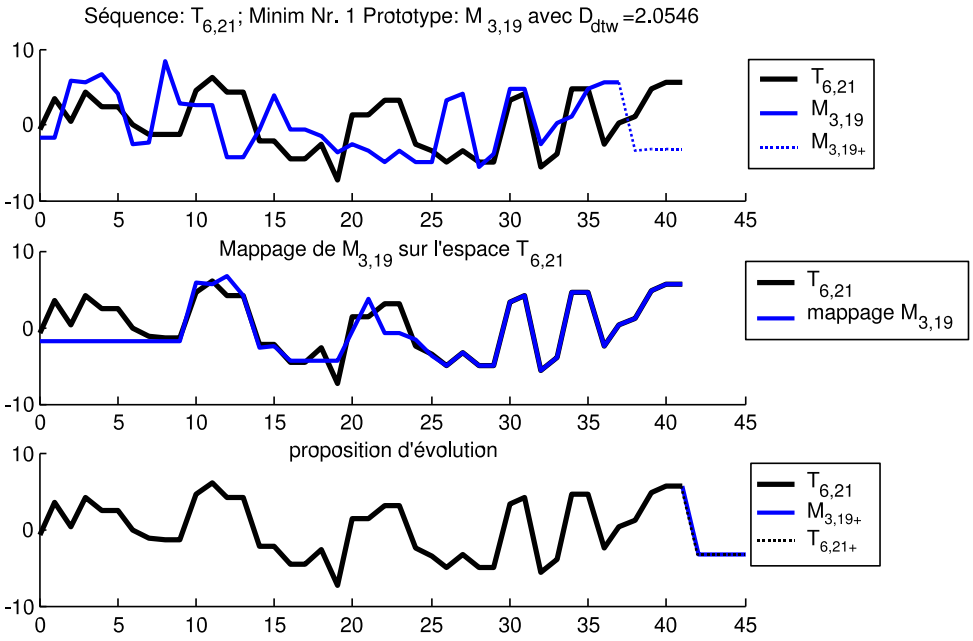


FIG. 4.15 – Première prédiction au stade d'évolution $T_{6,21}(t)$

Le graphique local se compose de trois sous-graphiques, et les représentations sont toutes dans la forme décodée avec la séquence type qui est représentée dans la couleur respective.

1. En haut, le stade d'évolution est représenté avec la séquence type correspondant au minimum analysé. Les deux parties de la séquence type, celle utilisée pour la localisation et le reste qui sera utilisé pour la prédiction, sont montrées.

   Le titre de ce graphique résume les informations sur le stade d'évolution, le minimum qui est analysé, la partie de la séquence type qui est utilisée pour la localisation DTW-LB, et la valeur de la distance DTW réalisée pour la localisation.

2. Au milieu, le mappage DTW de la localisation DTW-LB est représenté. Cela indique la qualité de similarité qui résulte de la localisation. En lien avec le premier sous-graphique, les parties de compression et d'étirement de la partie de la séquence type peuvent être détectées.

L'exemple montre que la dernière partie de la localisation recouvre exactement celle du stade d'évolution qui est issu de l'évolution en question M_3, ce qui montre que la localisation arrive bien a mettre ensemble les parties correspondantes.

3. En bas, la prédiction est comparée au reste de la séquence d'évolution. Pour ce stade d'évolution la prédiction est à 100%. Ceci résulte du fait que cette fois-ci le reste de la séquence d'évolution ne contient plus de changement d'appartenance.

   Il faut noter que premièrement la détection de la séquence type appropriée doit fonctionner, et que en plus la localisation DTW-LB doit positionner correctement la sous-séquence pour que la prédiction soit correcte.

Avec ces deux types de graphiques un grand nombre d'informations sur le comportement de la localisation DTW-LB dans le contexte de l'évolution temporelle est donné. Naturellement il est toujours possible d'analyser plus en détail les calculs qui sont faits, si un résultat n'est pas explicable sur base de ces graphiques.

Ces types de graphiques sont aussi utilisables pour le retour d'une proposition à l'opérateur. Le graphique global illustre tout l'historique de l'évolution, qui dans ce cas se construit avec l'avancement du stade d'évolution. Le graphique local donne l'information actuelle et la prédiction, respectivement la proposition d'évolution pour la suite du processus. L'opérateur aurait la possibilité de consulter les meilleures localisations avec la proposition respective.

4.1.4 Résultats pour l'environnement artificiel

Les détails des analyses basées sur l'environnement artificiel ne sont pas tous repris dans ce document, car cela dépasserait le contexte de ce travail. Dans cette section, seuls les principaux résultats qui sont obtenus par la méthode pour les trois groupes de séquences d'évolution, sont listés.

Globalement les résultats obtenus sur le corpus des séquences d'évolutions sont très positifs. Aucun résultat incompréhensible ou en contradiction avec ce qui était attendu n'a été constaté. Au contraire, les résultats étaient en général en adéquation avec les espérances.

Lorsqu'une séquence type a été utilisée directement comme séquence d'évolution (pas de groupe), la méthode a bien sélectionné cette séquence type pour la localisation DTW-LB et le mappage DTW et la prédiction est toujours à 100%. Cela semble peut-être évident, mais ce test était fait afin de voir le comportement des q meilleures localisations dans ce cas spécial.

Lorsque les séquences d'évolutions sont obtenues par étirement ou compression des séquences types (premier groupe), l'approche a toujours détecté la séquence type en question, avec une localisation DTW-LB et un mappage à 100%. Dans ce cas, la prédiction ne peut pas être optimale, car un étirement ou une compression future dans la séquence d'évolution n'existent pas dans la séquence type.

Lorsque les séquences d'évolutions étaient composées de parties de différentes séquences types non déformées (deuxième groupe et exemple T_6 utilisé pour l'explication), la méthode a bien su repérer les différentes parties qui ont composé la séquence d'évolution (les localisations DTW-LB sélectionnent les parties de la séquence type dans le même ordre). De plus, la prédiction est quasiment toujours placée au bon endroit, ce qui démontre la qualité de la localisation.

Lors de la mise à l'échelle d'une séquence type vers une autre, en donnant plus de poids à une séquence type qu'à une autre (troisième groupe), les résultats obtenus sont ceux qui étaient attendus.

La conclusion des analyses faites sur l'environnement artificiel est que cette méthode semble être au point pour être appliquée aux données qui proviennent de l'installation industrielle.

4.2 Application de la méthode au processus industriel

La méthode de contrôle prédictif telle qu'elle a été présentée auparavant (chap. 4.1) pour un environnement artificiel, est directement applicable aux données du processus. Car le même type de codage est utilisé, et dans le chapitre 3.3.3.4 des prototypes d'évolution ont été produits. Ils sont comparables aux séquences types de l'environnement artificiel sauf que cette fois-ci un contexte physique c.-à.-d. un processus réel existe. Et il sera montré que les propositions ont un réel caractère de prédiction.

Une différence en comparaison avec l'analyse basée sur l'environnement artificiel est que le codage des cycles de production est fait en utilisant différents alphabets (chap. 3.2.4). Les prototypes d'évolutions (fig. 3.42) produits auparavant utilisent les quatre alphabets de références (quatre prototypes pour chaque alphabet). Ces prototypes sont réalisés dans une approche de classification non-supervisée et reflètent de ce fait les comportements du processus. Pour l'analyse du comportement temporel et le contrôle en ligne, les différents codages sont mis en concurrence, puisque chaque prototype, indépendamment de son codage, contribue de façon équivalente à la recherche des minima pour la localisation DTW-LB. Les q meilleures localisations peuvent donc être issues de différents codages basés sur des alphabets différents. La mise en œuvre de cette concurrence est relativement simple car la localisation DTW-LB se base sur la forme codée du signal. Chaque prototype qui vient avec son codage spécifique, est comparé à la partie du cycle de production en utilisant le codage propre au prototype. Il en résulte pour chaque prototype une courbe de distance DTW (dernière colonne de la GDM) du genre présenté dans la figure 4.5. En mettant ensemble ces courbes (voir fig. 4.10 et fig.4.18) afin de trouver les q premiers minima, tous les prototypes sont en concurrence, et avec eux les différents codages.

Une autre approche, qui ne sera pas approfondie dans le cadre de ce travail, serait d'utiliser les différents codages, afin de réaliser une proposition pour l'opérateur avec différents niveaux de granularité. Pour cela chaque prototype d'évolution devrait exister dans les différents codages. Cela implique que la méthode utilisée pour la création des prototypes de comportement soit adaptée à cette particularité.

Le corpus de cycles de production utilisables (2471 cycles) est en partie ($\approx$25%) déjà exploité pour la classification non-supervisée, c.-à-d. l'extraction des prototypes d'évolution. Puisque cette analyse n'a pas réellement pu être réalisée en ligne, mais est faite sur base d'un corpus enregistré, la simulation d'évolution temporelle (chap. 4.1.3.1) est utilisée, afin de pouvoir appliquer la localisation DTW-LB dans le cadre d'une évolution temporelle. Il serait biaisé d'utiliser les cycles de production déjà utilisés pour la classification dans le contexte du contrôle prédictif. La sélection des exemples qui seront utilisés dans ce chapitre est réalisée dans la partie du corpus non utilisée dans des phases précédentes.

Trois cycles exemplaires sont utilisés afin de discuter les résultats qui sont obtenus par cette méthode : un premier cycle de production qui, selon l'avis des experts, est quasiment optimal, un deuxième cycle avec une température de coulée trop élevée, donc avec un éventuel potentiel d'optimisation, et un troisième cycle avec une évolution plutôt exceptionnelle qui est dû à une panne, afin de montrer le pouvoir d'adaptation de la localisation DTW-LB. Chaque exemple commence par la représentation du cycle avec le signal original et les reconstructions pour les quatre alphabets de référence, une courte spécification du cycle et le graphique global de l'évolution temporelle.

Afin d'éviter de surcharger le graphique global qui représente tous les stades d'évolutions analysés et de garder le plus clairement possible les résultats de cette analyse, un nombre restreint de stades d'évolutions et un nombre fixe de minima de localisations DTW-LB sont produits :

- toutes les 5 min. (300 s) une analyse du stade d'évolution est réalisée. Cette taille représente en même temps la taille de la plus grande forme dans les alphabets de références. Notez que le plus petit multiplicateur commun concernant la taille des alphabets de référence est 840, ce qui correspond à une durée de 70 min. (4200s) et dépasse la durée moyenne des cycles de productions. Cela implique que dépendant du prototype, une certaine partie de l'évolution ne sera pas prise en compte par la localisation DTW-LB.
- pour chaque stade d'évolution, les six meilleures localisations DTW-LB ($q = 6$) sont analysées et représentées dans le graphique global afin de donner un aperçu sur l'évolution temporelle des cycles analysés. Pour la localisation DTW-LB une pénalité symétrique $mphv = 0.9$ est utilisée par défaut.

Dans tous les graphiques de ce chapitre l'unité de l'axe de temps est donnée en secondes. Afin d'illustrer des détails, quelques analyses locales (localisation DTW-LB, mappage et prédictions) sont faites en cas de besoin ou d'intérêt.

Deux types de diagrammes sont ajoutés au graphique global tel qu'il était présenté dans la section 4.1.3.3. Les deux diagrammes se basent sur le fait que l'évolution est simulée et que les cycles de production sont connus dans leur globalité. Le premier de ces diagrammes (en troisième position verticale) représente la distance DTW entre la proposition et la fin réelle du cycle à chaque stade d'évolution, le deuxième (en dernière position verticale) donne une estimation du gain en énergie électrique en relation avec la proposition comparée à la consommation réelle du cycle. Cette estimation de gain d'énergie électrique est basée sur une intégration de la proposition (puissance électrique) qui est ajoutée à la

consommation au stade d'évolution et comparée à la consommation totale du cycle analysé.

Concernant l'estimation du gain en énergie électrique, quelques remarques sont à considérer dans l'utilisation de cette information :

- les prototypes d'évolution représentent des moyennes sur base de la similarité (méthode DTW) où des compressions et étirements non-linéaires sur l'axe du temps sont possibles. Ces adaptations ne sont pas sans effet sur le calcul estimatif de l'énergie électrique qui se base sur la proposition c.-à-d. la partie restante après la localisation DTW-LB.
- puisque durant l'apprentissage non-supervisé des prototypes d'évolution, la taille des prototypes n'est pas adaptée, seule la forme est apprise. L'initialisation des prototypes, et plus spécialement la taille des prototypes initiaux, importe de ce fait directement sur le calcul estimatif de l'énergie électrique.

Les informations de gain sont donc à considérer avec précaution et doivent être soutenues par d'autres informations. Notamment l'information de distortion temporelle qui est donnée par le deuxième diagramme du graphique global (fig.4.14) est à prendre en compte avec l'information de gain en énergie électrique.

Ces deux diagrammes ajoutés ne sont pas utilisables dans un contexte en ligne, puisque la fin réelle du cycle en cours est alors inconnue. Ils n'ont de ce fait qu'un intérêt pour l'interprétation des résultats dans le contexte d'une analyse évolutive sur base de cycles de production enregistrés. Cependant une prévision estimative de la consommation totale en énergie électrique d'un cycle en cours est réalisable sur base des propositions et pourra être utilisée par l'opérateur afin d'orienter le choix de la proposition à retenir.

4.2.1 Premier cycle avec un comportement quasi optimal

Ce cycle de production est référencé par le numéro 1987, la consommation globale d'énergie électrique est de 56.6 [MWh] et une température de coulée de 1626 °C a été mesurée. L'évolution de ce cycle (voir fig 4.16) est quasiment optimale et la consommation en énergie électrique est un peu en-dessous de la moyenne. La température de coulée correspond à la norme pour l'installation analysée et le contexte de la chaîne de production. Seule la durée entre la dernière coupure de la puissance électrique et la fin de la charge est plus longue que d'habitude.

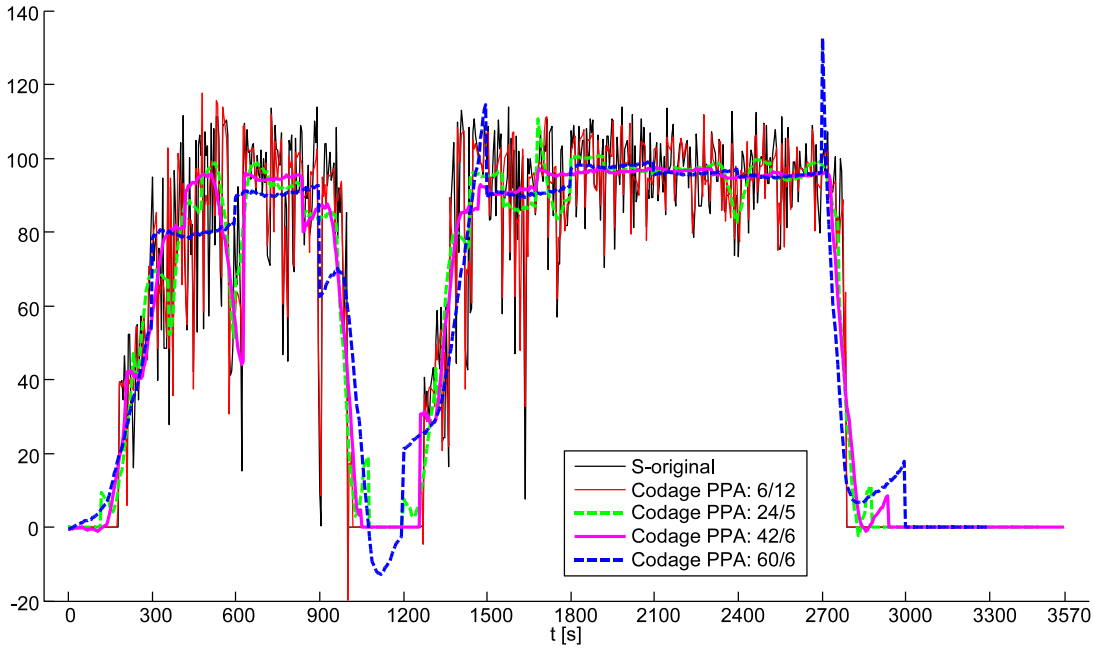


FIG. 4.16 – Ex. 1 - Nr-Ref. 1987 : Représentation du signal original avec les reconstructions basées sur les quatres codages

Pour un tel comportement peu d'améliorations sont envisageables. L'analyse de l'évolution temporelle de ce cycle est faite afin de montrer le comportement de la localisation DTW-LB sur un cycle de production réelle, de valider la méthode et d'illustrer quelques spécificités de nomenclature dans les diagrammes utilisés.

La figure 4.17 montre le graphique global de l'analyse de l'évolution temporelle de ce premier cycle de production.

