AIX-MARSEILLE UNIVERSITÉ

École Doctorale en Mathématiques et Informatique de Marseille - ED184

Laboratoire des Sciences de l'Information et des Systèmes - UMR CNRS 7296

Thèse présentée pour obtenir le grade universitaire de docteur

Discipline : Mathématiques et Informatique
Spécialité : Automatique

Julien MARINO

Diagnostic des équipements de production de semi-conducteurs par analyse statistique

Soutenue le JJ/MM/AAAA devant le jury composé de :

Didier THEILLIOL	Université de Lorraine	Rapporteur
Zineb SIMEU	Université Grenoble-Alpes	Rapporteur
Belkacem OULD BOUAMAMA	Université de Lille 1	Examinateur
Mitra FOULADIRAD	Université de Technologie de Troyes	Examinateur
Mustapha OULADSINE	Aix-Marseille Université	Directeur de thèse
Francesco ROSSI	Aix-Marseille Université	Co-directeur de thèse
Jacques PINATON	STMicroelectronics Rousset	Encadrant industriel

Numéro national de thèse/suffixe local : 2017AIXM0599/043ED184

Table des matières

Table des figures

Liste des tableaux

Résumé

Cette thèse présente la construction d'un algorithme de détection de fautes utilisable en temps réel dans l'industrie du semi-conducteur. La méthodologie proposée est basée sur l'exploitation statistique des données récoltées sur les équipements. La présence de fautes est évaluée de façon à la fois supervisée, par rapport au passé des équipements, et non-supervisée, par rapport aux données du présent de l'ensemble des équipements d'un même atelier de production. L'approche proposée est testée sur des équipements de STMicroelectronics, à partir de fautes ayant réellement eu lieu dans l'environnement de production. Cette thèse a été réalisée dans le cadre d'une collaboration CIFRE entre l'Université d'Aix-Marseille et l'entreprise STMicroelectronics Rousset.

Nous modélisons chaque équipement et chacun de ses modes de fonctionnement de façon individuelle, à partir de données de référence. Nous effectuons un test statistique, basé sur ce modèle, à chaque fois qu'un procédé est exécuté. Ce test est nommé Gaussian Time Error, car il est basé sur une modélisation gaussienne à chaque instant de mesure. Il permet de se prononcer sur la qualité de l'équipement par l'identification de différences relativement au modèle.

Une particularité de l'industrie du semi-conducteur est la maintenance régulière des équipements, qui apporte des variations par rapport à la modélisation initiale. Nous introduisons donc un test spécifique sur ces maintenances, pour vérifier que l'intervention humaine n'ait pas introduit d'erreur. Une adaptation de certains paramètres du Gaussian Time Error a lieu lorsqu'aucune faute n'est détectée. Cette procédure s'effectue sur une quantité faible de données, permettant une reprise rapide de la surveillance des équipements.

En parallèle de la surveillance individualisée de chaque équipement, nous proposons une étude commune, non-supervisée, de tous les appareils réalisant les mêmes procédés. Notre algorithme de comparaison d'équipements permet de distinguer d'éventuelles machines atypiques, sur la base de la forme de courbes représentatives. Une cohérence globale du comportement des outils de production est ainsi assurée à l'échelle de l'usine.

Mots-clés : Détection de fautes, apprentissage statistique, semi-conducteurs, maintenances.

Abstract

This thesis presents the construction of a fault detection algorithm, which can be used in real-time for semiconductor processes. The methodology is based on the statistical analysis of the data collected on the equipments. Faults are detected both with supervised learning, relatively to past data of a given equipment, and with unsupervised learning, relatively to the present data of groups of equipments. The proposed approach is tested on real equipments from STMicroelectronics, using faults which effectively occurred in the industrial environment. This thesis was made in collaboration between Aix-Marseille Université and STMicroelectronics Rousset.

Each equipment and each of its operating modes is individually modeled, using reference data. A statistical test, based on such model, is performed every time a process is realized. This test is named Gaussian Time Error, because it is based on Gaussian models computed at each measurement time. By identifying differences in the new data relatively to the model, it allows to detect faults on the equipments.

In the semiconductor industry, equipements are frequently maintained, which modifies the observed data. As a result, we introduce a specific test for the maintenances, to verify that no human error was made during the intervention. When no fault is seen, several parameters of the Gaussian Time Error are updated. This procedure is performed on a small quantity of data, thus allowing the equipment to be monitored quickly after maintenance.

On top of the individual monitoring of equipments, we propose to gather the data of all equipments performing identical processes, in order to provide an unsupervised fault detection method. This equipment matching algorithm allows to detect atypical process chambers, using a comparison of representative shapes. This method allows for an harmonization of the behavior of groups of equipments.

Keywords : Fault detection, statistical learning, semiconductors, maintenance.

Publications

Articles de conférence publiés

— J. Marino, F. Rossi, M. Ouladsine, and J.Pinaton. Gaussian Time Error : a fault detection index for semiconductor processes. In *American Control Conference (ACC)*. IEEE, 2016

— J. Marino, F. Rossi, M. Ouladsine, and J. Pinaton. Unsupervised semiconductor chamber matching based on shape comparison. In *IFAC World Congress*. IFAC, 2016

Article de journal

— J. Marino, F. Rossi, M. Ouladsine, and J. Pinaton. A maintenance-adaptive fault detection framework for semiconductor processes using Gaussian Time Error. *Submitted*, 2016 (en phase de révision pour le journal IEEE Transactions on Semiconductor Manufacturing)

Articles soumis

— J. Marino, F. Rossi, M. Ouladsine, and J. Pinaton. Gaussian Time Error : a statistical test for time series under time-shifts. *Submitted*, 2017 (soumis au journal IFAC Journal of Process Control)

— J. Marino, F. Rossi, M. Ouladsine, and J. Pinaton. A real-time supervised and unsupervised fault detection algorithm for semiconductor processes. *Submitted*, 2017 (soumis à la conférence IEEE European Control Conference (ECC))

Remerciements

Je remercie Didier Theilliol et Zineb Simeu, pour avoir accepté d'évaluer mon travail en tant que rapporteurs pour cette thèse.

Je tiens également à remercier Mitra Fouladirad et Belkacem Ould Bouamama pour leur participation au jury de thèse.

Je remercie Jacques Pinaton et Mustapha Ouladsine sans qui ces travaux n'auraient pas pu être réalisés.

Je remercie Francesco Rossi pour le temps qu'il a consacré à ma thèse.

Introduction

La discipline de la détection de fautes en temps réel dans l'industrie du semi-conducteur a connu un essor important depuis le début du XXI^{e} siècle. Celle-ci vise à repérer les équipements dont la qualité de la production est altérée.

Dans cette industrie, la fabrication du produit final, qui se nomme "wafer", se fait à partir de l'enchaînement de plusieurs centaines d'étapes élémentaires. Ainsi, entre la première étape et la dernière, seul moment où le produit peut véritablement être testé, il s'écoule en général plus d'un mois. Si un équipement est défectueux, le premier produit non-fonctionnel n'est dans certains cas testé qu'au bout de plusieurs semaines. L'absence d'interruption de l'équipement pour réparation peut alors présenter un coût très important. Cette industrie étant extrêmement compétitive, l'augmentation des coûts liés à la présence occasionnelle de fautes sur les équipements peut avoir des répercussions sur le bilan d'une entreprise.

Le développement de la collecte de données, et la possibilité d'un traitement informatisé immédiat de celles-ci, permet aujourd'hui d'appliquer des algorithmes d'analyse statistique dans le but de détecter des fautes sur les équipements dès leurs premières manifestations. Cette potentialité technique n'apporte cependant pas de réponse au problème méthodologique de la conception de tels algorithmes.

Les données de l'industrie du semi-conducteur présentent certaines difficultés spécifiques. La production se fait suivant un très grand nombre de procédés de fabrication différents, nommés "recettes". Il y en a en général plusieurs milliers dans une même usine. Pour effectuer une modélisation procédé par procédé, il faut ainsi avoir un algorithme de détection de faute le plus automatisé possible.