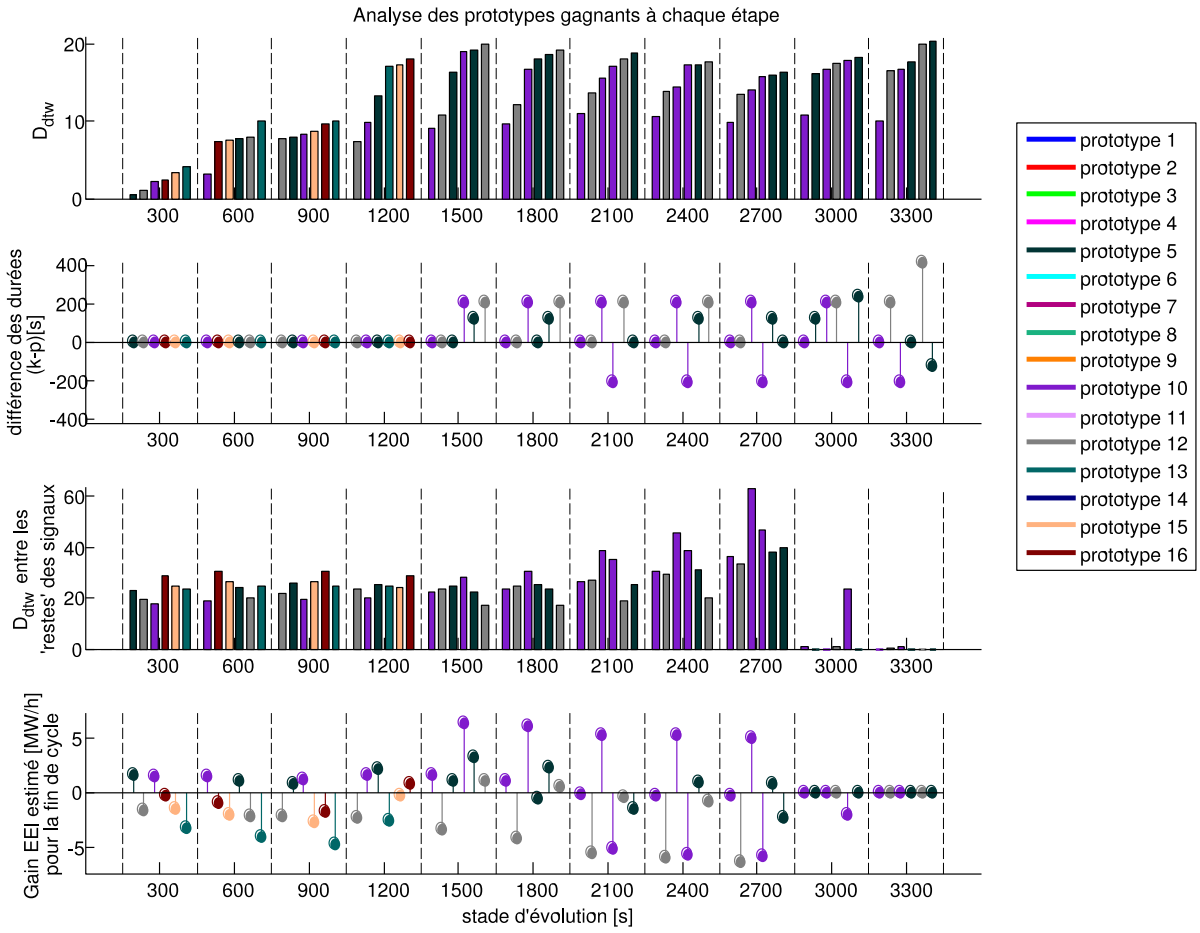


FIG. 4.17 – Ex. 1 - Nr-Ref. 1987 : Le graphique global pour les 11 stades d'évolution

Le premier diagramme représente pour chaque stade d'évolution les six meilleures localisations DTW-LB avec la distance DTW réalisée. Noter que pour les quatre premiers stades d'évolution (300s - 1200s) les six meilleures localisations DTW-LB sont réalisées pour six prototypes différents à chaque fois, et que aucune compression voire étirement n'étaient nécessaires (deuxième diagramme). Les prototypes utilisés (5,10,12,13,15,16) utilisent trois codages différents (A2,A3,A4), ce qui souligne que les codages sont dans une situation de concurrence pour une même localisation DTW-LB.

A partir du stade d'évolution cinq (1500s) seulement 3 prototypes persistent pour la localisation DTW-LB, et ils sont utilisés plusieurs fois avec des parties de taille différentes. Les stades d'évolutions de sept à dix (2100s - 3000s) illustrent cela par le prototype 10 qui est utilisé trois fois à chaque stade d'évolution. Le deuxième diagramme indique qu'à chaque fois le prototype est en premier lieu utilisé avec la taille de l'évolution ($k - p = 0$). En deuxième lieu la partie utilisée du prototype a une taille supérieure à celle de l'évolution ($k - p = 210(> 0)$), ce qui correspond à une compression de la partie prototype utilisée. Et finalement, en troisième lieu, la partie utilisée du prototype est inférieure à celle de l'évolution ($k - p = -210(< 0)$), ce qui correspond à un étirement de la partie utilisée. La figure 4.18 montre la localisation DTW-LB au stade d'évolution huit (2400s) pour tous les prototypes, avec en détail

la sélection des six meilleurs résultats.

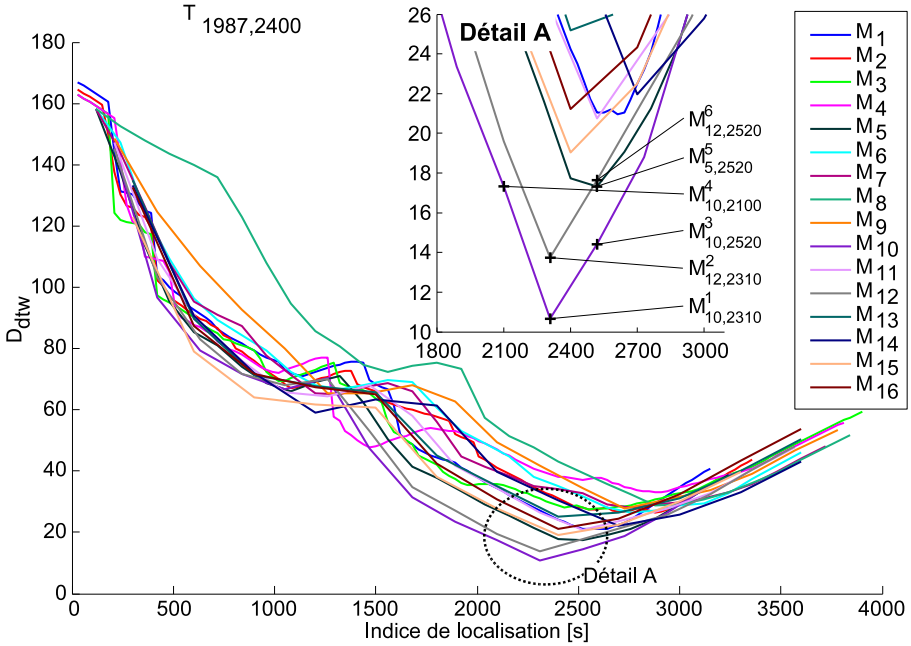


FIG. 4.18 – Ex. 1 - Nr-Ref. 1987 : Localisation DTW-LB pour le stade d'évolution huit (2400s)

Notez que pour ce graphique l'index de localisation est donné en secondes et ne représente pas l'indice de la GDM qui serait différent pour chaque alphabet. Cette forme de représentation a été choisie afin de mieux voir la relation temporelle de la localisation des différents codages.

La nomenclature pour les meilleures localisations qui est utilisée ultérieurement dans les graphiques locaux diffère légèrement de celle utilisée pour l'environnement artificiel, et sera donc expliquée ici.

Exemple : $\mathrm{M}^{q}_{np,\hat{k}[+]}$

q indique le numéro de la localisation DTW-LB,

np numéro du prototype utilisé,

$\hat{k}$ la taille du prototype, en secondes, utilisée pour la réalisation de la localisation DTW-LB,

[+] indique, si présent, que la partie restante (proposition) du prototype est représentée. Dans le cas où elle n'est pas présente, il s'agit de la partie utilisée pour la localisation qui est considérée.

Dans cet exemple $\mathrm{T}_{1789,2400}$ le stade d'évolution indique que 2400 secondes du signal sont utilisées. Pour la première localisation ($\mathrm{M}^{1}_{10,2310}$) le prototype 10, qui utilise l'alphabet A3, est sélectionné, mais seulement 2310 secondes sont utilisées. Cela vient du fait que pour A3 la taille des formes est de 42 échantillons à 5s de temps d'échantillonnage, ce qui résulte dans une taille de $42 * 5 = 210s$ par élément du code. Le multiple le plus proche du stade d'évolution est 2310, ce qui limite la partie utilisable pour la localisation DTW-LB pour ce prototype à 2310 secondes. Il en résulte que pour cette localisation DTW-LB, pas de compression ni d'étirement ne sont nécessaires. Pour la localisation numéro cinq ($\mathrm{M}^{5}_{5,2520}$) le prototype 5, qui utilise l'alphabet A2, est sélectionné avec 2520 secondes qui sont utilisées pour la localisation. A2 utilise une taille de 24 échantillons pour la forme, ce qui représente $24 * 5 = 120s$ par élément du code. L'entièreté du stade d'évolution est donc utilisée pour cette localisation, puisque 2400 est un multiple entier de 120. Cependant pour la localisation 2520 secondes du prototype sont utilisées, ce qui implique qu'une compression de la taille d'un élément du code est réalisée pour cette localisation.

Ce qui est aussi intéressant à faire remarquer est que le prototype 10 se retrouve dans tous les stades d'évolutions, et qu'à partir du cinquième stade d'évolution, ce prototype reste la meilleure localisation jusqu'à la phase finale. Cela montre que la localisation et la prédiction vont bien ensemble, et que le cycle évolue réellement selon cette évolution prototype.

Ce qui peut être intéressant dans ce contexte, c'est de voir quelle est la proposition issue de la première localisation après que le cycle tourne depuis 5 min. Le prototype 5 est gagnant (l'alphabet A2 est donc utilisé).

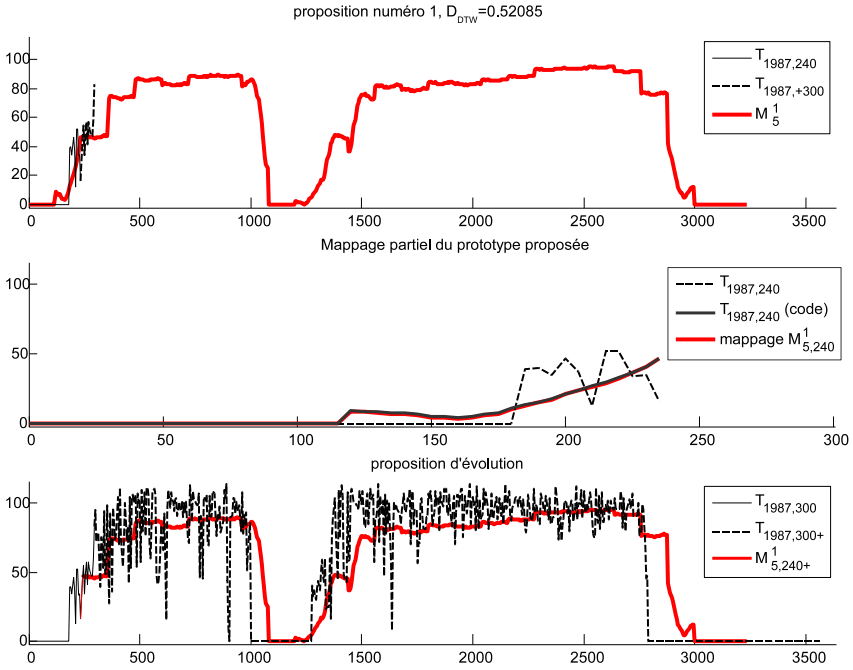


FIG. 4.19 – Ex. 1 - Nr-Ref. 1987 : Analyse locale après les premières 5 min du cycle de production.

La proposition est proche de l'évolution qui va réellement suivre. Cela confirme l'affirmation prédite que la proposition ne peut être adéquate que si le système décrit par les séquences est un système physique réel.

Le faible potentiel d'amélioration de cet exemple est aussi repris par le dernier diagramme du graphique global (voir fig.4.18), représentant l'estimation de gain en énergie électrique. La plupart des localisations DTW-LB ne proposent pas de gain pertinent, voire de faibles valeurs positives.

Seulement pour quelques instants dans l'évolution (stade d'évolution 5 - 9), une estimation positive (≈5[MWh]) peut être constatée. Il s'agit à chaque fois de la deuxième localisation du prototype 10 voire du prototype 5. Comparons les propositions avec l'information de mappage dans le deuxième diagramme. Pour toutes ces localisations une compression de la partie prototype utilisée pour la localisation peut être constatée. De plus ces propositions sont à chaque fois au préalable déjà utilisées sans compression. Cela indique que la proposition est nécessairement plus courte, et l'intégration pour l'estimation du gain résulte dans une consommation globale plus petite. Dans ces conditions, il est possible que la proposition qui en découle reflète une évolution qui n'est pas capable de réaliser la contrainte de la température de coulée. Ceci montre qu'il est important de pouvoir interpréter les résultats d'une localisation et d'utiliser ces informations pour le choix de la proposition qui sera utilisée pour la conduite de l'installation.

Avec ce résultat positif pour un cycle d'évolution quasi optimal, analysons maintenant un cycle pour lequel les experts estiment qu'il existe un potentiel d'amélioration.

4.2.2 Deuxième cycle avec potentiel d'amélioration

Ce cycle de production est référencé par le numéro 41, la consommation globale d'énergie électrique est de 61.1 [MWh] et une température de coulée de 1674 °C a été mesurée.

La consommation en énergie est légèrement au-dessus de la moyenne et ne donne donc pas l'impression d'un potentiel d'amélioration important. Cependant la température de coulée est nettement plus élevée

que la norme. En évitant cette surchauffe de l'acier liquide, une réduction de la consommation d'énergie électrique aurait été possible. Ce cycle de production comprend donc un certain potentiel d'optimisation.

La figure 4.20 montre l'évolution temporelle de ce cycle. A première vue, seulement une phase perturbée au début de la deuxième phase de fusion avec une embouchure au milieu de la phase peut être observée.

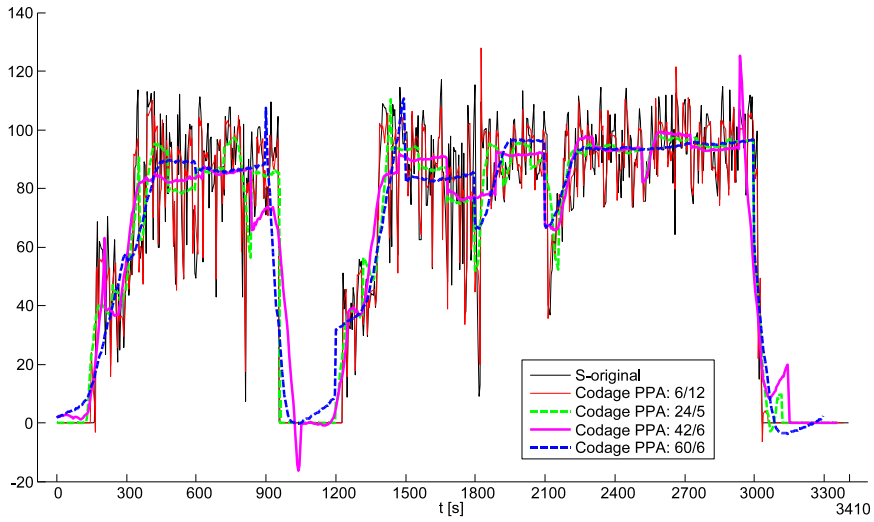


FIG. 4.20 – Ex. 2 - Nr-Ref. 41 : Représentation du signal original avec les reconstructions basées sur les quatres codages

La figure 4.21 montre le graphique global avec les stades d'évolutions pour ce deuxième exemple.

En premier lieu il faut constater que, à partir du stade d'évolution quatre, uniquement des compressions et pas d'étirement sont utilisés pour la localisation DTW-LB, ce qui indique que la première partie de l'évolution du cycle est plus courte que cette même partie dans les prototypes sélectionnés par la localisation DTW-LB. De plus, sur presque toute l'évolution une estimation de gain positive peut être constatée avec cette fois un niveau de gain légèrement plus élevé.

Analysons donc plus en détail le stade d'évolution trois qui est encore avant la partie de compression permanente. Les six localisations se basent sur différents prototypes (10,15,12,16,13,5) utilisant les alphabets (A2,A3,A4). Notez que ce sont les mêmes qui étaient utilisés dans le premier exemple durant la première phase. Cependant la distance DTW pour la première localisation DTW-LB est considérablement plus petite que celle des cinq autres localisations. Aucune compression ni étirement n'existe pour les six localisations de ce stade d'évolution. Analysons le dernier diagramme qui indique les gains en énergie électrique estimés pour ce stade d'évolution. Seulement les localisations 1 et 6 basées sur les prototypes 10 (A3) et 5 (A2) proposent un gain notable. La figure 4.22 montre les trois diagrammes du graphique local de la première localisation, et le diagramme de proposition de la sixième localisation.

Les deux propositions sont similaires, mais issues de deux prototypes différents. Dans les deux cas la proposition suggère de prolonger légèrement la première phase de fusion et d'arrêter la deuxième phase de fusion plus tôt, avec dans les deux cas, une estimation de gain d'énergie électrique. Notez que le cycle finit avec une température de coulée trop élevée, et arrêter la fusion plus tôt pourrait éviter cette température trop élevée.