Le fait que les équipements soient régulièrement entretenus à travers des maintenances, complexifie la modélisation statistique. En effet, chaque maintenance est susceptible de modifier la distribution statistique attendue dans les données récoltées sur les équipements. Il n'est alors pas possible de conserver un modèle statistique fixe dans le temps, car il est remis en

cause à chaque maintenance. De plus, la maintenance pose le problème de l'erreur humaine : toute intervention présente un risque, et donc l'absence de faute au redémarrage de l'équipement n'est pas garantie. Il est ainsi nécessaire de tester spécifiquement la présence de fautes après toute maintenance, avant de proposer une mise à jour de la modélisation statistique des données.

Les travaux effectués dans cette thèse sont basés sur notre expérience industrielle à STMicroelectronics Rousset. La thèse est découpée en 4 chapitres :

— Le Chapitre 1 inscrit la problématique de la détection de fautes dans le cadre industriel. Nous présentons l'industrie du semi-conducteur, avant d'aborder les moyens déployés pour la détection de faute à l'heure actuelle.

— Le Chapitre 2 traite des méthodologies existantes visant à réaliser de la détection de fautes en temps réel sur les données industrielles. Nous montrons la nécessité d'effectuer un alignement temporel des données avant toute étude statistique, puis nous mettons en valeur les manques des méthodologies disponibles dans la littérature concernant notre application spécifique.

— Le Chapitre 3 présente la méthodologie Gaussian Time Error (GTE) que nous proposons pour la détection de fautes. Il s'agit d'un test statistique, appliqué à chaque équipement individuellement, fait à chaque réalisation d'un procédé par celui-ci, dont les paramètres sont ajustés de façon supervisée. Les maintenances sont gérées de façon spécifique et automatisée : un test de validation est fait à chaque maintenance, avant mise à jour des paramètres.

— Le Chapitre 4 traite de la comparaison entre équipements effectuant les mêmes procédés. Nous proposons cette méthodologie pour compléter l'utilisation du GTE, en ajoutant un test sur les paramètres mis à jour à chaque maintenance, construit de façon non-supervisée. De plus, cette méthodologie permet d'assurer la cohérence des procédés réalisés dans les équipements d'un même atelier.

Chapitre 1

Problématique industrielle

Dans ce chapitre, nous introduisons les problématiques de la surveillance des procédés de fabrication de l'industrie du semi-conducteur. Pour cela nous faisons d'abord une présentation globale de cette industrie : ses acteurs, son intérêt, son organisation, et enfin une brève description des procédés auxquels les méthodologies présentées dans ces travaux s'appliquent. Nous traitons ensuite des méthodes de surveillance des équipements utilisées à l'heure actuelle à STMicroelectronics Rousset.

1.1 Présentation générale

L'industrie du semi-conducteur s'est illustrée par une croissance extrêmement importante au cours des dernières années [60]. En effet, les puces électroniques fabriquées par cette industrie, longtemps réservées à des applications spécifiques, ont pu être intégrées à une part grandissante de produits manufacturés. Cela est dû au fait de leur miniaturisation progressive et de la baisse des coûts de production qui l'a accompagnée. Ainsi, aujourd'hui, tout produit du quotidien est susceptible d'être amélioré par la présence de capteurs miniatures lui permettant de s'adapter à son environnement d'utilisation [56]. On parle alors d'objets "intelligents".

Pour la plupart des industriels du semi-conducteur, comme STMicroelectronics, la stratégie de production adoptée est dite "High-Mix Low-Volume". Dans ce cas, un produit donné n'est fabriqué que sur une courte période de temps, et en de faibles quantités. Il s'agit alors d'apporter de façon perpétuelle des modifications progressives à une technologie initiale : les coûts de recherche et de développement associés sont élevés, ajoutant également des difficultés d'organisation des usines de production où de nombreux produits différents doivent cohabiter. Les procédés de production se complexi-

fient avec le temps, mais la réduction des coûts s'opère par l'augmentation en valeur absolue de la taille des plaques de silicium sur lesquelles sont usinées les puces électroniques, en plus de l'augmentation relative conséquente à la miniaturisation. De plus, les nombreux procédés de fabrication exécutés relèvent tous d'étapes élémentaires communes. La stratégie des lignes ré-entrantes [48], qui veut qu'une même tâche sur un équipement donné puisse être réalisée plusieurs fois pour aboutir à un résultat donné, fait que tout type de produits peuvent circuler et être usinés les uns après les autres dans de mêmes équipements.

Une deuxième possibilité de réduction des coûts est de travailler à l'efficacité des procédés : compte tenu de la grande précision nécessaire à la fabrication, un pourcentage non-nul de produits sont non-fonctionnels en sortie d'usine. Le contrôle de la production se fait en cherchant à détecter les défaillances des équipements, afin de réduire le nombre de produits affectés dans de tels cas, et de corriger au plus vite les problèmes. L'objectif est de maintenir les équipements fonctionnels et d'empêcher les conséquences néfastes survenant suite à un dysfonctionnement non-anticipé. Notre thèse se focalise sur cet aspect de l'optimisation de la production.

1.1.1 Fabrication d'un dispositif à semi-conducteurs

La base initiale servant à la production dans les usines de l'industrie du semi-conducteur est appelée "wafer". Il s'agit d'une plaque de silicium d'une épaisseur d'environ 500 μm. Sa fabrication se fait à partir du sable, avec l'extraction de barreaux de silicium de plusieurs centimètres de diamètre, ensuite sciés en tranches. Il existe différentes tailles de wafers. À STMicroelectronics Rousset, les wafers usinés ont un diamètre de 8 pouces, soit environ 20 centimètres.

Sur le wafer, des couches de matériaux sont déposées successivement pour former des circuits intégrés, en suivant le plan établi par des "masques". Un masque est un support en quartz présentant les schémas électroniques pour une couche de matériau. Ils sont réalisés à partir d'une plaque de quartz sur laquelle est taillée une couche de chrome avec un laser. Un produit complexe peut nécessiter jusqu'à 30 couches de matériaux. La fabrication des masques est une étape préalable à l'usinage des wafers, et ne fait pas partie de notre champ d'étude.

On parle de partie "Front-end" de la production pour désigner le travail d'accumulation des couches sur les wafers. Sur un même wafer, de nombreux circuits intégrés sont créés côte-à-côte, et on parle de partie "Back-end" de la production pour évoquer la phase de découpe individuelle de

ces circuits intégrés, ainsi que la création des puces électroniques finales qui sont mises en boîtier. Ces travaux ne portent que sur la surveillance de la partie Front-end de la production. Cette phase se fait en de nombreuses étapes élémentaires, nommées "recettes", dont l'enchaînement spécifique est décrit par une "route" de production. Une route est constituée de jusqu'à 600 enchaînements de recettes (non nécessairement distinctes) pour certains produits.

La partie Front-end a lieu dans des usines très contrôlées : les circuits intégrés dessinés sur les wafers nécessitent une précision à l'échelle nanométrique. Ainsi, la moindre particule se déposant sur un wafer peut interférer avec le bon fonctionnement des puces électroniques. L'environnement contrôlé dans lequel les équipements de fabrication de l'industrie du semiconducteur sont disposés est nommé "salle blanche", et de nombreuses précautions y sont prises : régulation de l'air, port de combinaisons par les employés, confinement étanche des plaques...