Notez que ce genre de cycle est bien suivi par l'approche, et les propositions faites sont compréhensibles, explicables, et elles reflètent le potentiel d'amélioration bien réel dans ce cycle.

4.2.3 Troisième cycle à comportement exceptionnel

Ce cycle de production est référencé par le numéro 42, la consommation globale d'énergie électrique est de 60.2 [MWh] et une température de coulée de 1658 °C a été mesurée.

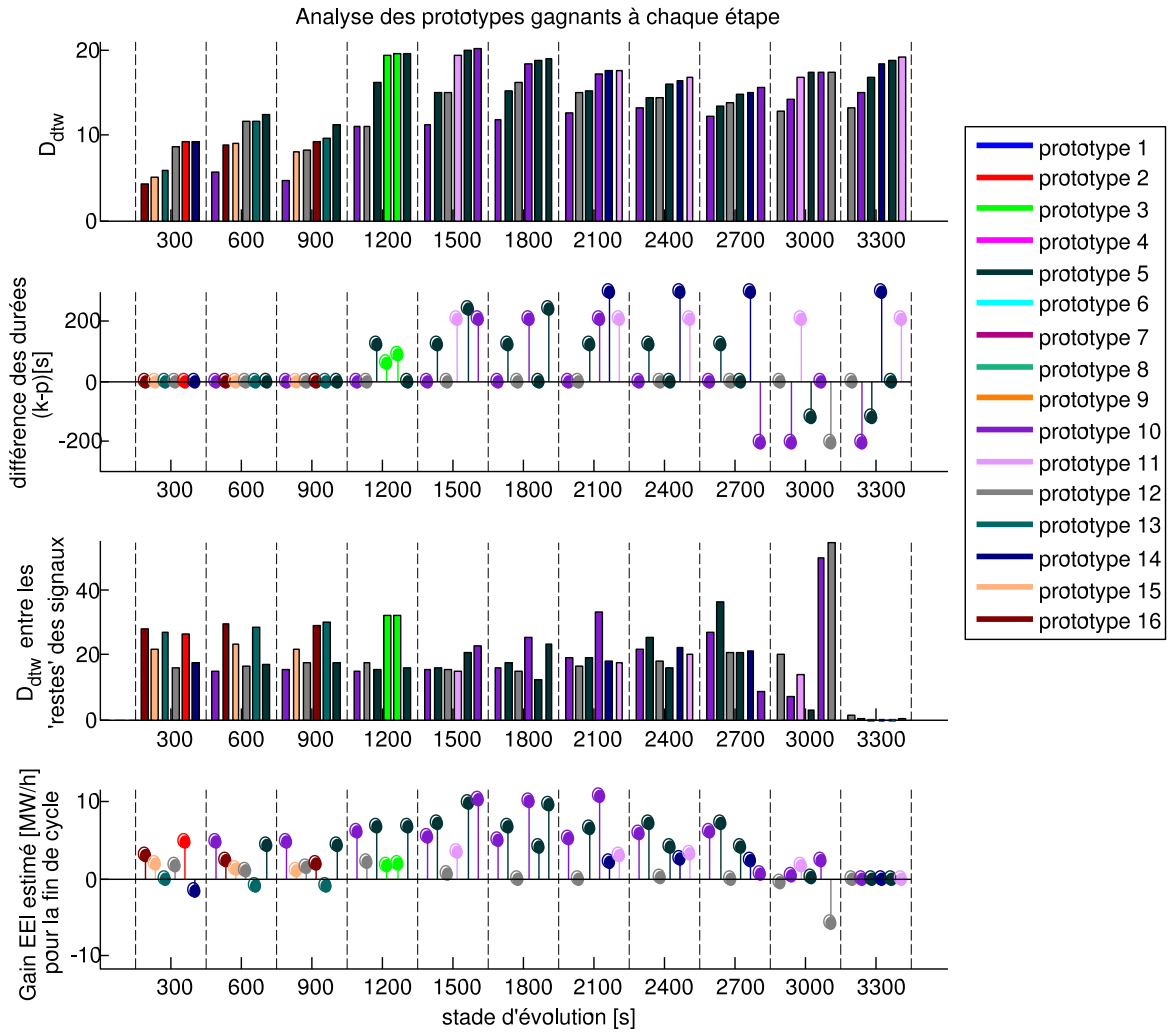


FIG. 4.21 – Ex. 2 - Nr-Ref. 41 : Graphique global pour les 11 stades d'évolution

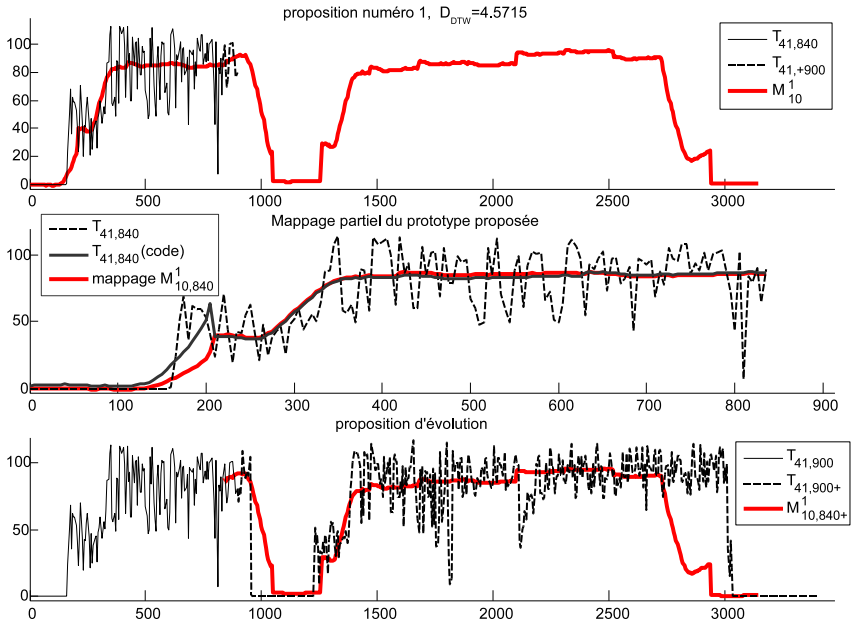


FIG. 4.22 – Ex. 2 - Nr-Ref. 41 : Graphique local du 3ème stade d'évolution (900 [s]) :
(a) détails de la première localisation ;
(b) diagramme de proposition de la sixième localisation.

La consommation en énergie est moyenne, et la température de coulée un peu au-dessus de la moyenne. Ce cycle ne représente donc pas des valeurs exceptionnelles, mais le comportement temporel de ce cycle est hors du commun (voir fig.4.23) et pour cela intéressant afin de démontrer le potentiel d'adaptation de la méthode proposée.

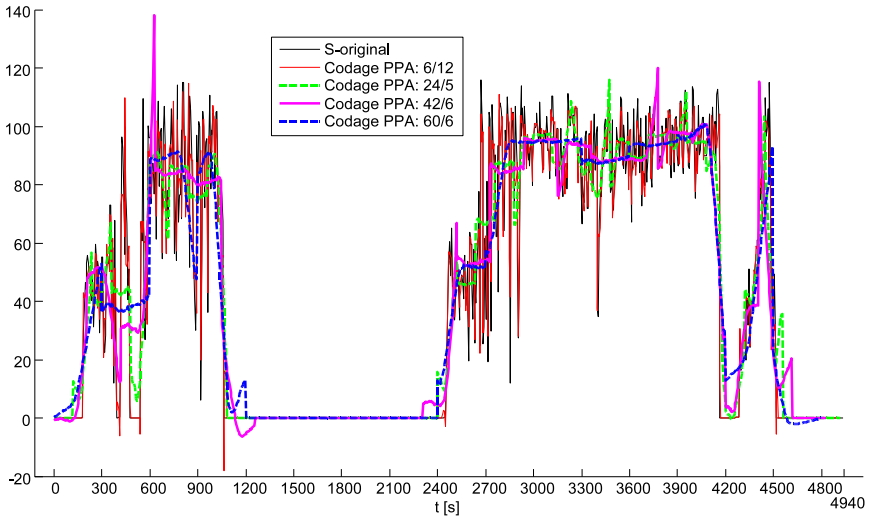


FIG. 4.23 – Ex. 3 - Nr-Ref. 42 : Représentation du signal original avec les reconstructions basées sur les quatres codages

Le cycle a une évolution hors du commun, parce que la durée entre la première et la deuxième phase de fusion est excessivement longue (presque 30 min). Dans les données manuelles de cette charge qui contiennent des informations supplémentaires fournies par l'opérateur, il est noté que l'électrode s'était brisée et qu'un pointage d'un nouvel élément graphite était nécessaire, ce qui explique la durée anormale de cette période. Notez aussi que différentes coupures de courant peuvent être observées durant la première phase de fusion, ce qui laisse penser que déjà à cet instant quelques problèmes ont été rencontrés. De plus, une coupure peut être observée à la fin de la deuxième phase de fusion. Cela peut venir d'une mesure manuelle de la température du bain qui nécessite une telle coupure. Une telle mesure manuelle est nécessaire si le robot qui effectue normalement cette mesure est en panne. Pour des causes de sécurité, les mesures manuelles de la température du bain ne sont autorisées que quand le four est hors tension.

Le graphique global (fig. 4.24) montre que presque tout au long de l'évolution un gain serait possible, ce qui est normal si l'on considère la perte d'énergie durant la phase d'attente. Une autre constatation est que déjà au stade d'évolution deux (600s), au début de la première phase de fusion, le niveau de la distance DTW obtenu pour la localisation DTW-LB est nettement plus élevé que dans les exemples précédents. Cela reflète les problèmes (coupures de courant) dans cette phase.

Ce qui est plus intéressant pour ce travail est comment la localisation DTW-LB gère la situation. Le graphique global montre que la localisation s'adapte en utilisant un étirement à partir du stade d'évolution six (1800s) quand la durée entre les deux phases de fusion devient plus longue que d'habitude.

Analysons plus en détail différentes propositions le long de l'évolution. La figure 4.25 montre le dernier diagramme avec les premières propositions pour les stades d'évolution trois (900s) avant la coupure, sept (2100) durant la coupure et neuf (2700s) après la coupure.

La proposition au stade d'évolution trois (900s) est standard, puisque à cet instant pas d'anomalies majeures ne sont apparues. De plus elle propose de prolonger la première phase de fusion. L'incident, "brise d'électrode" n'est pas prévisible et l'opérateur n'a pas d'autre choix que d'arrêter immédiatement le processus en cours pour remplacer l'électrode afin de pouvoir continuer le processus. Au stade d'évolution sept (2100s) la proposition suggère de lancer la deuxième phase de fusion afin d'être conforme à une des évolutions prototypes. En supposant que les travaux de réparation ne sont pas encore achevés à cet instant, l'opérateur ne peut pas appliquer cette proposition. Au stade d'évolution neuf (2700) la fusion du deuxième panier a été lancée et la localisation en tient compte en étirant la plage à puissance nulle

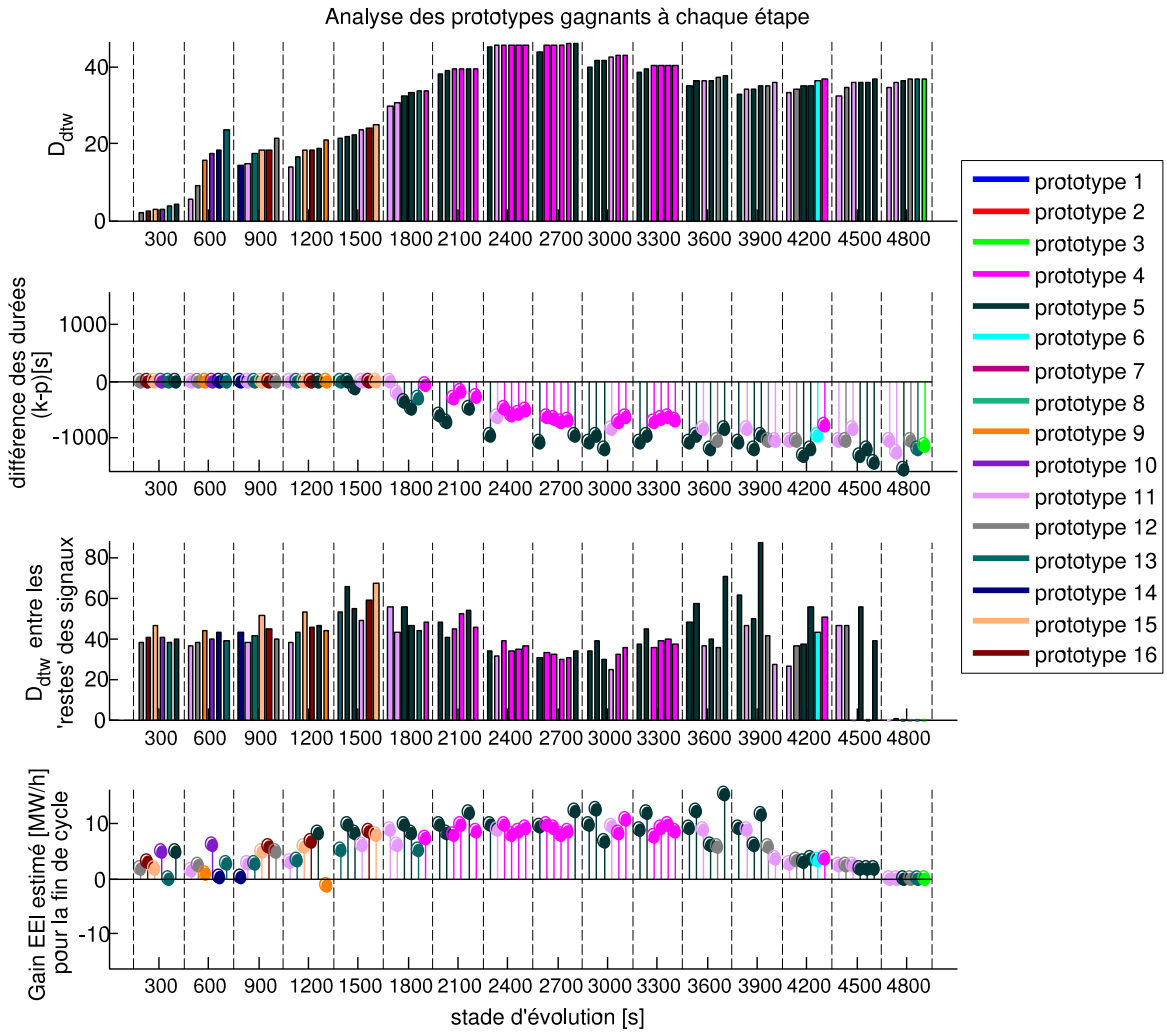


FIG. 4.24 – Ex. 3 - Nr-Ref. 42 : Le graphique global pour les 16 stades d'évolution

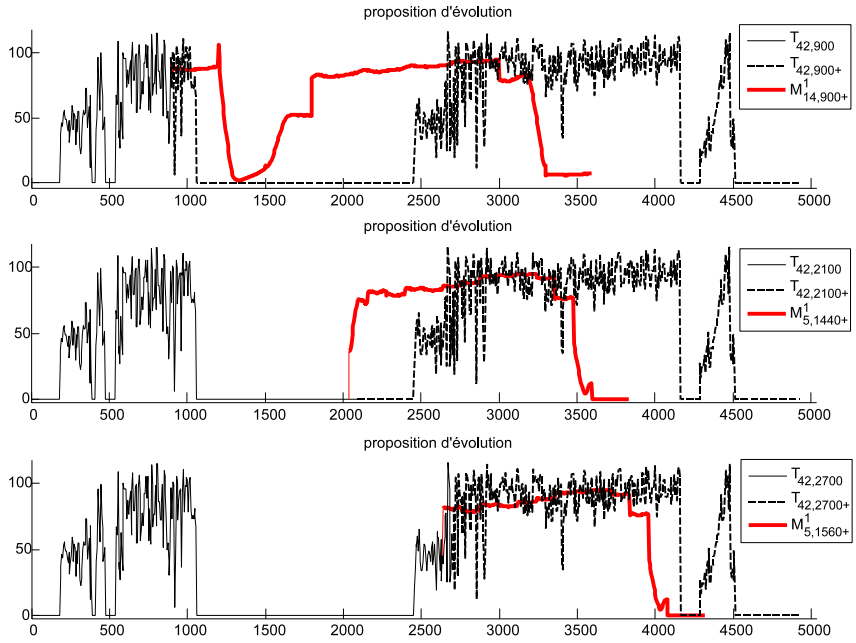


FIG. 4.25 – Ex. 3 - Nr-Ref. 42 : Analyse de l'évolution des premières propositions à 900s, 2100s et 2700s du cycle de production

et réalise une proposition adéquate. La méthode est capable de se synchroniser même dans des cas de comportement peu habituels. La proposition qui résulte de la localisation DTW-LB ne se base que sur la similarité entre les parties d'évolutions, et de ce fait ne tient pas compte de pertes de chaleur durant la phase d'attente. Cela explique pourquoi un arrêt avancé est proposé, qui dans le contexte de ce cycle spécifique donnerait probablement une température de coulée trop faible.

4.2.4 L'utilisation des critères d'optimisation

Ces trois exemples montrent bien que la méthode de localisation DTW-LB est capable de localiser des parties d'évolutions d'un cycle de production en cours et de détecter un ou plusieurs prototypes d'évolutions qui sont similaires avec ce qui peut être observé à ce stade d'évolution.

L'adaptation de la localisation selon le stade d'évolution est bien visible dans les exemples deux et trois. Un tel comportement est possible puisque le calcul DTW est capable de réaliser des déformations non-linéaires sur l'axe des temps. Même des situations spéciales avec des étirements importants n'influent que peu sur la localisation du reste de la séquence.

Les propositions qui sont réalisées semblent être appropriées. Il est dans ce contexte important de noter qu'il est indispensable de ne pas se limiter à un seul choix, puisque la localisation DTW-LB ne juge que selon des critères de similarités au niveau de l'évolution, et que cela peut résulter dans des évolutions non réalisables en ajoutant les contraintes du processus physique qui, elles, ne sont pas prises en compte dans l'analyse de localisation.

Remarquez aussi que les 16 évolutions prototypes qui sont issues de la classification non-supervisée reprennent plus ou moins bien des évolutions, voire le comportement caractéristique du processus. Cette constatation montre que des prototypes basés sur des alphabets considérés comme moins appropriés par la validation des alphabets (Annexe C) peuvent cependant être utilisés dans le contexte global. Néanmoins les prototypes 1 jusqu'à 4, qui utilisent l'alphabet A1 apparaissent plus rarement (voir fig. 4.21 et fig. 4.24) dans la localisation DTW-LB. Dans la validation, cet alphabet est considéré comme peu approprié. Sans ignorer le fait que cette constatation ne se base que sur quelques exemples analysés durant cette étude qui n'ont pas nécessairement une pertinence statistique, une question peut être soulevée : ne faudrait-il pas valider des éléments d'une approche globale, comme ici les alphabets et la classification non-supervisée

des évolutions temporelles, dans leur contexte global, qui est ici l'adéquation de proposition d'évolution pour l'opérateur. Il est évident qu'une telle validation n'est pas triviale et que des validations soi-disant "locales", comme la validation des alphabets, peuvent aider dans la démarche.

Finalement ces exemples montrent que, afin de pouvoir décider si l'une ou l'autre proposition est valable, plusieurs informations doivent être considérées ensemble, et une bonne connaissance du processus est utile, afin d'interpréter les réactions de la méthode, ainsi que les propositions qui sont faites. Ceci nous amène vers un sujet qui n'était pas encore discuté jusqu'ici, concernant l'utilisation des critères qu'il s'agit d'optimiser.

Les analyses et méthodes précédentes fournissent un moyen de sélectionner des évolutions prototypes qui ressemblent plus ou moins bien à l'évolution qui est en cours. Dans les exemples les six premières localisations DTW-LB sont listées. Dans la partie théorique il a été dit que le nombre des localisations est un paramètre qui peut être spécifié par l'utilisateur. Il a aussi été relevé et montré que les localisations peuvent être issues de plusieurs évolutions prototypes.

L'idée principale est d'utiliser les critères qu'il s'agit d'optimiser afin de sélectionner les propositions selon ces critères. Une première approche de cette vue a été introduite précédemment avec l'information de gain en énergie électrique. La notion de gain n'est naturellement pas directement utilisable, puisque la consommation finale du cycle en cours n'est pas encore connue. Mais une autre approche est possible en se basant sur des estimations de consommation des prototypes.

L'idée est de favoriser dans toutes les q localisations DTW-LB qui ne reposent que sur la similarité d'évolution, les évolutions prototypes pour lesquelles la consommation répond aux critères donnés.

En théorie, plusieurs critères différents seraient possibles, comme par exemple la consommation en gaz naturel (par les brûleurs), le volume de charbon (injecté durant le procédé) ou la consommation en oxygène pour n'en citer que quelques-uns. Tous ces apports influent sur le coût de production, ils sont donc sujets à optimisation.

Cependant, toute l'étude a été réalisée sur base d'un seul type de signal (une seule variable), la puissance électrique. Le prototype d'évolution est un représentant de la catégorie de cycles qui ont une évolution semblable en se limitant à la dimension de la puissance électrique.

Deux approches sont possibles dans ce contexte :

- Les évolutions prototypes peuvent être utilisées afin de faire une classification d'un grand nombre de cycles de production. S'il existe une corrélation entre l'évolution du processus et la consommation, cette classification devrait faire émerger cette relation.
  S'il y a relation, il serait possible de partir du prototype d'évolution, de favoriser les propositions pour lesquelles les prototypes sous-jacents remplissent le critère posé.
- Réaliser une approche multivariable en sélectionnant les variables à partir desquelles une estimation de consommation est possible pour les critères en question.

Dans la conclusion ce sujet sera repris et quelques réflexions et perspectives seront élaborées.

Déjà maintenant une optimisation selon un critère (l'énergie électrique) serait possible en se basant sur l'estimation de la consommation de l'énergie électrique pour les prototypes.

Toute opération d'optimisation d'un processus nécessite une possibilité de retour sur le processus en question. Sans pouvoir influencer le processus analysé, une amélioration est impossible. Il faut donc fermer la boucle du "contrôle".

4.2.5 Quand la boucle se ferme !

Afin de le formuler clairement, la méthode proposée se base sur des données d'un processus réel, et toutes les analyses et méthodes ont été réalisées avec l'idée d'une réalisation sur le site. Cependant le projet CECA s'est terminé avant qu'une implémentation sur le site de ProfilARBED Esch-Belval n'ait pu être réalisée.

Aujourd'hui l'opérateur se réfère à un catalogue de quelques consignes qu'il est sensé utiliser, dépendant de différentes conditions de l'installation. Ces consignes définissent des plages de la puissance électrique pour un cycle de production complet, en se basant sur l'énergie électrique consommée pour refléter l'avancement du processus.

En parallèle, l'opérateur a une connaissance propre et un modèle interne du comportement de l'installation. Il n'est pas possible de conduire l'installation en suivant strictement la consigne à cause d'un grand

nombre d'événements inattendus et imprévisibles qui se produisent tout au long de l'évolution. L'opérateur se base alors sur son modèle interne pour réagir à ces perturbations en manipulant les paramètres de l'installation en vue de stabiliser le processus.

L'idée était de donner à l'opérateur une proposition de scénarios afin de l'aider dans la tâche de minimiser les coûts de production. Les informations qui étaient prévues pour cela sont : les "q" meilleures propositions de l'évolution future du cycle actuellement en cours, les estimations de consommations pour ces propositions, et éventuellement, basé sur le graphique global, un graphique sur lequel l'opérateur pourrait suivre l'avancement des localisations et propositions depuis le début du cycle.

Sur base de ces informations, l'opérateur pourrait décider de suivre l'une ou l'autre proposition. Comparé à la consigne schématisée et fixe dont dispose l'opérateur aujourd'hui, le scénario proposé est une consigne qui s'adapte et évolue avec l'évolution du cycle de production. De plus, l'estimation de la consommation ainsi que le suivi du processus global pourraient enrichir les informations qui existent aujourd'hui.

Tel que illustré dans le schéma de l'approche de contrôle prédictif (voir fig.1.9) c'est l'opérateur qui est prévu pour fermer la boucle de contrôle en agissant sur les paramètres afin de suivre la proposition retenue.

4.3 Conclusion du chapitre

Dans ce chapitre la phase de contrôle en ligne a été développée. Elle se base sur la classification non-supervisée des évolutions du processus réalisée dans le chapitre 3. Afin de pouvoir faire une proposition de scénarios qui est en adéquation avec l'avancement du cycle de production, il est nécessaire de trouver les prototypes d'évolutions, dont le début est similaire à celui du cycle en cours. Il faut pour cela pouvoir localiser une sous-séquence (la partie connue du cycle en cours) dans une séquence (le prototype qui représente un cycle de production complet) en se basant sur la similarité.

L'approche qui a été développée repose sur la méthode déjà utilisée pour la classification non-supervisée basée sur la similarité, à savoir la méthode DTW. Puisque pour les deux séquences le début du cycle est commun, la localisation est bornée de ce côté (DTW-LB).

Étant donné que la subjectivité de la similarité joue un rôle et que la localisation se base seulement sur une partie d'une évolution complète, plusieurs scénarios sont possibles. Cela a été pris en compte par le fait de pouvoir spécifier le nombre de localisations DTW-LB qui va être calculé et desquelles résulte le même nombre de propositions.

Puisque une implémentation de l'approche sur le site de production n'était pas réalisable dans le cadre du projet CECA, une validation de l'approche globale a été impossible. Cependant, sur base d'exemples, il a été possible de démontrer en partie le potentiel de la méthode dans le domaine de la flexibilité d'adaptation et relatif à une estimation d'optimisation de la consommation en énergie électrique.

Il aurait naturellement été intéressant de voir comment le processus réel, voire l'opérateur sur site, réagit sur les propositions issues des prototypes d'évolutions, qui sont créés par une approche de fouille de données temporelles avec des données provenant de l'installation en se basant sur une méthode non-supervisée d'extraction de connaissances. De plus il serait intéressant de connaître le potentiel réel d'optimisation par une telle approche une fois que la boucle serait fermée.

Cependant, puisque l'étude frôle le domaine de la science cognitive, il est toujours possible que la boucle se ferme sur un autre niveau : car dans le cadre de cette étude, différentes analyses ont été réalisées sur base des données du processus. Cela a créé des connaissances supplémentaires qui peuvent influencer les réflexions que se feront les personnes qui ont participé au projet. Du point de vue d'une approche de la science cognitive la boucle est fermée par ce biais. Cependant, le temps de réponse d'un tel système cognitif sur la consommation du processus de production contient un important potentiel d'amélioration.

Chapitre 5

Conclusions et perspectives

Les travaux de recherche présentés dans cette thèse tournent autour de trois sujets majeurs : la fouille de données temporelles, la similarité et les approches connexionnistes dans le domaine de l'extraction de connaissances par la classification non-supervisée. Les recherches étaient réalisées dans le cadre d'un projet de recherche CECA réunissant plusieurs partenaires autour d'une même problématique, chacun avec une approche propre à lui. Un processus réel, à savoir un four à arc électrique, qui est un élément de la chaîne de production de l'acier, a fourni le contexte applicatif de ce travail. Suite à ce cadre projet, une certaine optique a orienté les démarches de ce travail de thèse vers le sujet d'extraction de connaissances et les méthodes connexionnistes. C'est le contexte de l'application industrielle qui a fait émerger deux hypothèses qui ont guidé tous les travaux de recherche.

La première hypothèse se base sur des observations du processus et des discussions avec les experts de l'installation industrielle. Elle consiste dans l'affirmation qu'il existe une relation entre la consommation globale du processus et l'évolution temporelle avec laquelle le processus est alimenté en énergie. Même si cette hypothèse a été dressée dans le cadre de ce projet spécifique, il est légitime de supposer qu'elle peut être valable pour un bon nombre de processus industriels, et qu'elle peut être étendue à d'autres éléments que l'apport en énergie. Cette première hypothèse a fourni le but principal de ce travail dans le domaine applicatif qui consistait à développer une méthode prédictive, afin de guider l'opérateur dans la conduite du processus tout en visant la réduction de la consommation en énergie.

La deuxième hypothèse se base aussi sur les observations du processus et affirme que, si la relation de la première hypothèse existe, une certaine souplesse dans l'analyse de l'évolution temporelle doit être garantie, car des événements peuvent être décalés dans le temps, voire des phases avoir des durées variables, sans que pour autant cela ait un impact important sur le comportement global du processus. Cela a mené vers les domaines de la similarité et des séries temporelles, deux sujets de recherches à part entière. Chercher la similarité de séries temporelles tout en gardant une souplesse dans l'alignement temporel est un sujet de recherche ouvert. Dans ce travail, le sujet a été abordé avec une orientation connexionniste, avec comme résultat une méthode de classification non-supervisée de séries temporelles de taille arbitraires, et une méthode de codage qui s'adapte de façon non-supervisée au signaux qui sont analysés.

Revenons sur la première hypothèse et les résultats applicatifs issus de ce travail. Durant tout ce travail il a été montré qu'il est possible de formuler une proposition de scénarios prédictifs sur base d'analyses des évolutions temporelles du processus. En outre, sur base d'une estimation de la consommation en énergie électrique des scénarios proposés, un premier moyen est présenté, qui est utilisable afin de guider l'opérateur dans le choix du scénario en vue de réduire la consommation globale du processus. Dans la dernière partie du travail, quelques cycles de production exemplaires mettent en évidence de façon qualitative le potentiel d'optimisation de l'approche. D'autres exemples traités durant les analyses, qui ne sont pas exposés dans ce manuscrit, confirment ce potentiel. De plus, il a été montré dans un des exemples qu'un autre critère, notamment la température de coulée, qui n'est pas explicitement représentée par le signal analysé, aurait pu être amélioré. Cela souligne la relation qui existe entre l'évolution temporelle du cycle de production et les critères d'optimisation.

Cependant une réelle preuve de la première hypothèse n'a pas pu être établie, parce qu'une implémen-

tation de l'approche sur le site de production n'était pas réalisable dans le cadre du projet de recherche CECA. Même si différents cycles analysés confirment l'hypothèse, seul le contexte du processus réel avec la fermeture de la boucle par l'opérateur peut démontrer la réelle influence de l'approche sur le processus, et en apporter la preuve.

Concluons qu'il existe des indices réels qui reflètent la relation entre l'évolution temporelle et la consommation globale et qu'une approche basée sur l'analyse de cette évolution a le potentiel de produire des propositions d'évolutions qui peuvent améliorer la consommation globale. Sans oublier que cette relation est basée sur des connaissances des experts de l'installation.

Ces résultats du domaine de l'application industrielle ne sont pas les seuls résultats des travaux réalisés dans le cadre de cette thèse. Car avant de pouvoir faire une proposition de scénarios, différents sujets ont dû être résolus. Tout d'abord une localisation de sous-séquences temporelles est nécessaire. Afin de réaliser une localisation qui tient compte des propriétés du domaine temporel, une méthode a été mise au point, qui repose sur une mesure de similarité. La méthode de localisation a été appelée "localisation DTW-LB", parce que : premièrement la "distance" DTW a été utilisée comme mesure de similarité, et deuxièmement la localisation est bornée d'un côté (du côté gauche qui représente le début des séquences "left bounded"). Cette localisation DTW-LB permet une classification d'un cycle de production dès le début de son évolution. Ce sont l'attribution à un sous-ensemble de classes d'évolutions, la localisation DTW-LB et le prototype d'évolution des classes, qui permettent l'établissement d'une proposition de scénarios.

Cette localisation DTW-LB a la propriété de trouver la partie d'une séquence qui ressemble le plus à une sous-séquence donnée, tout en garantissant une souplesse dans l'alignement temporel. Elle a été développée dans le contexte de ce travail, analysée sur des données artificielles et appliquée aux données de l'installation. Cependant elle peut être utilisée pour toutes sortes de séquences, temporelles ou autres, pour lesquelles une souplesse de l'alignement peut être intéressante. Bien entendu d'autres méthodes avec des propriétés similaires existent, et il serait intéressant de les comparer à la "localisation DTW-LB". Cependant, en général les méthodes de localisation ne sont pas bornées, ce qui est une propriété qui n'est pour l'instant pas couverte par la méthode proposée. Mais il est fortement probable que la condition LB (left bounded) qui a été utilisée et qui est indisponsable dans le contexte de ce travail, peut être surmontée.

Un autre sujet qui a dû être résolu, afin de pouvoir réaliser une proposition de scénarios, est l'établissement des classes et prototypes d'évolutions, sur lesquels reposent la localisation DTW-LB et finalement la proposition de scénarios. Dans la section 3.3 de cette thèse, un classificateur non-supervisé, qui est basé sur une mesure de similarité, et qui est capable de traiter des séquences de taille arbitraire, a été développé. La mesure de similarité utilisée est la "distance" DTW, et elle réalise un alignement souple, qui est capable d'accomplir des compressions et étirements non-linéaires sur l'axe des temps. L'introduction de paramètres assure que cette mesure peut être adaptée à la subjectivité qui est liée à la similarité et aux domaines d'applications.

La classification non-supervisée utilise une approche basée sur des prototypes. Les prototypes initiaux utilisés pour l'apprentissage non-supervisé jouent, comme pour la plupart des approches basées sur des prototypes, un rôle important. Durant la phase d'apprentissage, les prototypes sont modifiés, afin de représenter au mieux les classes d'évolutions qui existent dans le corpus d'apprentissage. Cependant l'implémentation de cette modification est actuellement limitée à l'ajustement des amplitudes. La taille, c.-à-d. la longueur des prototypes, ce qui représente la durée d'un cycle de production, reste inchangée. Cela nécessite une initialisation représentative qui dépend des évolutions contenues dans le corpus d'apprentissage, même si l'alignement souple sur l'axe des temps compense cet impact.

Un apprentissage non-supervisé de la taille des prototypes, en même temps que sa forme, serait intéressant, afin d'améliorer la représentation d'une classe par son prototype, puisque en général les classes produites par cette méthode contiennent chacune des séquences de diverses tailles.