Les routes de fabrication sont spécifiques pour chaque produit, mais les recettes qui les constituent sont essentiellement les mêmes pour la plupart des procédés : l'ajustement se fait sur les paramètres variables de ces recettes ainsi que sur les ordres d'enchaînement. On distingue quelques grands types de recettes :

— Lithographie : les dessins des masques sont reproduits dans une résine photosensible déposée sur le wafer.
— Dépôt de couches : des matériaux sont déposés sur le wafer par réaction avec des gaz injectés dans l'équipement. Les principaux cas d'étude de cette thèse sont basés sur des recettes de ce type : 4 exemples (Sections 3.6.1, 3.6.2, 4.5.1 et 4.5.2) sont étudiés.
— Gravure : il s'agit de l'inverse du procédé de dépôt. Des matériaux sont retirés au wafer, afin de réaliser le motif du masque. Dans la Section 3.3.2, nous employons des données issues d'une recette de gravure pour illustrer le point méthodologique présenté.
— Retrait résine : la résine déposée lors de la lithographie est brûlée pour être retirée. Nous présentons deux exemples aux Sections 3.6.3 et 4.5.3.
— Chimie : ces procédés permettent le nettoyage des wafers dans des baies.
— Diffusion : procédé à haute température (environ 1000°C) permettant d'introduire des impuretés sur les wafers.
— Implantation ionique : il s'agit de charger électriquement par des faisceaux d'ions, comme le bore, le phosphore et l'arsenic, certaines zones des wafers. Une discussion sur les données issues de ces équipements est menée à la Section 3.3.3.

Les travaux de cette thèse ont été conçus dans le cadre d'une approche générale, avec pour objectif une application à tous types d'équipements. Cependant, des contraintes pratiques, liées à la fréquence d'apparition de défauts à l'intérieur des différents ateliers de production, ou encore à la priorité donnée par l'entreprise à certaines étapes, font que nous ne proposons pas d'exemples issus de chaque type d'ateliers.

1.1.2 Spécificités de l'industrie du semi-conducteur

Dans cette section, nous traitons de l'organisation de la production dans l'industrie du semi-conducteur. Celle-ci se fait suivant les notions de lot et de chambre de production.

Les wafers sont regroupés par lots de 25, et il s'agit de l'unité de production pour l'industriel : un équipement donné prend en entrée un ou plusieurs lots de wafers, indépendamment du fait que de façon interne le traitement se fasse avec une plaque, avec une paire de plaques ou même avec 6 lots. Toutes les plaques d'un lot donné passent par la même route de production, et donc par les mêmes recettes. Cette notion de lot nous concerne dans cette thèse seulement du point de vue de l'évaluation concrète de ce que représentent les données récoltées : le lot est l'unité de mesure pour l'industriel, et certains ajustements doivent être faits relativement à cette exigence.

La plupart des équipements sont constitués de chambres de production qui réalisent ou non des procédés similaires. On parle d'équipements de type "cluster". Cela permet l'enchaînement d'étapes sans latence liée au transfert des lots vers un autre équipement. Les mêmes procédés sont disponibles sur plusieurs équipements distincts, afin de permettre la continuité de la production malgré les temps d'arrêt d'équipements individuels, et afin de façon générale à accélérer la production. Dans l'ensemble de ces travaux, le terme d'équipement désigne une chambre de production. Les parties communes servent généralement au transfert des wafers d'une chambre à une autre. Dans notre cadre d'étude, l'interdépendance entre les chambres d'un même équipement est quasi-inexistante et donc deux chambres d'un même équipement peuvent en général être considérées comme des équipements distincts.

1.1.3 Métrologie et tests sur les wafers

Dans cette section, nous traitons des tests effectués sur les produits dans l'environnement industriel.

La première approche de contrôle des équipements, est d'évaluer la qualité des produits qui en sortent. Pour s'assurer de la qualité de la production, des mesures directes sont effectuées tout au long du cycle d'un wafer : la métrologie contrôle les wafers pendant la production, et des tests sont réalisés une fois l'usinage achevé.

Il y a deux types de mesures en ligne, c'est-à-dire effectuées entre deux déroulements de recettes. Le premier, la métrologie, mesure des caractéristiques des wafers. Il s'agit de mesurer par exemple l'épaisseur d'une couche de matériau. Sur les plaques de production, seules des mesures non destructrices peuvent être réalisées, alors que des mesures plus poussées peuvent être faites sur des plaques de test. De nombreux types de défauts ne sont pas visibles avec ces mesures. L'autre type de mesure effectué entre deux recettes est la mesure dite de "défectivité". Il s'agit d'évaluer les défauts causés par des particules sur les plaques. Pour certains types d'équipements, qui représentent une faible proportion de l'usine, certaines mesures en ligne sont intégrées. Pour tous les autres cas, la mesure en ligne est coûteuse de par le ralentissement de la production engendré, et reste donc réduite à des échantillons sélectionnés.

En fin de production, il y a deux types de mesures effectuées. Le "Parametric Test" (PT) correspond à des mesures électriques sur les wafers. Plusieurs centaines de variables sont récoltées, et de nombreux problèmes peuvent être observés [66]. Enfin le test "Electrical Wafer Sorting" (EWS) donne le verdict sur la qualité du wafer, en mesurant la réponse électrique de chaque puce individuellement. La mesure en sortie de production est faite de façon systématique.

1.2 Détection et isolation de fautes dans l'industrie du semi-conducteur

Cette section présente la problématique de la détection de faute dans l'industrie du semi-conducteur. Usiner un wafer nécessite une précision nanométrique, et la moindre particule non désirée peut être source de dysfonctionnement du produit final. Pour cela, il est nécessaire de maintenir les équipements dans un état optimal de fonctionnement. La discipline nommée "détection et isolation de fautes" (FDC, relativement au nom anglais "Fault Detection and Classification") fournit des outils, déjà employés dans l'industrie, permettant d'appréhender le contrôle de la production.

1.2.1 Variabilité des procédés

La complexité des procédés mis en jeu dans un équipement de fabrication de l'industrie du semi-conducteur fait qu'une modélisation purement déterministe de ceux-ci n'est pas possible. Le résultat des phénomènes physiques est soumis à une propension de réalisation, telle que définie par Popper ([31]) : dans le cas où les conditions de réalisation sont maintenues constantes, le résultat exact obtenu en sortie peut être amené à varier avec une certaine probabilité. Dans le cas d'un phénomène déterministe, les mêmes conditions initiales donnent toujours exactement le même résultat, ce qui n'est pas le cas ici. Il s'agit d'une première source de variabilité des procédés.

Une seconde source de variabilité est due au fait que les conditions initiales ne sont que très rarement maintenues constantes. En effet, par exemple, l'accumulation de cette première source de variabilité au cours des recettes déjà exécutées oblige un ajustement des réactions ayant lieu dans l'équipement. De nombreux phénomènes peuvent entrer en jeu, comme de subreptices variations des conditions de pression et de température, et les résultats finaux ne seront donc jamais exactement identiques.

La réunion de ces deux sources de variabilité est nommée "variations de causes aléatoires" [51]. L'hypothèse fondamentale est que ces variations ne sont pas une source de défaillance des procédés. On parle ainsi d'un procédé sous contrôle quand celui-ci n'est soumis qu'aux variations de causes aléatoires.

Les variations qui sont sources de défaillances sont nommées "variations de causes assignables". Il s'agit alors d'un dysfonctionnement du système, qui ne réalise plus la tâche attendue. Ces variations peuvent également se répartir en deux classes : l'échec de l'équipement, et l'erreur humaine. L'équipement est le lieu dans lequel s'effectuent les processus physiques et chimiques permettant l'usinage des produits. Les différentes parties qui le constituent sont soumises à de fortes contraintes, et ainsi les résidus des procédés passés influencent la capacité de l'équipement à réaliser les procédés futurs, sous forme d'encrassement. De plus, certaines pièces ont une utilisation limitée dans le temps, et finissent donc par devenir défectueuses. La notion d'erreur humaine est nettement plus imprévisible : pièce mal serrée entraînant une fuite, pièce montée à l'envers, outil oublié, mauvais réglage, mauvaise communication entre deux personnes intervenant sur l'équipement...

L'objectif est de détecter la présence de causes assignables et d'en identifier la source afin de permettre le retour à la production normale. Les variations dues aux causes aléatoires doivent être évaluées, et leur présence

dans le système est tolérée, car aucune garantie ne peut être donnée quant à leur réduction.

1.2.2 Interventions de maintenance

Pour traiter les variations dues aux causes assignables, les équipements sont entretenus fréquemment. L'objectif est de maintenir l'environnement de production le plus stable possible [57].