L'implémentation actuelle de l'apprentissage des prototypes se base sur un algorithme de mappage construit sur le calcul DTW, afin de ramener les séquences d'évolutions dans l'espace du prototype. Ce mappage est réalisé de manière pragmatique, et consiste à couper (enlever) les parties qui sont comprimées, et à dédoubler une valeur existante pour l'étirement. Une approche plus souple, qui analyse plus en détail la partie comprimée, respectivement le voisinage de la partie étirée serait envisageable. Une telle démarche ouvrirait à première vue deux perspectives : une première dans la souplesse de l'intervention réalisée, et

une deuxième qui résulterait dans la possibilité de créer un espace intermédiaire entre prototype et séquences mappées avec la possibilité d'adapter la taille des prototypes durant l'apprentissage.

Une fois que les séquences qui sont attribuées à une classe sont mappées vers l'espace du prototype, la modification de ce dernier est réalisée selon une approche similaire à l'algorithme des K-means respectivement des centres mobiles. De nouveau il s'agit d'une réalisation préliminaire de la classification non-supervisée, dans l'optique d'utiliser des méthodes connexionnistes dans le futur. Il est facilement imaginable d'étendre la méthode implémentée, en utilisant des approches de construction dynamiques de réseaux comme le "Growing Neural Gaz" ou des approches de cartographies comme les cartes auto-organisatrices "SOM", ce qui pourrait apporter des avantages dans le domaine de la convergence, ou réaliser une topologie des prototypes, respectivement adapter de façon non-supervisée le nombre de prototypes qui sont utilisés pour représenter au mieux la diversité du corpus analysé.

Notons encore que cette classification non-supervisée représente un moyen d'indexation des cycles de production sur base de leur l'évolution temporelle. Cette propriété n'est pas puisée dans le cadre de ce travail, mais pourait faciliter la recherche de cycles similaires en vue d'analyses et de recherches futures.

Ces méthodes de classification non-supervisée et de classification en ligne utilisant la localisation de sous-séquences, se basent sur le calcul du DTW dont le temps de calcul est d'ordre $O(n^2)$ par rapport à la taille des séquences. Cette propriété a conduit vers un autre domaine dans le traitement des données temporelles, notamment la réduction du volume de données. Dans la section 3.2 une méthode non-supervisée de codage est proposée. Elle se base sur une forme de codage bien connue, l'approximation par parties constantes (PAA), à laquelle est ajoutée une composante de forme, qui est extraite de manière non-supervisée du signal analysé. Ce codage a été appelé PPA (approximation par parties de formes). Une particularité de ce codage est que les codages PAA et PLA, qui utilisent des parties constantes respectivement linéaires pour l'approximation, y sont intégrés. Un déficit de cette étude est l'absence d'une comparaison fondée de ce codage avec d'autres types de codage, mais la diversité de sujets abordés s'opposait à cette démarche.

Concernant ces types de codage utilisant des sections de taille constante, une variante intéressante a récemment été proposée (APCA) [33], qui consiste à utiliser des sections de taille variable, et qui de ce fait adapte la précision du codage à la complexité du signal. Cela semble une approche particulièrement intéressante, et une première démarche pour le codage PPA pourrait être réalisée sans trop de modifications, en autorisant l'utilisation de formes issues de différents alphabets dans la phase d'encodage. Cela produirait en même temps une approche concurrentielle entre les formes des alphabets, et pourrait éventuellement être utilisé en tant que validation du codage afin de produire un alphabet global qui contiendrait des formes de différentes tailles. Une autre approche pourrait s'inspirer de la réalisation qui est utilisée dans APCA, à savoir la fusion consécutive des plages voisines, qui minimise l'augmentation de l'erreur d'approximation. Une méthode pour créer la forme primitive d'une plage fusionnée devrait être réalisée.

Toujours dans le contexte du codage, la réalisation d'une approche multivariable a été évoquée plusieurs fois déjà. Tous les outils utilisés pour le codage se basent sur des méthodes qui ont le potentiel nécessaire pour cette extension. Il suffirait, sans vouloir avancer que cela serait trivial, de réaliser une gestion du code (triplet) pour le cas multivariable, et d'adapter le calcul de distance locale dans l'algorithme DTW, afin d'étendre le codage à une approche multivariable.

Il faut, dans un cadre plus global de l'approche proposée, rappeler que la méthode de classification non-supervisée d'évolutions temporelles repose sur des exemples (des observations, des mesures). Dans le contexte du processus industriel cela mène à la conclusion, que seuls les cycles de production qui ont réellement été réalisés et qui sont stockés dans la base de données peuvent être pris en compte pour l'extraction de prototypes d'évolutions. En théorie, ceci veut dire que l'optimisation maximale qui pourrait être espérée, serait la production constante du meilleur résultat produit dans le passé (voir fig. 1.8). Il ne sera donc pas possible, en se basant sur cette méthode, de trouver une évolution innovatrice qui pourra baisser la consommation en-dessous de ce "meilleur" résultat. Cependant la nouvelle connaissance et la relation entre évolution et consommation pourra aider les experts dans leurs analyses et fournir un moyen pour faire des extrapolations en vue de trouver de nouveaux scénarios.

Notez aussi qu'ici l'évolution du processus a été attribuée au signal de la puissance électrique. Une question qui a été soulevée par les experts de ProfilARBED Recherche durant la réalisation manuelle de la catégorisation de cycles est : Qu'est-ce qui définit le comportement d'un processus ? Pour les experts il

est clair qu'un seul signal ne peut pas représenter la complexité du comportement de toute l'installation. Cependant, combien de signaux faut-il prendre en compte pour bien représenter une évolution temporelle, et quels sont les signaux les plus pertinents afin de représenter le comportement, tout en minimisant le nombre de données à traiter ? De quelle manière une validation du comportement représentée par ces signaux pourrait-elle être réalisée ? Des questions qui ouvrent un espace pour d'autres projets de recherche.

Directement raccordée aux questions précédentes est la problématique qui a déjà été soulevée : comment et avec quels critères une optimisation globale des alphabets (taille et nombre de formes), des prototypes de la classification (nombre et taille de prototypes) et des paramètres de la distance DTW (pénalité) est-elle réalisable ? Une approche pourra être l'incorporation des méthodes dynamiques dans la construction du classificateur non-supervisé tel que "Growing Neural Gaz" ou "Growing Grid" ce qui ont déjà été évoquées auparavant. L'optimisation est un domaine vaste et de nombreuses méthodes existent aussi pour des cas très complexes. Il est donc envisageable que pour ces questions des solutions peuvent être trouvées.

Globalement ce travail de thèse a produit différents outils plus ou moins génériques et utilisables dans le domaine du traitement de données temporelles et applicables pour l'extraction de connaissances de telles données, tout en tenant compte du fait que les phénomènes temporels ne sont pas rigides et qu'une certaine souplesse dans ces démarches est une propriété non négligeable. La somme de ces outils a permis une création de connaissances en forme d'évolution type pour les cycles de productions qui jusqu'à présent n'ont pas encore été analysés de cette façon.

Dans le cadre d'un projet de recherche appliquée, le CRP Henri Tudor va essayer d'utiliser cette approche d'extraction de connaissances et de prédiction dans un tout autre domaine que celui analysé ici. Il s'agit d'analyser le comportement d'une station d'épuration d'eaux usées qui représente un processus continu. Pour de telles installations, il est important de pouvoir prévoir le comportement de la station et de son environnement. Cela pourra se faire sur base de mesures du comportement du processus en définissant des cycles naturels comme la journée, la semaine etc. Dans ce contexte l'extention multivariable est considérée comme un des travaux les plus importants, et elle sera probablement réalisée durant ce projet.

Cette étude a abordé deux sujets complexes, à savoir la similarité et le traitement du temps dans le connexionnisme. Quelques outils ont été développés afin de résoudre les demandes et questions qui se sont posées durant ce travail. Même si de part et d'autres des implémentations sont loin ou ne frôlent que légerement le domaine du connexionnisme, ces travaux ouvrent la possibilité d'une incorporation dans des approches de ce domaine. Avec ce travail de nouvelles questions se sont posées. Pour quelques-unes, des premières pistes ont pu être proposées, et il est probable que de premières démarches vont être réalisées dans le cadre des projets futurs, tandis que d'autres seront rendues à la communauté scientifique.

Annexe A

Détails techniques du système d'acquisition et de gestion de données

Dans cette annexe sont regroupés les détails techniques du système d'acquisition mis en place sur le site de ProfilARBED Esch-Belval avec les détails concernant la gestion de données du côté CRP Henri Tudor.

A.1 Système d'acquisition sur site

L'expérience que ProfilARBED Recherche a faite avec l'implémentation de systèmes d'acquisition sur d'autres sites a conduit à un système basé sur une plate-forme de type PC industriel (Pentium III sous Microsoft Windows NT 4) utilisant le réseau "Ethernet Industriel"[14] qui était déjà en place sur le site de ProfilARBED Esch-Belval.

Les signaux de tous les capteurs sont récupérés et digitalisés par une infrastructure de bas niveau représentée par des automates programmables, en l'occurrence essentiellement des systèmes S5 et S7 de Siemens. La communication entre l'ordinateur du système d'acquisition et les automates programmables est réalisée à l'aide d'une carte de communication (PCI2000ETH) de la firme Applicom®. Cette carte, équipée de son propre processeur, garantit d'un côté le transfert de données via "Ethernet Industriel" à partir des automates programmables. De l'autre côté la communication avec les applications du système d'acquisition est réalisée sur la base d'un serveur OPC Applicom® utilisant l'API activeX Applicom®.

Les applications du système d'acquisition réalisé par ProfilARBED Recherche se basent sur l'environnement professionnel "INPRISE Borland Delphi 5".

Les données de type statique sont gérées dans un SGBD SyBase sur un système VAX. L'accès à cette base de données passe par des requêtes SQL et l'interface standard ODBC. Le logiciel utilisé chez ProfilARBED est MS-Access.

Le SGBD SyBase contient aussi une partie de données de type cyclique. L'échantillonnage se fait dans ce cas avec un intervalle de trois secondes mais les données ne sont gardées que pendant une courte durée afin de limiter les ressources nécessaires.

A.2 Gestion de données

A.2.1 Gestion primaire

Sur le système d'acquisition sur le site de production les données sont stockées dans des fichiers ASCII dans un format similaire au standard CSV (Comma Separated Value). Les valeurs sont cependant délimitées par tabulateurs ce qui est le type utilisé par défaut dans le logiciel MS-Excel. Pour des raisons de gestion et de manipulation de fichier, quatre fichiers sont créés par jour contenant six heures de données

[14] Nouveau nom du système de communication de Siemens connu sous l'ancien nom SINEC H1

chacun. Ces fichiers sont comprimés afin d'augmenter la capacité de stockage sur le système d'acquisition et pour accélérer le transfert des fichiers par le réseau informatique.

Parallèlement d'autres types de fichiers sont créés afin de simplifier l'accès et de faciliter l'utilisation aux données pour des analyses de différents niveaux qui sont réalisées par les différents acteurs humains. On y trouve des fichiers qui ne gardent que les données quand l'installation est sous puissance, ou des informations de bilans par charge pour n'en citer que quelques-uns. Pour ce projet ces fichiers ne jouent qu'un rôle secondaire.

Avec l'avancement du projet et pour diverses raisons, d'autres formes de mise à disposition des données ont été adoptées, notamment la gestion d'un certain nombre de données par un système de gestion de base de données (SGBD) installé sur le système d'acquisition. Le serveur MySQL pour la plate-forme MS-Windows est utilisé et tourne sur le système d'acquisition.

Une des raisons était la flexibilité d'accès aux données produites par les analyses réalisés sur le système d'acquisition. Entre autres la visualisation en ligne de résultats de ces analyses sur les postes de travail des opérateurs dans la salle de contrôle est réalisée en utilisant cet accès.

Une autre raison était la nécessité de découpler le processus d'acquisition du maniement de données du processus de traitement et de l'analyse de données qui coexistent sur le même système. Pour cette raison la base de données comporte un vecteur complet de données cycliques, brutes et calculées, qui est mis à jour par intervalle d'échantillonnage. Ainsi toutes les approches d'analyses ont la possibilité d'accéder aux variables cycliques du processus de façon autonome et asynchrone. Grâce à la structure client/serveur du SGBD et dû à la possibilité d'un accès par réseau, il est possible de distribuer les processus d'analyse ou encore de tester des nouvelles méthodes à distance à partir d'un poste de travail dans les bureaux de ProfilARBED Recherche.

A.2.2 Gestion secondaire

Du côté de la gestion secondaire, c.-à-d. la gestion des données au niveaux du CRP Henri Tudor les données numériques provenant de l'installation sont copiées périodiquement sur un disque portable (JAZ-Drive 2GB) et sauvegardées sur un Poste de travail au CRP Henri Tudor. Au début du projet avec des périodes plus ou moins régulières de deux semaines et ensuite dans des intervalles de quatre semaines. Finalement le transfert a été arrêté quand il a été jugé que la taille du corpus d'exemples était suffisamment grand. En parallèle, les données manuelles sont photocopiées et organisées dans un classeur afin de pouvoir vérifier et valider des comportements spéciaux détectés par des analyses.

Afin de mettre les données à disposition pour les diverses analyses, un SGBD MySQL est utilisé. Le tableau A.1 résume les données cumulées pendant la période du premier décembre 2000 au sept juillet 2001.

Nom de la base de données	**Nom du tableau**	**Nombre lignes**	**Taille Données [MB]**	**Taille Index [MB]**
B001_BPC_40_100 (01.12.2000 – 15.03.2001)	Calcdata_tbl	1478820	998,2	35,1
	Cyclicdata_tbl	1478820	2122,1	35,1
	Duree_fe_tbl	1332	0,6	0,03
	Temp_fe_tbl	1332	0,2	0,03
B001_BPC_40_101 (17.03.2001 – 07.07.2001)	Calcdata_tbl	1819911	1257,6	43
	Cyclicdata_tbl	1820112	2611,9	43,1
	Duree_fe_tbl	1309	0,6	0,03
	Temp_fe_tbl	1309	0,2	0,03

TAB. A.1 – Résumé de la base de données des données cycliques

Il a été essayé d'adopter une nomenclature utilisée par une des équipes de recherche du Centre dans le contexte du développement de logiciels. Cependant une extension de cette dernière était nécessaire puisque les bases de données n'y étaient pas spécifiées. La nomenclature spécifie, de gauche à droite : un identifiant du projet (B001), un acronyme du contenu (BPC : données provenant du système d'acquisition), un identifiant du type de format (40 : BD MySQL) et un numéro de version.

En raison de la taille de la base de données et parce que différentes adaptations étaient faites au système d'acquisitions concernant les données cycliques, deux versions de cette base de données ont été nécessaires.

Afin de contrôler l'accès aux données spécifiques du projet, deux types d'utilisateurs ont été créés : un utilisateur "lecteur" avec un droit d'accès restreint à la lecture sur les bases du projet, et un utilisateur "administrateur" avec le droit d'ajouter, de modifier ou de supprimer des bases de données, des tableaux ou des données dans les tableaux.

Concernant les protocoles de communications les deux types de méthodes, accès direct au SGBD par l'API et l'accès par protocole standardisé, ont été utilisées :

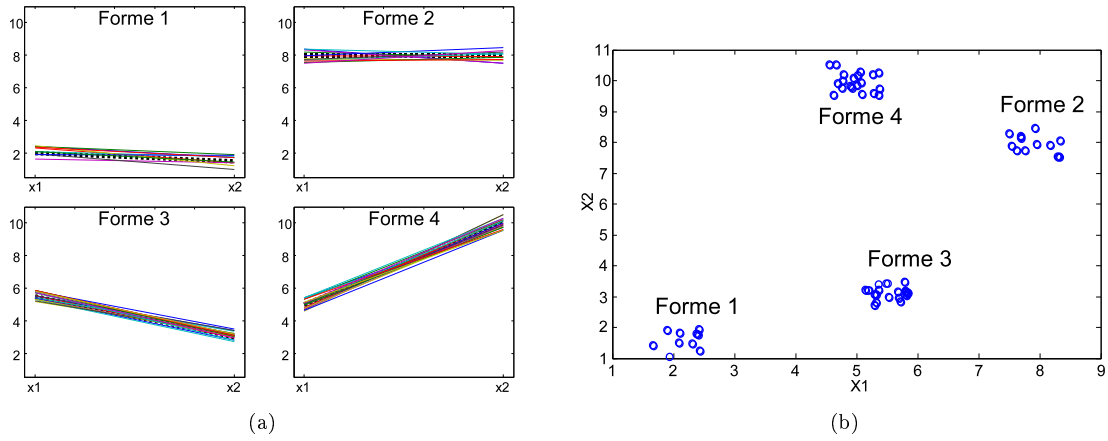
- L'insertion des données ASCII provenant de la gestion primaire dans la base de données est réalisée en utilisant les outils de base MySQL qui reposent sur l'API MySQL.
- Tous les programmes d'analyse reposent sur la plate-forme de développement MATLAB. Entre autres le développement de la connexion avec des SGBD repose sur la toolbox spécifique qui contient les outils pour l'accès aux bases de données. Cette boite à outils est compatible avec le protocole standard ODBC et avec la variante plus récente JAVA appelée JDBC. Pour des raisons pratiques seule la passerelle ODBC a été utilisée.

Annexe B

Analyse de l'apprentissage d'une SOM

Les cartes auto-organisatrices sont généralement utilisées pour créer de façon non-supervisée une cartographie deux voire trois dimensionnelle d'observations de dimensions plus élevées. La méthode essaye de respecter la topologie multidimensionnelle c.-à-d. les relations de proximités des observations qui existent dans l'espace d'origine. La méthode qui est utilisée repose sur une relation de voisinage entre les unités de la carte auto-organisatrice (chap.2.2.2.5).

Dans le contexte de la création des alphabets pour le codage les cartes auto-organisatrices sont utilisées pour l'extraction des formes primitives. Le but n'est donc pas directement la création d'une cartographie mais l'extraction des formes représentatives. Les paramètres classiques pour l'apprentissage d'une carte SOM ne sont pas appropriés pour cette tâche. Un exemple artificiel utilisant un corpus de formes dans l'espace R^2 représenter par un vecteur de deux dimensions $\vec{x} = [x_1, x_2]$ va illustrer la problématique.



(a) (b)

FIG. B.1 – Représentation du corpus artificiel avec ses quatre classes de formes 2D
(a) Le corpus est représenté dans l'espace des signaux
(b) Le même corpus dans l'espace des variables (chaque point représente une forme 2D)

La figure B.1 montre le corpus artificiel. Il est composé de quatre classes (Forme 1 - Forme 4), chacune comprenant des vecteurs $\vec{x} \in R^2$ qui représentent des lignes à deux points. L'exemple en deux dimensions peut être vu comme une extraction de formes primitives de taille $G = 2$ où le nombre de classes est connu ($C = 4$). De plus le corpus artificiel est créé de façon à ce que les classes soient bien séparées (fig. B.1(b)).

Les centres des classes sont à $C_1 = [2, 1.5], C_2 = [8, 8], C_3 = [5.5, 3], C_4 = [5, 10]$. Notez que le nombre d'individus dans les classes n'est pas uniforme ($N_1 = 10, N_2 = 12, N_3 = N_4 = 20$) ; par contre la distribution intra-classe est plus ou moins la même car elle est réalisée de façon aléatoire avec une variance de ± 1.

Une approche classique dans le contexte des cartes auto-organisatrices est la réalisation d'une cartographie 2D du corpus de l'exemple artificiel. Le résultat d'une telle cartographie, utilisant une carte à

$8x21 = 168$ unités sur la carte, est illustré dans la figure B.2. Notez que l'exemple représente un cas de figures spécial où la dimension des entrées (formes 2D) est la même que celle de la topologie de la carte (2D $8x21$).

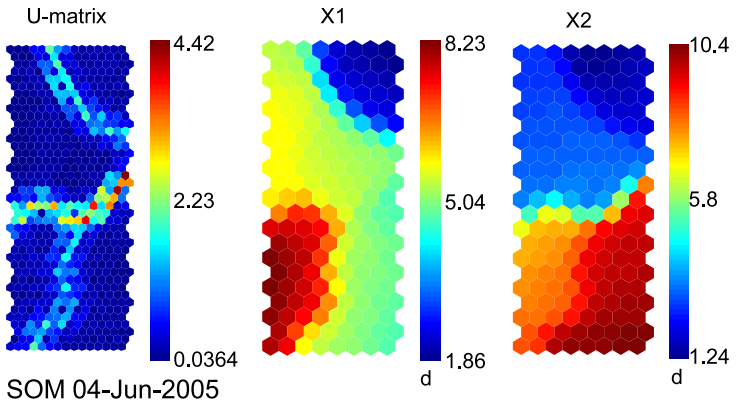


FIG. B.2 – Cartographie 2D du corpus artificiel utilisant une SOM

La figure montre une première carte (U-matrix) qui représente les distances entre les unités de la carte. Les parties plus claires indiquent des frontières de séparations (avec des distances plus importantes) entre des zones de la carte. Les deux autres cartes représentent la distribution des valeurs de chacune des deux variables sur la carte.

Les quatre classes sont clairement visibles (U-matrix) et il est possible d'attribuer chaque forme à une des zones de la carte. Par exemple : la "forme 1" en haut à droite sur la carte et la "forme 3" en haut à gauche.

Dans cet exemple une carte à 168 unités pour un corpus de 62 exemples a été apprise. Il est donc clair que cette carte est sur-dimensionnée pour le problème, mais la propriété de voisinage entre les unités de la carte produit des contraintes suplémentaires et de ce fait rend cette cartographie quand-même représentative pour les observations.

Cependant dans l'extraction des formes primitives le but est d'extraire du corpus un nombre réduit de prototypes représentatifs (formes caractéristiques) qui ne sont pas connus au préalable. En résumé on peut dire que non seulement les formes caractéristiques sont à extraire, mais aussi le nombre de prototypes qui représentent au mieux le corpus analysé n'est pas connu. Ce sujet est discuté dans l'annexe C.

Dans cette annexe le nombre de classes dans le corpus est connu et il s'agit d'utiliser l'algorithme SOM afin d'extraire les prototypes des classes. Pour cela noter que chaque unité de la carte représente un point dans l'espace des entrées, et de cette manière les unités bien placées au centre d'une classe représentent un prototype de cette classe.

Le nombre élevé d'unités de la carte rend difficile la sélection d'un prototype pour chacune des zones de la carte. De plus, avec le nombre d'unités de la SOM le temps de calcul pour la phase d'apprentissage augmente. L'idée est d'utiliser une carte avec un nombre réduit d'unités. De plus il est envisageable de réduire la SOM à une seule dimension. La carte se réduit alors à une ligne d'unités qui sont connectées chacun seulement aux deux voisins directs (sauf pour les deux unités du début et de la fin de la ligne seulement un voisin direct existe).

La figure B.3 montre la position des unités après apprentissage de cinq cartes unidimensionnelles de taille variée (Nombre d'unités entre 2 et 6). Les lignes qui relient les unités représentent les connexions entre les unités de la SOM. Cela donne une information sur la topologie des cartes unidimensionnelles.

Avec le nombre d'unités de la SOM, la représentation de la distribution du corpus artificiel s'améliore. Cependant même avec six unités, deux de plus que de classes dans le corpus, les unités ne sont pas utilisables en temps que prototypes des classes. Ce comportement s'explique avec la propriété de voisinage (forces d'attraction) entre les unités de la SOM. Le niveau de cette force d'attraction est paramétrisable en influençant la fonction de voisinage utilisée.

Classiquement une forme gaussienne (voir fig. 2.2(a)) est utilisée. Pour cette fonction (éq. (2.15)) le

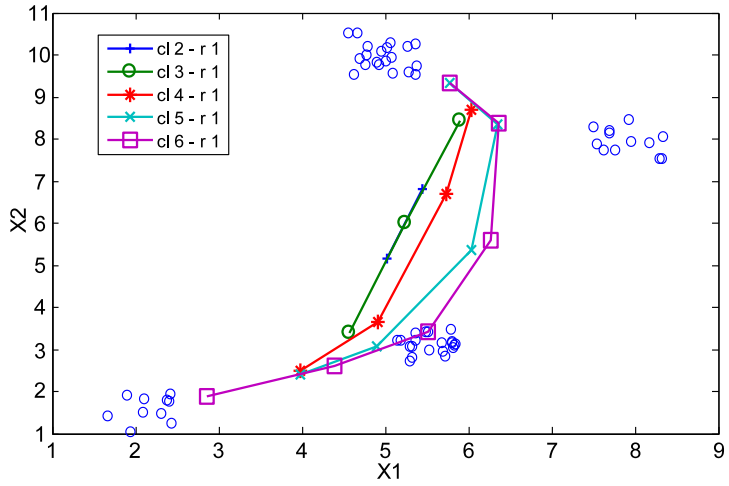


FIG. B.3 – Résultat de l'apprentissage de cinq SOM unidimensionnelles avec une variation du nombre d'unités de 2 à 6.

rayon d'influence $r = \sigma$ ainsi que la distance des unités au niveau de la carte d sont utilisés comme paramètres. Pour une carte unidimensionnelle la distance d est simplement la différence entre les index des unités sur la carte. La figure B.4 représente cette influence pour différents rayons.

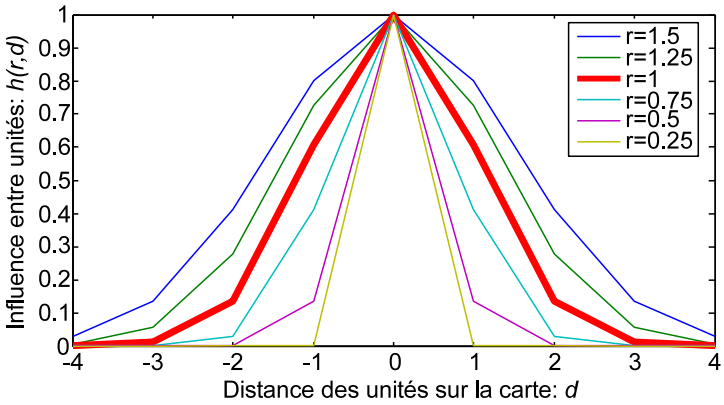


FIG. B.4 – Niveaux d'influence des unités de la carte en fonction de leur distance sur la carte et du paramètre rayon d'influence

Dans le chapitre 2.2.2.5 la méthode d'apprentissage est détaillée (éq. (2.13)- (2.16)) et il y est expliqué que le rayon d'influence est diminué avec l'avancement de l'apprentissage. Ici le rayon final de cet apprentissage est analysé.

Pour la figure B.3 un rayon de $r = 1$ (niveau par défaut) a été utilisé, ce qui explique les attractions entre les unités de la SOM.

Afin d'analyser l'influence de ce paramètre sur l'apprentissage de la SOM, une carte de quatre unités est apprise avec différents rayon d'influence. La figure B.5 montre le positionnement après apprentissage des quatre unités en fonction de ce rayon.

Il est clairement visible que en diminuant le rayon, les unités se positionnent de plus en plus vers le centre des classes. La position des unités de la SOM pour $(r = 0.25)$ est quasi au centre de chacune des quatre classes.

Afin de comparer le résultat avec celui obtenu pour les cinq SOM de taille différente (fig.B.3) le même calcul est réalisé pour un rayon final $r = 0.25$. La figure B.6 montre les positions des unités de ces cinq

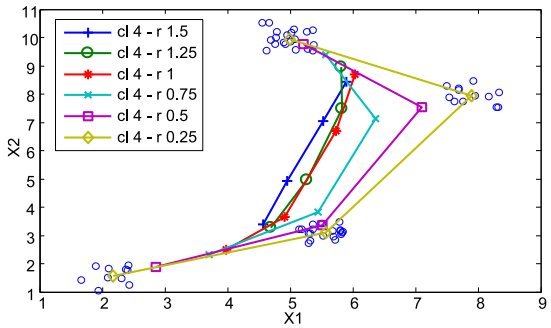


FIG. B.5 – Évolution des positions des prototypes (formes primitives) avec le paramètre de voisinage de la SOM

SOM unidimensionnelles (fig. B.3) après l'apprentissage.

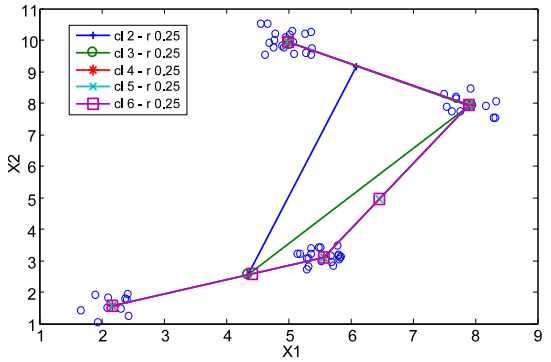


FIG. B.6 – Résultat de cinq SOM avec une variation du nombre d'unités de 2 à 6, avec un faible voisinage

Cette fois-ci les unités se positionnent bien dans les centres des regroupements. Même pour trois unités deux classes sont bien représentées. De plus la relation entre les unités n'est pas perdue, les lignes entre les unités représentent cette relation.

L'information de cartographie n'est pas complètement perdue car il est visible que les positions des unités sont influencées par la taille et la distance entre les regroupements.

- Pour le cas de deux unités les positions sont plus près des classes avec un nombre élevé d'individus (C_3, C_4).
- Pour le cas de trois classes les classes les plus proches (C_2, C_4) sont localisées et la troisième unité se trouve entre les deux autres mais, plus proche de celle avec le plus grand nombre d'individus.
- Pour le cas de cinq et six classes (une ou deux unités de plus que nécessaire) les unités libres sont positionnées entre deux classes reliées topologiquement. La position est influencée par la taille des classes et la distance entre les classes.

En utilisant une telle stratégie pour la construction des SOMs cette méthode est utilisable pour l'extraction de formes caractéristiques avec le prototype qui est représenté par la position des unités.

Annexe C

Validation “d’alphabets”

Dans le chapitre 3.2.2 le sujet de l’extraction de formes primitives dans le contexte du codage PPA est présenté. La méthode qui est présentée consiste à extraire les formes les plus caractéristiques appelées les formes primitives en se basant sur une classification non-supervisée. Chaque méthode de classification non-supervisée appliquée à un corpus quelconque de formes arrive à produire un regroupement. Cependant la qualité de la classification résultante dépend de la méthode et des paramètres utilisés.

Dans cette étude les prototypes d’une classification non-supervisée représentent un alphabet qui est utilisé dans la phase d’encodage. Chaque réalisation d’une classification produit un nouvel alphabet. Dans l’étude une approche multiéchelle est visée, ce qui fait que les deux paramètres suivants sont indépendants de la méthode de classification utilisée :

- la taille de la forme c.-à-d. la taille de la fenêtre qui est utilisée pour extraire les formes. Avec ce paramètre la granularité de l’approche multiéchelle est influencée, ce qui influence le taux de compression du codage.
- le nombre de formes qui vont être extraites. Ce paramètre influence le taux de filtrage réalisé localement. Avec peu de formes une plus grande généralisation est réalisée, ce qui implique en même temps une plus grande perte de détails.

Afin de limiter le nombre d’alphabets qui est utilisé et de garantir en même temps une qualité d’approximation du codage, il s’agit de trouver les meilleures classifications dans l’espace dressé par ces deux paramètres. En principe il s’agit de trouver les A meilleurs alphabet pour le codage des signaux.

La méthode utilisée pour la création des alphabets est une classification non-supervisée. Des méthodes sont analysées pour la validation de telles classifications. Afin d’expliquer et de visualiser la problématique, la figure C.1 illustre un problème unidimensionnel basé sur une distribution d’observations de la variable x.

Il s’agit de trouver le nombre de classes qui représente au mieux cette distribution. Pour cela il est important de définir un indicateur qui représente la qualité de la classification. Dans un grand nombre de cas la qualité de la classification est exprimée en analysant la compacité des classes et la distance entre les classes. Une classification avec des classes compactes et bien séparées les unes des autres est considérée comme discriminante.

De nombreuses analyses ont été faites sur ce sujet. On donne ici seulement deux références qui re-

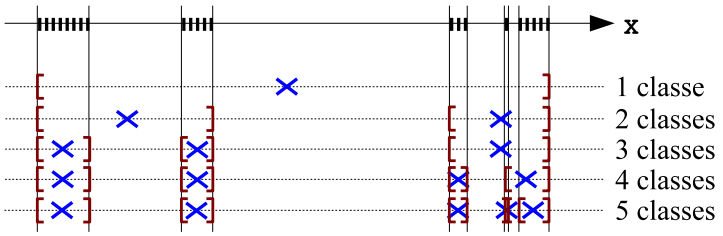


FIG. C.1 – Comportement d’une classification selon le nombre de classes.

groupent un grand nombre de méthodes et techniques utilisées dans ce domaine [27], [18]. Dans le contexte de ce travail deux méthodes de validation de classification ont été analysées et vont être détaillées dans cette annexe. Il s'agit d'une méthode basée sur la variance intra-classe cumulée et d'un index pour la validation de classifications qui a été proposé par Davies et Bouldin [10].

Finalement les deux méthodes présentées sont appliquées à un corpus artificiel et au calcul des alphabets issus du codage adapté aux données proposé dans cette thèse.