Il y a deux types de maintenances :

— La maintenance préventive concerne les maintenances réalisées quand l'équipement est fonctionnel. Dans ce cas, les opérations usuelles sont le nettoyage de la chambre, pour éviter l'encrassement dû aux réactions chimiques passées, et le remplacement de pièces dont l'usure est connue. Ces opérations sont faites après une certaine durée d'utilisation depuis la dernière maintenance, après un certain nombre de produits traités, ou bien relativement à un indicateur d'usure pour certaines pièces.

— La maintenance corrective est réalisée lorsque l'équipement est défectueux. Dans ce cas, l'action est ciblée relativement au défaut, et l'objectif est de remettre l'équipement en état de fonctionnement correct.

La maintenance préventive pose le problème de la rentabilité : maintenir l'équipement, c'est interrompre la production pendant un certain laps de temps, et c'est également parfois payer des pièces de rechange. Exploiter l'équipement le plus longtemps possible tant qu'il fonctionne est plus rentable pour l'industriel, mais l'expose au risque de subir des défaillances plus fréquentes, avec un impact sur les produits. La maintenance préventive est également une source d'erreur humaine. En effet, un équipement fonctionnel sur lequel aucun technicien n'intervient est soumis à un risque d'erreur humaine nul. Dès qu'un technicien intervient, la possibilité d'erreur apparaît, et certaines erreurs humaines peuvent être tout aussi graves que des défaillances d'usures.

Toute méthodologie de contrôle des conditions de production d'un équipement doit tenir compte de la maintenance. En effet, les conditions de redémarrage suivant deux maintenances sont rarement identiques : certains paramètres sont réglés manuellement, certaines pièces ont une influence électrique (qui ne sera pas toujours la même après un remplacement)... Une première possibilité est d'éviter l'étude de paramètres dépendant de la maintenance. Une autre, celle que nous proposons dans le cadre de ces travaux, est de traiter la maintenance de façon spécifique dans le modélisation

du procédé.

1.2.3 Données récoltées

Dans cette section, nous présentons les caractéristiques des données récoltées lors de la production.

S'agissant de surveiller la stabilité des conditions de production, les données à récolter les plus pertinentes sont celles directement issues de l'équipement lui-même, pendant l'usinage. On parle de "run" pour qualifier l'exécution d'une recette dans un équipement donné. La notion de run recouvre ainsi l'ensemble du procédé ayant lieu dans un équipement particulier : les différentes variables physiques affectées par la réalisation du procédé sont les témoins de son bon déroulement. Les variables pouvant être récoltées sont très diverses : la pression de la chambre dans son entier ou dans une zone spécifique, le débit d'un gaz, une puissance électrique, une température... On suit alors l'évolution du run de son début jusqu'à sa fin par la récolte de ces variables choisies au préalable. La Figure 1.1 montre l'exemple d'une variable de pression pendant un run pour un équipement de dépôt.

Sur ce graphique, les points effectivement récoltés sont représentés, un segment étant tracé entre chacun d'eux. Il y a un point récolté à chaque seconde, soit 115 points sur cet exemple. De plus, les différentes étapes de la recette, nommées "steps", sont délimitées par des traits verticaux : il y a 13 steps dans cet exemple. Il s'agit d'une notion importante dans la récolte des données. En effet, la plage de valeurs de la pression est très large sur l'ensemble du procédé, mais chaque step correspondant à une réalité physique différente, les valeurs attendues dépendent grandement du step où la mesure est récoltée.

Les données récoltées pour un run sont ainsi une trajectoire temporelle multi-variée : il y a deux dimensions, d'une part les variables physiques et d'autre part le temps. Dans la suite de ces travaux, nous parlons de "données FDC" pour référer à ces trajectoires.

La FDC s'applique donc à exploiter ces données afin de détecter et diagnostiquer des fautes sur le procédé. L'impact sur la qualité des produits est ainsi indirect : l'objectif est de maintenir la plus faible variabilité possible d'un produit à l'autre, et celui-ci est rempli en s'assurant que la variabilité des conditions de traitement d'un produit à l'autre a été la plus faible

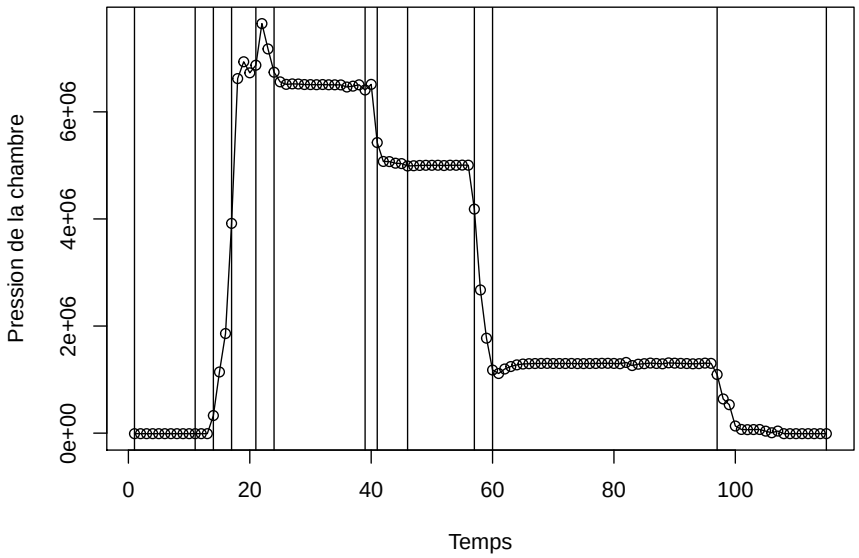

FIGURE 1.1 – Récolte de la pression de la chambre pendant un run de dépôt : pression en fonction du temps (en secondes).

possible pendant tout le cours de la production. Ces mesures sont disponibles pour chaque run : le run représente l'unité de base pour l'étude de la variabilité sur les équipements.

Nous ne traitons pas de la détection de fautes relative aux données de métrologie en ligne. Les limitations d'une telle approche, liées notamment au fait qu'il s'agisse de mesures hors ligne réalisées sur un faible nombre de produits et d'étapes de la route, sont décrites dans les travaux de Thieullen [70].

1.2.4 Notion d'indicateur statistique

La surveillance des données d'un run ne se fait en général pas sur les données brutes : l'information récoltée est transformée sous forme d'indicateurs.

Un indicateur est une grandeur statistique calculée sur un échantillon de données récoltées pendant un run. Par exemple, si on se base sur la Figure 1.1, on observe une valeur moyenne d'environ 0 pour la pression de la chambre au premier step de la recette étudiée. La moyenne pour chaque

run pour la pression sur ce premier step est un exemple d'indicateur, qui prend la valeur 0 dans cet exemple. L'usage d'indicateurs plus élaborés est parfois nécessaire. Ceux-ci font par exemple intervenir plusieurs variables physiques, la variabilité sur un step, ou encore le rapport entre les valeurs minimales ou maximales de différents steps.

La détermination des indicateurs statistiques à calculer pour un run donné se fait par connaissance experte dans le cadre de la détection de fautes telle qu'elle est pratiquée aujourd'hui dans l'industrie. Nous proposons une approche de calcul et de surveillance automatisée d'indicateurs statistiques au Chapitre 3.

1.2.5 Cartes de contrôle

L'approche aujourd'hui utilisée dans les usines comme STMicroelectronics Rousset est celle de l'emploi de cartes de contrôle mono-variées. Il s'agit à partir des données décrites Section 1.2.3 d'extraire une liste d'indicateurs statistiques qui sont ensuite observés de façon indépendante.

Choix d'indicateurs statistiques par connaissance experte

La démarche de sélection des indicateurs employés pour la méthodologie des cartes de contrôle est basée sur la connaissance experte. Dans le cadre de cette approche, le fait d'extraire des indicateurs a pour objectif de proposer une grille de lecture des données centrée autour de la connaissance des fautes possibles du système et de leurs manifestations. L'objectif pour chaque indicateur est de couvrir une manifestation possible d'un défaut connu. La sélection des indicateurs part d'analyses issues de la méthodologie "Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité" (AMDEC) [22]. Cette méthodologie organise une analyse fonctionnelle complète de l'équipement, condensant la connaissance experte pour relier les différentes causes possibles de défaillances aux manifestations qui en résultent. L'analyse de chaque équipement est faite composant par composant, fonctionnalité par fonctionnalité : il s'agit d'une méthodologie coûteuse à mettre en place, et qui peut avoir besoin de révisions importantes à chaque changement, même faible, des recettes de production. Après une telle analyse, les défaillances les plus critiques doivent avoir été identifiées avec une proposition d'indicateurs associés, permettant de constater leurs manifestations.

En plus du coût d'une telle démarche, sa limitation la plus importante est le fait de rejeter des données. En effet, il n'est pas possible d'avoir des indicateurs pour chaque step et chaque capteur, et chaque mode de fonction-

nement de chaque équipement. En effet, cela aboutirait à plusieurs millions de cartes de contrôle et deviendrait ingérable logistiquement pour maintenir à jour les cartes et traiter les alarmes associées. Cela n'est pas non plus souhaitable, car l'avantage de l'indicateur est bien de condenser un mode de défaillance connu de l'équipement : l'action corrective à prendre lorsqu'un comportement anormal de l'indicateur est détecté doit être connue à l'avance [11]. On observe par ailleurs un biais de réalisation pour les fautes déjà subies dans l'usine : la démarche qualité veut qu'un équipement ne redémarre pas après une défaillance initialement inconnue (détectée sur les produits finaux) sans la garantie de pouvoir la reconnaître en cas de récidive. Peu importe la rareté de la défaillance, si elle s'est déjà produite, alors une carte de contrôle y est associée. Des fautes potentiellement plus probables, n'ayant pas encore eu lieu, ne sont pas nécessairement couvertes par un indicateur : l'approche de mesure de la fréquence observée d'une faute donnée s'avère limitée face à la multiplicité des fautes possibles.

Suivi d'un indicateur par carte de contrôle

Une fois les indicateurs sélectionnés, leur comportement est évalué par l'utilisation de cartes de contrôle.

Une carte de contrôle sert à l'encadrement d'un indicateur donné. Elle est constituée de 4 éléments : une valeur cible, correspondant à la valeur attendue de l'indicateur, une limite supérieure de contrôle, qui correspond à la valeur maximale tolérée, une limite inférieure de contrôle, qui correspond à la valeur minimale tolérée, ainsi qu'une fonction de filtrage des valeurs, qui décrit la dépendance de la valeur testée relativement aux échantillons récoltés.

Les trois premiers éléments peuvent être fixés en se basant uniquement sur la connaissance métier : par exemple, l'équipement doit délivrer un débit de gaz précis à un step donné, et on connaît ce qui constitue une valeur trop faible ou trop élevée. On peut également fixer ces trois éléments par apprentissage statistique : un échantillon de données issus d'une sélection de runs est utilisé pour ajuster les paramètres d'une loi de probabilité, le plus souvent une loi normale. La valeur cible est par exemple la moyenne, et les limites sont prises parmi les quantiles de la loi [79]. Chaque carte de contrôle étant observée indépendamment, l'utilisation de quantiles soulève la problématique des fausses alarmes. Lorsque de nombreuses cartes surveillent le même run, la probabilité qu'au moins une carte soit hors limites est très grande relativement à la probabilité qu'une carte en particulier soit hors limites.

La fonction de filtrage est fixée relativement aux types de fautes atten-

dus. Les écarts d'amplitude instantanés (un seul run est très éloigné de la valeur cible) sont détectés par exemple avec une carte Shewhart, où aucun filtrage n'est appliqué : la valeur testée face aux limites est la dernière valeur récoltée. Les dérives peuvent par exemple être détectés avec une carte de type somme cumulée (CUSUM), où la valeur observée est l'accumulation des derniers indicateurs récoltés auxquels on soustrait la valeur cible. De nombreux types de filtres peuvent être appliqués, pour correspondre à la manifestation du défaut attendu [33].

1.2.6 Ajustement d'équipements différents exécutant un même procédé

Dans la Section 1.2.1, nous évoquons que le fait de maintenir les conditions de production les plus stables possible est un moyen de réduire la variabilité des produits finaux. Cependant, même au sein d'un même lot, les wafers passent à travers des équipements distincts. En effet, pour un grand nombre de recettes d'une route, le lot est scindé en plusieurs parties qui sont assignées à différentes chambres de fabrication. La même recette est réalisée, mais par des équipements différents : cela est une source de variabilité puisque les conditions de réalisation des recettes ne sont pas strictement identiques d'un équipement à l'autre. Les raisons sont multiples : cycles de maintenance distincts, vieillissement d'un équipement plus ancien que les autres, réglages manuels différents... Il en résulte le besoin de s'assurer que les écarts de condition de réalisation des recettes sont dus à des causes aléatoires, à savoir que les équipements sont ajustés du mieux possible et que la variabilité induite observée est simplement due au fait qu'ils sont distincts, et non pas à des causes assignables.

Ajustement en temps réel

Une première possibilité est de proposer des cartes de contrôle, décrites Section 1.2.5, communes aux différents équipements pour une même recette. Cependant, cela rajoute une contrainte supplémentaire dans le choix des indicateurs, puisque le suivi des valeurs dépendant de réglages spécifiques des équipements devient difficile. On peut par exemple devoir sélectionner des limites très larges pour contenir toutes les données de fonctionnement normal, ou même devoir éviter ce type d'indicateurs : dans les deux cas, il y a une perte d'information pouvant masquer une défaillance du système.

Ajustement hors-ligne

Une deuxième possibilité est de mener des analyses spécifiques hors-ligne, c'est-à-dire sur l'historique des données, afin d'évaluer les différences entre équipements réalisant les mêmes recettes. Ces analyses ne sont pas menées de façon systématique, et sont faites par exemple lorsque des disparités entre les produits sortant d'un même parc d'équipements sont observées.

De telles analyses sont menées via la recherche visuelle de chambres atypiques, basée sur l'utilisation de l'Analyse en Composantes Principales (ACP) appliquée aux indicateurs des cartes de contrôle communs aux différents équipements. Nous détaillons l'ACP dans la Section 2.3.3. Nous indiquons seulement ici qu'il s'agit d'une méthodologie permettant d'effectuer une projection d'un ensemble de données vers un nouvel espace dont les axes, nommés composantes, sont calculés afin de maximiser la variance observée sur chaque axe. Ainsi, on peut obtenir une représentation fidèle des données dans un espace de dimension plus faible : en prenant les deux premières composantes, on peut alors visualiser des données initialement de grande dimension. La Figure 1.2 montre un exemple de graphique obtenu pour une telle application de l'ACP. Chaque point correspond aux données d'un run, et chaque couleur correspond à un équipement, 10 au total. Dans cet exemple, on observe que chaque équipement correspond à une faible portion de la surface représentée, et 7 équipements sont regroupés sur la partie droite et haute du graphique. Un équipement est nettement plus bas, et deux équipements sont nettement plus à gauche. Pour identifier la cause de ces différences, on utilise les contributions aux composantes : les composantes sont corrélées aux variables initiales, et le décalage est supposé dû à des différences liées aux variables les plus corrélées avec ces deux premières composantes.

Cette procédure demande donc une investigation manuelle, centrée autour de choix subjectifs : un autre individu pourrait penser qu'une des trois chambres que l'on a désigné n'est pas atypique, ou bien qu'il y en a une quatrième à repérer. En exploitant les données des indicateurs au lieu des données FDC complètes, les défauts des cartes de contrôle s'additionnent à cette subjectivité. Il n'y a ainsi pas, au moment de la conception de ces travaux, de méthodologie systématique de comparaison entre équipements employant les données FDC complètes. Nous proposons une telle méthode dans le Chapitre 4. Il s'agit d'une des contributions principales de cette thèse.