C.1 La variance intra-classe cumulée

Pour un corpus de données $\mathcal{X}$ il est possible de calculer la variance intra-classe cumulée. Elle est définie par l'équation (C.1).

$$cvar_{IC}(\mathcal{X}) = \sum_{C} var_{IC} \tag{C.1}$$

var_{IC} est la variance intra-classe qui est définie par :

$$var_{IC} = var(\mathcal{X}_C) \tag{C.2}$$

Avec $\mathcal{X}_C$ l'ensemble des données attribué à la classe C. Pour la classification on définit que chaque données du corpus $\mathcal{X}$ ne peut être attribué à une seule classe d'une classification (C.3) :

$$\mathcal{X} = \cup \mathcal{X}_C \quad \text{avec } \mathcal{X}_{Ci} \cap \mathcal{X}_{Cj} = \emptyset \quad \forall i \neq j \tag{C.3}$$

La variance d'un ensemble X est définie par :

$$var(X) = \sum_{X} (x - \mu)^2 P(x) \tag{C.4}$$

Dans les situations où la population n'est pas entièrement connue, ce qui veut dire que ni la moyenne μ ni la probabilité $P(x)$ ne sont connues, on se limite généralement à une estimation de la variance qui se base sur une estimation de moyenne donnée par les exemples connus. Cette estimation est définie en deux variantes : la variante biaisée (C.5) et la variante non-biaisée (C.6) (qui sera utilisée ici).

$$s^2(X) = \frac{1}{N} \sum_{X} (x_i - \bar{x})^2 \tag{C.5}$$

$$s^2(X) = \frac{1}{N-1} \sum_{X} (x_i - \bar{x})^2 \tag{C.6}$$

Où $\bar{x}$ est l'estimation de la moyenne de l'ensemble X qui elle est définie par :

$$\bar{x} = \frac{1}{N} \sum_{X} (x_i) \tag{C.7}$$

Cette définition de la variance intra-classe cumulée est utilisable pour la validation de classification monovariable, car la variance (C.4) n'est définie que dans cet espace.

Cependant dans le contexte de la validation des alphabets utilisés pour le codage adapté, une méthode multivariée est indispensable puisque une forme primitive a toujours une taille supérieure à un. De plus la taille des formes primitives n'est pas fixe à cause de l'approche multiéchelle.

La variance n'est cependant pas simplement extensible au cas multivarié. Dans [51] la norme $\|\Sigma\|$ de la matrice de covariance Σ est utilisée comme variance généralisée. Puisque la population des classes n'est pas entièrement connue une estimation de cette variance généralisée est notée par $\|\mathbf{S}\|$. Cette norme représente la compacité d'un corpus de données multivariées.

Il est donc nécessaire d'échanger la var_{IC} dans (C.1) et (C.2) par $\|\mathbf{S}_{IC}\|$ avec

$$\mathbf{S}_{IC} = cov(X_{IC}) = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1p} \\ S_{21} & S_{22} & \cdots & S_{2p} \\ \vdots & \vdots & & \vdots \\ S_{p1} & S_{p2} & \cdots & S_{pp} \end{bmatrix} \tag{C.8}$$

où S_{ij} est défini dans la variante non-biaisé

$$\begin{aligned} S_{ij} &= cov(x_i, x_j) \\ &= \frac{1}{N-1} \sum_X ((x_i - \bar{x_i})(x_j - \bar{x_j})) && \text{(C.9)} \\ &= \frac{1}{N-1} \left[\sum_X x_i x_j - \frac{1}{N} \sum_X x_i \sum_X x_j \right] && \text{(C.10)} \end{aligned}$$

La variance intra-classe cumulée devient donc

$$cvar_{IC}(\mathcal{X}) = \sum_C \|\mathbf{S}_{IC}\| \tag{C.11}$$

Pour un corpus $\mathcal{X}$ donné la variance intra-classe cumulée diminue en augmentant le nombre de classes. Une stagnation de cette évolution indique que l'augmentation du nombre de classes n'améliore pas pour autant la discrimination de la classification. Le point juste avant une telle stagnation, voire le premier point d'un tel plateau indique un nombre de classes intéressant. L'observation d'importantes chutes de la variance intra-classe cumulée suivies d'une stagnation sont les points les plus importants dans l'évolution de cet indicateur.

Il est cependant nécessaire de définir à partir de quand il est question d'une chute, respectivement d'une stagnation. Les deux paramètres sont dépendants de la nature du corpus et doivent être ajustés au mieux par un expert du domaine qui est analysé (voir chap. C.4).

C.2 L'index de Davies et Bouldin

La deuxième méthode analysée ici était proposée par deux chercheurs D.L Davies et D.W. Bouldin et se base sur une mesure de la compacité des classes en relation avec la distance entre les prototypes des classes. L'index est défini par l'équation (C.12).

$$DB = \frac{1}{C} \sum_{i=1}^{C} \max_{i \neq j} \left\{ \frac{S_n(X_i) + S_n(X_j)}{S(X_i, X_j)} \right\} \tag{C.12}$$

Avec C le nombre de classes. Au numérateur $S_n(X_C)$ est la distance moyenne entre les éléments de la classe "C" et le prototype respectif. Elle est défini par (C.13).

$$S_n(X_C)^2 = \frac{1}{N_C} \sum_{x \in X_c} \|x - C_C\|^2 \tag{C.13}$$

Avec X_C l'ensemble des éléments de la classe C et N_C le nombre d'éléments dans cette classe.

Au dénominateur, avec $S(X_i, X_j)$, se trouve la distance entre les prototypes des classes i et j définis par (C.14).

$$S(X_i, X_j) = \|C_i - C_j\| \tag{C.14}$$

La distance utilisée ici est la distance euclidienne (C.15)

$$\|A - B\|^2 = \sum_k (a_k - b_k)^2 \quad \text{avec } A, B \in \Re^k \tag{C.15}$$

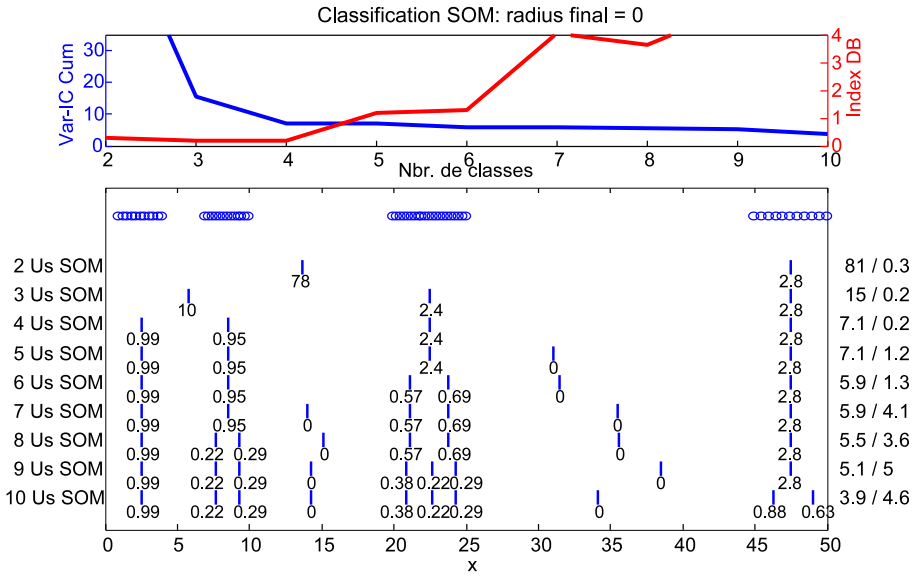


FIG. C.2 – Comportement des indices de validation pour un exemple 1D

Une extension au cas multivarié est donc triviale. De plus une normalisation selon les dimensions est réalisée par le rapport de la distance moyenne intra-classe sur la distance des prototypes.

La méthode cumule pour chaque classe i le maximum du rapport des moyennes des distances intra-classes avec la distance entre les prototypes. Ce maximum est obtenu pour deux classes proches dans l'espace $\Re^k$ et dont les étendues sont importantes, c.-à-d. deux classes qui sont difficilement séparables. La méthode n'est utilisable que pour un nombre de classes $C >= 2$.

En analysant différentes classifications réalisées avec un nombre croissant de classes, le minimum de cet indicateur indique la réalisation qui contient les classes les plus compactes et les plus séparées les unes des autres.

Cette méthode est utilisable s'il est possible de définir un prototype d'une classe. Pour toutes les méthodes basées sur des prototypes cela ne pose pas de problèmes ; pour les autres il est nécessaire de définir la méthode de calcul d'un prototype en se basant sur les membres d'une classe.

C.3 Exemples artificiels

Afin d'illustrer le comportement des deux méthodes, deux exemples artificiels sont utilisés. Afin de simplifier la représentation graphique, un premier exemple unidimensionnel est proposé (voir fig.C.1). Il s'agit d'une distribution d'observations d'une seule dimension qui est représentée sur l'axe des x (fig C.2).

Dans la figure C.2 deux axes sont représentés : l'axe supérieur montre l'évolution des deux indices avec le nombre de classes, l'axe inférieur montre tout d'abord les observations selon l'axe des x (petits cercles bleus) et en-dessous les positions des prototypes des classes pour les différentes réalisations de classification. Du côté gauche le nombre d'unités SOM utilisées sont données pour chaque réalisation, et à droite les valeurs des deux indices de validation ("variance intra-classe cumulée" / "Index de Davies et Bouldin") sont affichées. En-dessous du marqueur de chaque prototype la valeur de la variance intra-classe est affichée.

Les valeurs représentent quatre groupes plus ou moins compacts avec différents nombres d'observations par groupe. La classification non-supervisée est réalisée par une carte auto-organisatrice (SOM) en utilisant la même méthode qui est utilisée pour l'extraction des formes primitives, c.-à-d la création des alphabets (voir annexe B pour plus de détails).

Pour la variance intra-classe cumulée, la première stagnation dans l'évolution commence pour $C = 4$. Les diminutions pour les valeurs de $C > 4$ ne sont que relativement faible. Cependant déjà le gain entre

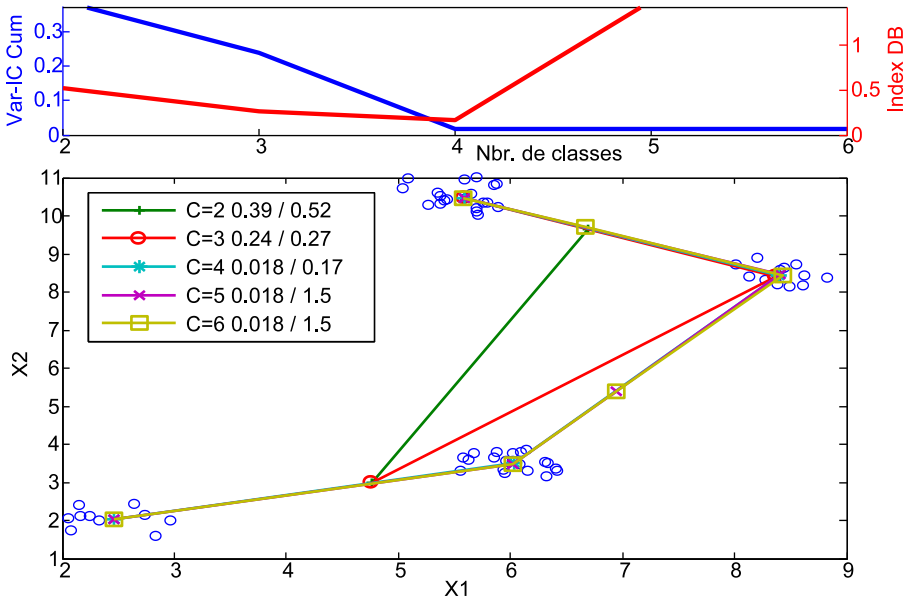


FIG. C.3 – Comportement des indices de validation pour un exemple 2D

la classification réalisée avec $C = 3$ et $C = 4$ n'est plus très élevé.

En analysant l'index DB le minimum est réalisé pour $C = 4$, mais la différence de l'index DB entre les réalisations $C = 3$ et $C = 4$ n'est que très faible.

Pour les deux indicateurs de validation, la meilleure classification est obtenue pour quatre classes, mais le résultat pour trois classes est envisageable.

Un deuxième exemple de groupements de données dans l'espace de deux dimensions (X_1 et X_2) est représenté dans la figure C.3. Le calcul de la variance intra-classe cumulé se base sur la norme de la matrice de covariance.

Dans cette représentation les différentes réalisations pour $C = 2$ jusqu'à $C = 6$ sont superposées. Les couleurs et symboles séparent les différents prototypes qui, pour des causes de visibilité, sont reliés entre eux par des lignes. En même temps ces lignes représentent la relation des unités SOM sur la carte unidimensionnelle (voir Annexe B). Dans la légende de la figure les valeurs des deux indices de validation ("variance intra-classe cumulée" / "Index de Davies et Bouldin") sont reprises.

Pour cet exemple aussi les deux indices indiquent que la meilleure classification est produite pour $C = 4$ et retrouvent donc aussi le nombre de groupes qui existent dans le corpus de données.

C.4 Application à la validation des alphabets

Afin d'appliquer ces méthodes à la validation des alphabets, l'approche suivante a été utilisée. Dans la phase 1a (fig 3.1) un grand nombre d'aplhabets est produit en variant les deux paramètres principaux de l'extraction de formes primitives à savoir la taille de la forme (G) et le nombre de formes (C) dans un alphabet. Une grille complète de ces deux paramètres est réalisée pour G allant de 6 à 60 par intervalle de 6 et pour C de 3 à 30 par intervalle de 1. L'intervalle de la taille de formes était choisi plus grand afin de limiter le temps de calcul et le volume de données à manipuler, car depuis le début une approche multiéchelle était visée, ce qui correspond à une sélection de plusieurs tailles. Le but de la validation est double : d'abord il faut trouver les échelles qui semblent les plus pertinentes, et en même temps pour ces échelles trouver le nombre de formes qui représentent au mieux cette échelle. Pour la production des alphabets 50% des signaux du corpus ont été sélectionnés de façon aléatoire.

Dans la phase 1b (fig 3.1) tous les signaux ont été encodés sur base de tous ces alphabets. Afin d'éviter que le corpus d'apprentissage ne soit utilisé pour la validation les autres 50%, des signaux ont été utilisés. L'encodage représente une classification dans le sens d'une attribution d'une partie de signal à une classe

c.-à-d. une forme primitive d'un alphabet. Les informations de cette classification sont utilisées pour la validation.

A cause du nombre élevé de signaux respectivement de parties des signaux qui sont attribués aux formes primitives, un calcul de validation a posteriori tel qu'il était utilisé pour les exemples n'est pas réalisable. Une méthode incrémentale pour le calcul des indices de validation était nécessaire.

Afin de pouvoir calculer les deux indices de validation, les variables suivantes sont gérées pour chaque classe de chaque alphabet durant le codage :

1. Le nombre d'éléments attribué à la classe. Pour chaque élément attribué à une classe cette variable est incrémentée
   $N_C(k) = N_C(k-1) + 1$
2. Un vecteur avec la somme des éléments attribuées à une classe
   $A_C(k)[i] = A_C(k-1)[i] + x_i \ \ \forall i \in R^p$
3. Une matrice (symétrique) avec la somme des combinaisons des éléments i, j attribués à une classe pour la matrice de covariance
   $B_C(k)[i,j] = B_C(k-1)[i,j] + x_i x_j \ \ \forall i,j \in R^p$
4. et la somme des distances entre le prototype et l'élément attribué à une classe
   $D_C(k) = D_C(k-1) + \|x - C_C\|^2$

Après le codage les indices de validation sont calculés de façon suivante en se basant sur les variables N_C, A_C, B_C et D_C :

S_{ij}, utilisé dans le calcul de la variance intra-classe cumulé (C.11), est calculé par :

$$S_{Cij} = \frac{1}{N_C - 1}\left[B_C[i,j] - \frac{1}{N_C} A_C[i] A_C[j]\right] \ \ \forall i,j \in R^p \tag{C.16}$$

$Sn(X_C)$ utilisé pour l'index de Davise et Bouldin (C.13) se calcule par :

$$S_n(X_C) = \sqrt{\frac{D_C}{N_C}} \tag{C.17}$$

Il est donc possible de calculer pour chaque alphabet correspondant à une paire de paramètres (G,C), avec G la taille de la forme primitive et C le nombre de formes dans l'alphabet, une valeur pour les deux indicateurs. Le résultat est représenté dans les figures C.4(a) et C.4(b).

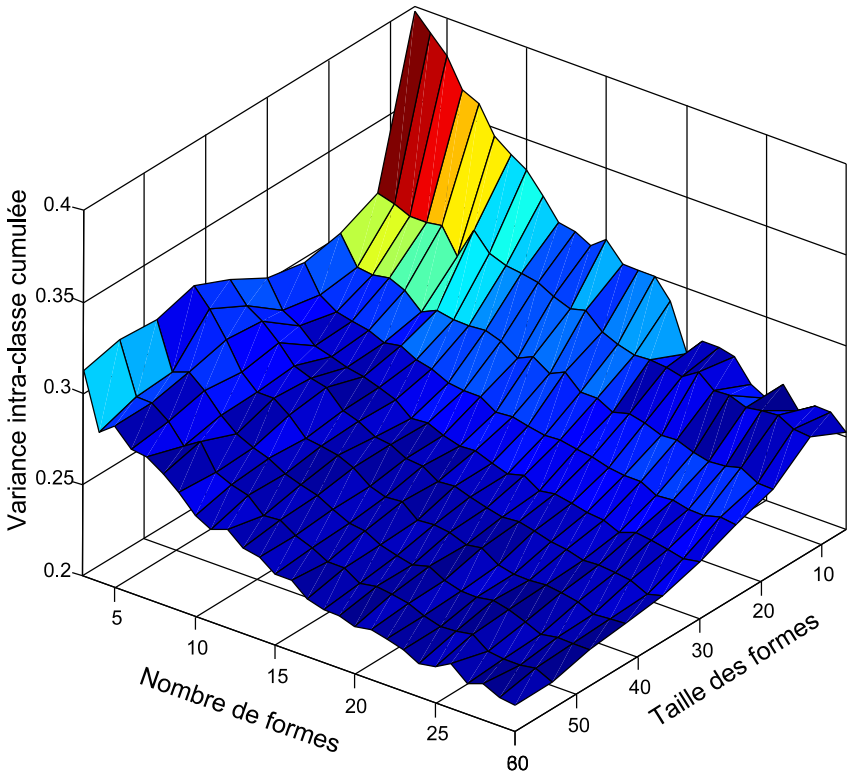
Un premier constat est que les comportements des deux indices ne sont pas aussi cohérents que dans les deux exemples 1D et 2D. Mais en globalité il est possible de dire que la plage des alphabets avec $G > 24$ & $C > 15$ est inintéressante dans les deux cas.