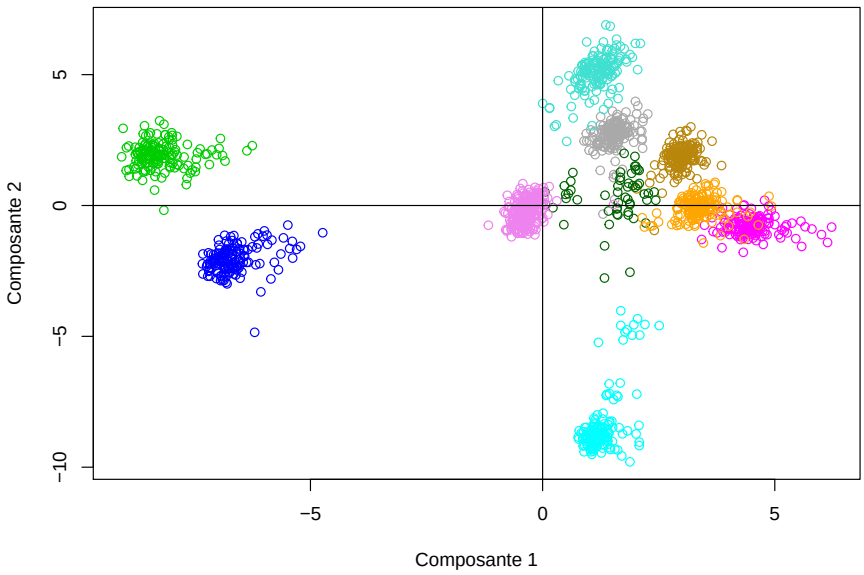

FIGURE 1.2 – Représentation en deux dimensions des données d'un parc d'équipements par ACP.

1.3 Conclusion

L'industrie du semi-conducteur se développe de façon très rapide, entraînant une concurrence importante, avec le besoin d'un rendement accru des procédés. Détecter les équipements défaillants le plus vite possible permet d'augmenter ce rendement. Privilégier la surveillance des équipements eux-mêmes, plutôt que des produits qu'ils traitent, est un moyen de s'affranchir de nombreuses contraintes, allant de l'imprécision des mesures faites sur des produits non finis au ralentissement de la production nécessaire à la réalisation des mesures. Une telle surveillance s'exerce grâce à la récolte de paramètres physiques, déterminés par des experts, spécifiques à chaque équipement et représentatifs du procédé réalisé par celui-ci. Dans la pratique industrielle, l'analyse de telles données est généralement réalisée par des cartes de contrôle : il s'agit d'une stratégie d'identification et de caractérisation préalable des fautes possibles de l'équipement.

Nous proposons, dans les Chapitres 3 et 4, une méthodologie de surveillance permettant de s'affranchir du travail de caractérisation préalable des fautes, afin de pouvoir détecter toute différence sans a priori.

Chapitre 2

Détection de fautes sur des trajectoires temporelles multi-variées : un état de l'art

Dans le Chapitre 1, nous soulevons les enjeux de la surveillance des procédés industriels du semi-conducteur par l'utilisation des données FDC complètes. Nous n'évoquons donc dans ces travaux que des méthodes pouvant s'appliquer à de telles données, et excluons les méthodologies basées sur des choix d'indicateurs, les données de métrologie en ligne ou la connaissance préalable des fautes possibles du système.

Les méthodes s'appliquant à l'intégralité des données FDC nécessitent une phase préliminaire de pré-traitement. En effet, nous n'avons pas connaissance de propositions dans la littérature permettant de se passer d'une transformation des données initiales. Ce besoin est soit implicite, car les hypothèses faites sur la nature des données ne sont pas vérifiées dans la pratique [19, 20, 55, 76], soit directement spécifié [39, 70, 85]. Nous présentons ainsi d'abord les spécificités des données engendrant le besoin d'un pré-traitement, avant d'aborder les méthodologies permettant ce pré-traitement. Enfin, nous aborderons les travaux sur le sujet de la détection de défauts appliquée aux données pré-traitées.

2.1 Caractéristiques des trajectoires temporelles multi-variées récoltées

Les données FDC correspondent à une récolte de données échantillonnée pour une sélection de variables physiques. L'objectif est de suivre le

procédé pendant tout son déroulement.

Cependant, l'inscription dans le temps des différentes étapes est variable : certains steps sont conditionnels, au sens où le step se poursuit tant qu'un certain processus physique n'est pas achevé, et l'influence de l'exécution des recettes précédentes peut amener certains steps à se dérouler plus ou moins rapidement.

De plus, des contraintes techniques viennent s'ajouter à ces deux sources de variabilité du temps du procédé : l'organisation de la collecte des données est soumise à la disponibilité du réseau. En effet, le même réseau fait circuler une grande quantité d'informations, comme par exemple les recettes à appliquer ou encore la disponibilité des équipements, et les données de surveillance récoltées ne sont pas prioritaires sur ces informations servant directement à la production. On observe de manière fréquente l'absence d'une ou deux mesures dans un run, et également des décalages, faibles mais non négligeables, dans la réalisation de l'échantillonnage souhaité. Par exemple, si une mesure doit être récoltée à la seconde 1 et à la seconde 2, on pourra récolter à la place une mesure à la seconde 0,9 et une autre à la seconde 2,1.

Enfin, dans le contexte industriel, les recettes similaires sont réunies sous la notion de stratégie de collecte, ou stratégie FDC. Par similaire, on désigne des recettes dont seul le temps d'exécution des steps change. Par exemple dans le cas d'un dépôt de matériau, le step où effectivement a lieu le dépôt peut être rallongé pour un dépôt plus épais. Dans ce genre de cas, aucune autre différence n'est à prévoir, et l'exécution de telles recettes est mélangée : l'ajustement du temps de procédé est souvent fait à travers des boucles de régulation [15]. Les informations récoltées sur les runs précédents amènent à faire varier volontairement le temps d'un run donné. Le résultat de cette régulation est que deux runs consécutifs peuvent correspondre à des temps d'exécution significativement différents.

Il est donc nécessaire pour l'analyse des données d'employer une méthodologie qui soit robuste à ces décalages temporels, soit car l'égalité des temps de mesure n'est pas un pré-requis, soit car les différences de temps ont été gérées en amont par une procédure de pré-traitement des données. La plupart des méthodologies basées sur des calculs matriciels font implicitement l'hypothèse d'une dimension fixe des données d'un run à l'autre. Il est alors nécessaire de faire passer les données initiales de chaque run par une première étape où les dimensions individuelles de ces données de runs sont réajustées vers une dimension commune.

2.2 Alignement temporel des runs

L'alignement temporel est l'étape principale de pré-traitement pour les données du semi-conducteur [70]. Il s'agit d'appliquer une transformation des données initiales afin que tout run soit représenté par une matrice de dimensions fixes. Le nombre de capteurs est supposé invariable, car les changements du plan de collecte sont rares. En effet, les capteurs sont en général onéreux, et la bande passante est limitée : le plan de collecte est soigneusement déterminé, et reste identique pour une longue période de temps. Ce pré-traitement concerne ainsi l'ajustement des données du point de vue de la dimension temporelle uniquement.

On distingue 4 méthodologies principales pour l'alignement temporel de deux trajectoires multi-variées [39] :

— La combinaison de la troncature des trajectoires trop longues et de la complétion des trajectoires trop courtes ;
— La mise à l'échelle linéaire du temps, qui s'appuie sur des variables indicatrices du déroulement du procédé [81] ;
— La déformation optimisée suivant les corrélations [49] (généralement désignée par le sigle COW issu de son nom anglais "Correlation Optimized Warping"), qui résout par programmation dynamique une somme de problèmes d'optimisation locaux visant à maximiser la corrélation entre les deux trajectoires alignées ;
— La déformation temporelle dynamique [32] (généralement désignée par le sigle DTW issu de son nom anglais "Dynamic Time Warping"), qui résout par programmation dynamique un problème d'optimisation global visant à minimiser une distance entre les deux trajectoires alignées.