Concernant l'index Davise-Bouldin, où les valeurs minimales représentent de bonnes classifications, les alphabets avec $G > 24$ ont le minimum pour des nombres de classes ($C <= 7$). Le meilleur alphabet, celui qui représente le minimum global, correspond aux paramètres $G = 60$ et $C = 6$. Aussi l'alphabet avec $G = 6$ obtient de faibles valeurs pour cet index. Pour cette échelle le nombre de classes à prendre en compte est $C = 12$.

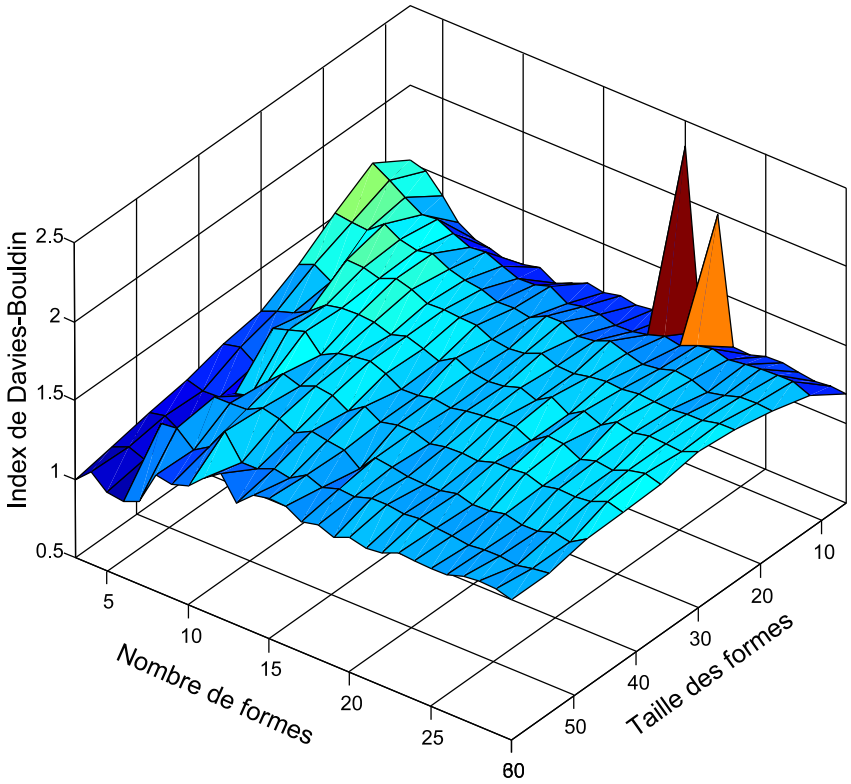
En analysant la variance intra-classe cumulée, pour laquelle la pente et les positions de paliers locaux indiquent les alphabets intéressants, seulement pour $G >= 48$ les petits nombres de classes $C = 4$ seraient intéressants. Les alphabets les plus intéressants selon cet indicateur sont dans le domaine $18 <= G <= 24$ et $4 <= C <= 8$, car les pentes y sont les plus importantes en comparaison avec les paliers qui suivent ces pentes. Analysons encore l'échelle $G = 6$ afin de comparer ce résultat avec l'index Davise-Bouldin. Pour $C = 12$ un petit palier peut être détecté si l'on se limite à l'axe du nombre de formes, ce qui correspond au même alphabet indiqué par l'index Davise Bouldin.

Pour la suite de l'étude une sélection de quatre alphabets a été décidée. Il n'est cependant pas trivial, en nous basant sur ces indicateurs, de clairement identifier les alphabets les plus pertinents. Mais les indicateurs peuvent donner une orientation.

Afin de garantir une approche multiéchelle, les quatre alphabets seront sélectionnés dans des échelles différentes. Un premier alphabet pour $G = 6$ & $C = 12$, où les deux indicateurs semblent plus ou moins cohérents. Un deuxième pour $G = 24$ & $C = 5$, qui est le plus intéressant selon la variance intra-classe cumulé. Un troisième à $G = 42$ & $C = 6$ et un dernier qui correspond au minimum de l'index Davise-Bouldin avec $G = 60$ & $C = 6$.



(a)



(b)

FIG. C.4 – Résultat des calculs d'indices de validation pour les 280 alphabets : (a) surface représentant la variance intra-classe cumulée (b) surface représentant l'index de Davise et Bouldin.

Table des figures

Liste des tableaux

Glossaire

Ce glossaire ne reprend que les abréviations spécifiques utilisées dans ce document.

APCA : angl. Adaptive Piece-wise Constant Approximation

BMU : Prototype gagnant dans une approche d'apprentissage compétitive (angl. Best Mapping Unit)

CSV : Valeurs séparées par virgules ; angl. Comma Separated Value

DTW : Dynamic Time Warping (eng.), cheminement temporel dynamique
DTW-LB : DTW bornée à gauche. Angl. : Dynamic Time Warping Left Bounded

GDM : matrice globale des distances cumulées (angl. Global Distance Matrix)

LDM : matrice des distances locales (angl. Local Distance Matrix)

MF : Moyenne Flottante (angl. Moving Average)
MLP : Perceptron multicouche. Angl. Multi Layer Perceptron

PAA : angl. "Piecewise Aggregat Approximation" ou encore "Piecewise Constant Approximation" (PCA)
PLA : angl. Piecewise Linear Approximation
PPA : Piecewise Pattern Approximation

SGBD : Système de gestion de bases de données

TF : Transformée de Fourier

UdtwC : Classification non-supervisée basée sur la distance DTW, angl. Unsupervised DTW-based Clustering

Bibliographie

[1] J.B. Allan and L.R. Rabiner. A unified approach to short time fourier analysis and synthesis. *Proc. IEEE*, 65 :1558–1564, 1977.

[2] J.-C. Baumert, J.-L. Rendueles Vigil, P. Nyssen, J. Schaefers, G. Schutz, and S. Gillé. Improved control of electrical arc furnace operations by process modelling. Technical steel research series EUR 21411, European Commission, Luxembourg EC publication office, 2005.

[3] R. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

[4] R. Bellman and S. Dreyfus. *Applied Dynamic Programming.* Princeton University Press, Princeton, New Jersey, 1962.

[5] D. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the KDD Workshop*, pages 359–370, Seattle, July 1994.

[6] Laurent Bougrain. *Étude de la construction par réseaux neuromimétiques de représentations interprétables : Application à la prédiction dans le domaine des télécommunications.* PhD thesis, Université Henri Poincaré - Nancy 1, 2000.

[7] Michael H. Brackett. *Data sharing using a common data architecture.* Wiley Professional Computing. Katherine Schowalter by John Wiley & Sons, Inc, 1994.

[8] C.H. Chen, editor. *Fuzzy Logic And Neural Network Handbook.* IEEE Press, 1996.

[9] G. Cybenko. Approximation by superpositions of a sigmoid function. *Mathematics of Control, Signals and Systems*, 2 :303–314, 1989.

[10] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1(2) :224–227, April 1979.

[11] Stéphane Durand. *TOM, une architecture connexionniste de traitement de séquences. Application à la reconnaissance de la parole.* PhD thesis, Université Henri Poincaré - Nancy 1, 1995.

[12] J.L. Elman. Finding structure in time. *Cognitive Science*, 14 :179–211, 1990.

[13] J.-P. Haton et M.-C. Haton. L'intelligence artificielle. *Que sais-je ?*, 2444, 1989.

[14] M.A. Fischler and O. Firschein. *Intelligence : The Eye, the Brain, and the Computer.* Addison-Wesley Publishing Co., Boston, MA, 1987.

[15] B. Fritzke. Some competitive learning methods, 1997.

[16] S. Gillé and G. Schutz. Improved control of electric arc furnace operations by process modelling, chap 2. ECES 7210-PR-12 - Fifth Technical Report, March 2002.

[17] M.T. Hagan, H.B. Demuth, and M. Beale. *Neural Network Design.* PWS Publishing Comp., 1996.

[18] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3) :107–145, 2001.

[19] Babak Hassibi and David G. Stork. Second order derivatives for network pruning : Optimal brain surgeon. In Stephen José Hanson, Jack D. Cowan, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 164–171. Morgan Kaufmann, San Mateo, CA, 1993.

[20] M. H. Hassoun. *Fundamentals of Artificial Neural Networks.* The MIT Press, Cambridge, Massachusetts, London, England, 1995.

[21] S. Haykin. *Neural Networks, A comprehensive Foundation*. Prentice-Hall, New Jersey, 1999.

[22] D.O. Hebb. *The Organization of Behavior : A Neuropsychological Theory*. Wiley, New York, 1949.

[23] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*. Addison–Wesley, Redwood City, 1991.

[24] J.J. Hopfield and D.W. Tank. Computing with neural circuits : A model. *Science*, 233 :625–633, 1986.

[25] K. Hornik, M. Stinchcombe, and H. White. Multi-layer feedforward networks are universal approximators. *Neural Networks*, 2 :359–366, 1989.

[26] W.T. Miller III, R.S. Sutton, and P.J. Werbos, editors. *Neural Networks for Control*. The MIT Press, Cambridge, Massachusetts, London, England, 1996.

[27] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering : a review. *ACM Computer Surveys*, 1999.

[28] M.I. Jordan. Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the Eighth Annual Cognitive Science Society Conference*, Hillsdale, 1986. Erlbaum.

[29] Jean-Daniel Kant. *Modélisation et mis en oeuvre de processus cognitifs de catégorisation à l'aide d'un résau connexionniste*. PhD thesis, Université de Rennes I, 1996.

[30] Dr Eamonn Keogh. A tutorial on indexing and mining time series data, 2002.

[31] E. Keogh. Exact indexing of dynamic time warping. In *In 28th International Conference VLDB*, pages 406–417, Hong Kong, 2002.

[32] E. Keogh, K. Chakrabarti, M. Pazzani, and Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Journal of Knowledge and Information Systems*, pages 263–286, 2000.

[33] E. Keogh, K. Chakrabarti, M. Pazzani, and Mehrotra. Locally adaptive dimensionality reduction for indexing large time series databases. In *In Proc of ACM SIGMOD Conference on Management of Data*, pages 151–162, 2001.

[34] S. Köhle. Einflussgrößen des elektrischen energieverbrauchs und des elektodenverbrauchs von lichtbogenöfen. *Stahl u. Eisen*, 1992.

[35] S. Kim, S. Park, and W. Chu. An index-based approach for similarity search supporting time warping in large sequence databases. *ICDE*, 2001.

[36] T. Kohonen. *Associative memory : a system theoretic approach*. Springer-Verlag, Berlin, 1978.

[37] T. Kohonen. Self-organized formation of topological correct feature maps. *Biological Cybernetics*, 43 :59–69, 1982.

[38] M. Krishnan, C.P. Neophytou, and Glenn Prescott. Wavelet transform speech recognition using vector quantization, dynymic time warping and artificial neural networks. Centre for Excellence in Computer Aided System Engineering and Telecommunication & Information Sciences Loboratory.

[39] Y. Lallement, M. Hilario, and F. Alexandre. Neurosymbolic integration : cognitive grounds and computational strategies. In *Actes du congrès WOCFAI*, Paris, 1995.

[40] B.P. Lathi. *Signal Processing & Linear Systems*. Berkeley, Cambridge, 1998.

[41] Y. LeCun, J. Denker, S. Solla, R. E. Howard, and L. D. Jackel. Optimal brain damage. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems II*, San Mateo, CA, 1990. Morgan Kauffman.

[42] M. Lemasson. Etude de séries temporelles à l'aide de cartes auto-organisatrices. Mémoire de maîtrise, UFR-MIM Université de Metz, Août 2001.

[43] Y. Linde, A. Buzo, and R.M.Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communication*, 1980.

[44] Lennart Ljung. *System identification. Theory for the user*. Prentice Hall PTR, 1999.

[45] D. Marcus. Détection et extraction d'événements dans un flux de données temporel. Mémoire de Maîtrise, Mathématiques Discrètes, Université Louis Pasteur de Strasbourg, Octobre 2002.

[46] T.M. Martinetz and K.J.Schulten. *Artificial Neural Networks*, chapter A "neural-gas" network learns topologies, pages 397–402. North-Holland, Amsterdam, 1991.

[47] J.L. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5 :115–133, 1943.

[48] C. S. Myers and L. R. Rabiner. A level-building dynamic time warping algorithm for connected word recognition. *IEEE Trans. Acoust., Speech, Signel Processisng*, 29 :284–297, April 1981.

[49] A.V. Oppenheim, A.S. Willsky, and I.T. Young. *Signals and Systems*. Prentice-Hall Inc., New Jersey, 1983.

[50] Duc Truong Pham and Liu Xing. *Neural networks for identification, prediction and control*. Springer, 1995.

[51] Alvin C. Rencher. *Methods of Multivariate Analysis*. 1995.

[52] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(1) :43–49, February 1978.

[53] N. Schaaf. Classification de sous-séquences de signaux temporels, en se basant sur des modèles prédéfinis de séquences. Mémoire de maîtrise, MIM Université de Metz, Août 2003.

[54] G. Schutz, F. Alexandre, and S. Gillé. Neural networks in dynamic process analysis. *Proceedings of IAR/ICD Workshop on* 16^{th} *IAR anual meeting*, pages 105–110, november 2001.

[55] G. Schutz, F. Alexandre, and S. Gillé. Unsupervised extraction of temporal evolution patterns on an electric arc furnace process. *Preprints of the* 11^{th} *IFAC MMM Symposium*, september 2004.

[56] A. Tarsitano. A computational study of several relocation methods for k-means algorithms. *Pattern Recognition*, 36 :2955–2966, april 2003.

[57] O. Tim, L. Firoiu, and P. Cohen. Clustering time series with hidden markov models and dynamic time warping, 1999.

[58] E.R. Tufte. The visual display of quantative information. *Graphics Press*, 1983.

[59] T. K. Vintsyuk. Speech discrimination by dynamic programming. *Kibernetika*, 4 :81–88, January-February 1968.

[60] D.J. Willshaw and C. von der Malsburg. How patterned neural connections can be set up by self-organization. In *Proc. R. Soc. Lond. B.*, volume 194, pages 431–445, 1976.

[61] Stuart N. Wrigley. Speech recognition by dynamic time warping, 1998.

[62] B. Yi, H. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. *ICDE*, pages 23–27, 1998.

[63] B.K. Yi and C. Faloutsos. Fast time sequence indexing for arbitrary lp norms. In *Procceedings of the 26st Intl Conference on Very Large Databases*, pages 385–394, 2000.

Résumé

Cette étude consiste à étudier l'apport de réseaux de neurones artificiels pour améliorer le contrôle de processus industriels complexes, caractérisés en particulier par leur aspect temporel. Les motivations principales pour traiter des séries temporelles sont la réduction du volume de données, l'indexation pour la recherche de similarités, la localisation de séquences, l'extraction de connaissances (data mining) ou encore la prédiction.

Le processus industriel choisi est un four à arc électrique pour la production d'acier liquide au Luxembourg. Notre approche est un concept de contrôle prédictif et se base sur des méthodes d'apprentissage non-supervisé dans le but d'une extraction de connaissances.

Notre méthode de codage se base sur des formes primitives qui composent les signaux. Ces formes, composant un alphabet de codage, sont extraites par une méthode non-supervisée, les cartes auto-organisatrices de Kohonen (SOM). Une méthode de validation des alphabets de codage accompagne l'approche.

Un sujet important abordé durant ces recherches est la similarité de séries temporelles. La méthode proposée est non-supervisée et intègre la capacité de traiter des séquences de tailles variées.

Mots-clés: connexionnisme, réseaux de neurones, apprentissage non-supervisé, carte auto-organisatrice, dynamic time warping DTW, similarité

Abstract

This study is interested in analyzing the contribution of artificial neural networks in order to improve the control of complex industrial processes that are mainly characterized by their temporal behavior. The main motivations of the time series analysis are data reduction, indexation based on similarity, localization of sequences, knowledge extraction and prediction.

The analyzed industrial process is an electric arc furnace for the liquid steel production in Luxembourg. The proposed approach is a concept of predictive control based on unsupervised learning techniques with the aim of knowledge extraction.

Our signal coding method is based on primitive patterns that compose the signals. These patterns, building the coding alphabet, are extracted using an unsupervised method, the self organizing maps of Kohonen (SOM). An alphabet validation approach is proposed.

One of the important subjects of this research is the similarity of time series. The proposed method is unsupervised and able to handle sequences of arbitrary size.

Keywords: connectionist, neural networks, unsupervised learning, self organising maps, dynamic time warping, similarity