Les deux premières méthodes ne peuvent pas être pertinentes dans notre cas. En effet, la troncature et la complétion se font en fin et en début de trajectoires, et comme décrit dans la Section 2.1, une grande partie des désynchronisations ont lieu au milieu des signaux récoltés, par exemple lorsqu'un step a une durée variable. La mise à l'échelle linéaire du temps s'applique lorsqu'une variable a une progression linéaire dans le temps et porte l'information de l'avancée du procédé. C'est un cas trop rare parmi les équipements étudiés pour que cette méthode puisse être pertinente en général.

2.2.1 Déformation optimisée suivant les corrélations

Nous décrivons le fonctionnement et les limitations, de la méthode COW. Cet algorithme a été initialement conçu pour l'alignement des don-

nées de chromatographie, discipline qui étudie la séparation physico-chimique d'un mélange de substances [49].

On fixe une trajectoire de référence $Y = (y_1, ..., y_m)$ sur laquelle on aligne une autre trajectoire $X = (x_1, ..., x_n)$ où $m \geq n$ sont deux entiers désignant le nombre de points récoltés pour Y et X respectivement. Le principe de cet algorithme est de déterminer des associations entre sections des signaux. Le contenu des sections de X est ajusté par l'ajout ou le retrait de points adjacents de telle façon à maximiser la corrélation avec les sections correspondantes de Y. Les étapes de l'algorithme sont les suivantes :

— Tout d'abord, les deux signaux sont scindés en N sous-sections, où N est un entier choisi par l'utilisateur. On fixe des entiers $k_0^x, ..., k_N^x$ et $k_0^y, ..., k_N^y$ avec la contrainte $k_0^x = k_0^y = 0$ et $k_N^x = n$, $k_N^y = m$. On pose ainsi $Y = (Y_1, ..., Y_N)$ et $X = (X_1, ..., X_N)$, avec pour un entier i donné $X_i = (x_{k_{i-1}^x+1}, ..., x_{k_i^x})$ et $Y_i = (y_{k_{i-1}^y+1}, ..., y_{k_i^y})$.

— La première section X_1 de X est comparée à la première section Y_1 de Y. On détermine $\tilde{X}_1$, qui est la transformation de X_1 permettant de maximiser la corrélation avec Y_1. L'ensemble $\tilde{X}_1$ correspond à $\tilde{X}_1 = (x_1, ..., x_{k_1^x+\tilde{r}})$ où $\tilde{r} = \arg\max_{r\in\{-s,...,s\}} Cor(Y_1, (x_1, ..., x_{k_1^x+r}))$. Le paramètre s, fixé par l'utilisateur, fait qu'il n'y a qu'un nombre fini de possibilités. Toutes les corrélations correspondantes sont calculées. Le premier et le dernier point de X sont fixes et ne peuvent pas être décalés. Dans les autres cas, les points de début et de fin de chaque section peuvent être décalés d'une valeur allant jusqu'au paramètre s, vers l'avant ou vers l'arrière. Le sous-ensemble $\tilde{X}_1$ est étiré ou raccourci par interpolation linéaire afin d'avoir autant de points que Y_1, pour permettre le calcul de corrélations.

— La même procédure est répétée pour X_2, avec la différence que le premier terme n'est pas fixe dans ce cas. On détermine donc $(\tilde{r}_1, \tilde{r}_2) = \arg\max_{r_1,r_2\in\{-s,...,s\}} Cor(Y_1, (x_{k_1^x+1+r}, ..., x_{k_2^x+r}))$ et ainsi de suite jusqu'à la détermination de $\tilde{X}_N$. La maximisation de chaque corrélation individuelle fait que l'ensemble $\{\tilde{X}_1, ..., \tilde{X}_N\}$ constitue une solution d'alignement globale par programmation dynamique (résolution optimale de sous-problèmes pour atteindre la solution globale).

L'avantage principal de cette méthodologie est que le signal aligné final obtenu ne présente qu'une très faible distorsion. Cette méthodologie présente cependant un inconvénient majeur dans notre cas d'étude, qui la rend inutilisable. En effet, le choix du paramètre s pèse lourdement sur la complexité informatique, et garder s faible garantit une distorsion minimale. Cependant, dans l'industrie du semi-conducteur, le temps d'exécution de

certaines recettes sous une même stratégie de collecte peut doubler. Dans ce cas, le paramètre s doit représenter la longueur totale du vecteur X : la complexité informatique devient alors extrêmement importante, et une application en temps réel n'est plus possible. De plus, la distorsion de la trajectoire initiale peut devenir non négligeable vu la liberté de variation permise. Cette méthodologie n'est pas conçue pour comparer des trajectoires de tailles très différentes [75].

2.2.2 Déformation temporelle dynamique

L'algorithme DTW est issu de recherches sur l'alignement de données du domaine de reconnaissances de la parole [32]. Le problème résolu est le même que pour l'algorithme COW. Cependant, l'approche proposée a une moindre complexité informatique, et s'adapte à des différences de longueurs arbitraires entre le signal à aligner et la référence.

On considère X_r une trajectoire de référence. L'objectif est d'aligner les points de X_c, trajectoire cible, relativement à cette référence. On dispose de m variables, et n_r (respectivement n_c) observations.

L'algorithme vise à trouver $S = \{c(1), ..., c(K)\}$ où $c(k) = (i(k), j(k))$ est le k^e point de la trajectoire alignée, mettant l'observation numéro $i(k)$ de X_r en correspondance avec l'observation numéro $j(k)$ de X_c. Le nombre d'observations retenues K est, dans le cadre général, déterminé par l'algorithme. Dans le cadre du pré-traitement d'un jeu de données complet, il est nécessaire d'utiliser une contrainte dite "asymétrique", dans laquelle $K = n_r$ est fixé au préalable : ainsi, après alignement toutes les trajectoires auront exactement le même nombre d'observations. Sans cette contrainte, l'alignement perd son sens puisque rien ne garantit la taille finale des trajectoires alignées.

Pour déterminer S, on détermine un coût pour chaque trajectoire, définie par les associations entre les points de X_r et les points de X_c, calculé comme la somme cumulée de coûts locaux pour chaque association. Soit d une distance sur $\mathbb{R}^m$. On définit la matrice des coûts locaux comme étant : $D = (d(X_r(i), X_c(j)))_{i,j \leq n_r, n_c}$. La minimisation se fait suivant 3 contraintes :

- La contrainte de frontière, à savoir que les deux échantillons aux extrémités sont mutuellement associés : $c(1) = (1, 1)$ et $c(n_r) = (n_r, n_c)$;
- La contrainte de monotonie, qui oblige l'impossibilité du retour en arrière : les fonctions $k \mapsto i(k)$ et $k \mapsto j(k)$ sont croissantes ;
- La contrainte de continuité, qui permet d'éviter la suppression d'échantillons en limitant l'écart maximal entre deux observations : pour

tout $k \leq n_r - 1$, $c(k+1) - c(k) \in \{(1,0),(0,1),(1,1)\}$.

La trajectoire S est alors obtenue en minimisant $T_S = \frac{\sum_{k=1}^{n_r} d(i(k),j(k))w(k)}{\sum_{k=1}^{n_r} w(k)}$. Ceci signifie que l'on minimise la somme des distances individuelles entre chaque assignation. S'agissant de la minimisation d'une distance cumulée, l'approche par programmation dynamique permet d'apporter la meilleure solution.

Le paramètre $w(k)$ est une fonction de pondération. Le choix le plus largement utilisé est le suivant : $w(k) = i(k) - i(k+1) + j(k) - j(k+1)$, avec le choix fixé $i(0) = j(0) = 0$. Il s'agit d'une pondération dite symétrique : il n'y a pas de prévalence de la trajectoire de référence sur la trajectoire cible.

En général la distance utilisée pour DTW est la distance euclidienne :

$$\begin{aligned} d(i(k), j(k)) &= d(X_r(i(k)), X_c(j(k))) \\ &= (X_r(i(k)) - X_c(j(k)))(X_r(i(k)) - X_c(j(k)))^T. \end{aligned}$$

2.2.3 Application à la dérivée des trajectoires

Un défaut connu du DTW tel que présenté, s'appuyant sur la distance euclidienne, est le fait que l'algorithme privilégie la recherche d'amplitudes égales pour les associations. Ainsi, à l'intérieur des trajectoires, des pics d'amplitudes légèrement différentes mais correspondant à la même phase du procédé peuvent ne pas être associés. La méthode dite DDTW (Derivative Dynamic Time Warping) corrige ce problème en alignant les trajectoires en fonction des dérivées du signal plutôt que des amplitudes : ceci permet de prendre en compte ses variations.

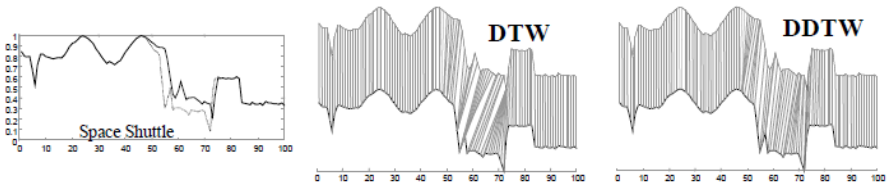

FIGURE 2.1 – Comparaison entre DTW et DDTW.

Dans l'exemple Figure 2.1, extrait de l'article de Keogh et Al. introduisant DDTW [34], on aligne deux courbes l'une par rapport à l'autre, en utilisant d'une part DTW et d'autre part DDTW. Les courbes sont décalées en amplitude afin de pouvoir lisiblement désigner les associations de points par un trait reliant les deux courbes. On remarque notamment que, dans le

cas DTW, il y a des changements brutaux d'associations dans la partie centrale du signal. En effet, le pic visible dans la courbe du bas est d'amplitude inférieure, et c'est donc le point de même amplitude le plus proche qui est choisi dans la courbe du haut. Cela n'est pas le cas pour DDTW.

Pour obtenir ce résultat, on utilise la distance euclidienne présentée précédemment, non pas entre les points, mais entre les dérivées estimées en ces points. On peut utiliser par exemple une estimation classique de la dérivée comme celle ci-dessous, en désignant le temps au point i comme étant t_i :

$$d_X(i) = \frac{X(i+1) - X(i)}{t_{i+1} - t_i}. \tag{2.1}$$

Les différences d'amplitude à l'intérieur de mêmes étapes du procédé sont fréquentes dans les données de l'industrie du semi-conducteur : l'algorithme DTW appliqué à ces signaux entraîne des déformations non souhaitées. L'application de DDTW à de telles données industrielles s'avère efficace dans la pratique [70]. Dans ces travaux, nous appliquons l'algorithme DDTW dans le cadre du pré-traitement des données (voir Section 3.3).

2.2.4 Problématique de la trajectoire de référence

Dans la Section 2.2.2, nous nous plaçons dans le cas où une trajectoire de référence est disponible, et les trajectoires cibles, correspondant aux données récoltées, sont toutes alignées relativement à cette référence pour permettre un alignement commun. De la même façon qu'il n'est pas possible de prédire l'ensemble des paramètres du procédé par la seule connaissance des paramètres d'ajustement de l'équipement, il n'est pas possible de construire par expertise métier de trajectoire de référence fiable couvrant l'ensemble du déroulement d'une recette. Les méthodologies existantes partent donc d'un ensemble d'apprentissage, et la trajectoire de référence en est soit issue directement soit calculée à partir de celui-ci.

La première possibilité est de choisir la trajectoire de référence parmi les trajectoires disponibles dans les données. Dans le cadre de l'alignement de signaux de reconnaissance vocale [32], un choix complètement aléatoire de la trajectoire de référence est proposé. Cette solution s'avère très peu recommandable pour notre étude vu le risque de choisir une trajectoire affectée par des problèmes de collecte. Nous n'avons pas connaissance de l'application d'un choix aléatoire de la référence pour le pré-traitement de données de l'industrie du semi-conducteur.

Thieullen [70] propose une approche de sélection d'un sous-ensemble

des runs, sur lequel est appliqué un algorithme permettant le calcul d'une moyenne de l'ensemble sous la pseudo-distance DTW, dû à Petitjean [54], nommé "Dynamic Barycenter Averaging" (DBA). Partant d'un ensemble de runs dont les données ont été collectées, une mesure de similarité nommée "Extended Frobenius Norm", introduite dans [82] est calculée pour chaque paire de runs. Cette mesure est basée sur les produits scalaires des vecteurs singuliers à droite issus de la décomposition en valeurs singulières de chacun des deux runs. Les dimensions sont compatibles puisque le nombre de capteurs récoltés sont les mêmes malgré les différences de temps de déroulement entre runs. À partir de cette étape, les 10% de runs dont la mesure de similarité moyenne avec les autres est la plus faible sont rejetés. Sur les 90% restants, l'algorithme DBA est appliqué.

Les différentes étapes de l'algorithme DBA sont les suivantes :

1. Choix d'une référence initiale, temporaire, aléatoirement parmi les trajectoires disponibles ;

2. Alignement par DTW de toutes les trajectoires restantes sur la référence temporaire ;

3. La référence temporaire est mise à jour en remplaçant chaque point par la moyenne des points qui lui ont été assigné par DTW dans les autres trajectoires ;

4. Répétition des étapes 2 et 3 jusqu'à l'atteinte d'un critère de convergence : la trajectoire de référence temporaire à cette étape là est alors la référence finale.

Parmi les critères de convergence possibles, le plus simple est de choisir un nombre fixe d'itérations, tel que proposé dans [70], où est observée une décroissance rapide de l'impact de chaque itération. D'autres critères, comme par exemple la variation de la pseudo-distance DTW entre la trajectoire temporaire actuelle et la précédente, peuvent être utilisés. Les défauts principaux de l'algorithme DBA sont la grande corrélation entre le résultat final et la trajectoire de référence initiale choisie, l'important coût en termes de temps de calcul, ainsi que l'obtention de trajectoires d'apparence bruitée aux instants auxquels les amplitudes des différentes trajectoires de l'ensemble d'apprentissage ne coïncident pas. Nous discutons dans la Section 3.3.1 de la détermination d'une trajectoire de référence pour les données étudiées dans ces travaux.

2.3 Détection et localisation de fautes appliquée aux données pré-traitées

Après la phase d'alignement, les données des runs pour un processus donné se présentent sous forme de matrices de dimensions $K \times J$, où K est le nombre d'instants de mesure correspondant à la référence d'alignement et J le nombre de capteurs de l'équipement. Il est supposé que les différentes étapes du processus sont superposées : la k^{e} ligne de chacune des matrices représente la même information pour les différents runs. Nous allons tout d'abord délimiter le type de problèmes dont relève ce travail, avant de présenter les méthodologies servant usuellement à y apporter une solution.

2.3.1 Classification par apprentissage statistique supervisé mono-classe

L'apprentissage statistique est une discipline traitant de la possibilité de faire remplir une tâche complexe à une machine par l'exploitation de données statistiques. Dans notre étude, la tâche à réaliser est celle de savoir reconnaître quand l'équipement fonctionne correctement, et quand ce n'est pas le cas, à partir des données collectées pendant les runs. Cette tâche relève de la notion de classification. Un algorithme de classification permet d'assigner des données d'entrée à des groupes connus à l'avance. Ici, il n'y a que deux groupes, qui sont "bon fonctionnement" et "mauvais fonctionnement". D'autres problèmes typiques de la classification sont le partitionnement de données (souvent désigné par son nom anglais "clustering"), qui réalise la même tâche que la classification mais en découvrant dans les données les groupes dans lesquels celles-ci doivent être classées, ou encore la régression, où il s'agit d'expliquer la valeur d'une ou plusieurs variables en fonction des données d'entrées, pour réaliser par exemple des prédictions.

Un modèle d'apprentissage statistique part d'un ensemble dit d'apprentissage sur lequel une modélisation est conçue. Des données similaires externes à cet ensemble peuvent alors être analysées à travers la modélisation qui a été faite.

Il existe trois grands types de classification :

— Classification supervisée : dans ce cas, les données d'apprentissage sont labellisées. Ainsi, la classe de chaque exemple d'apprentissage est connue ;

— Classification semi-supervisée : dans ce cas, certaines données d'apprentissage sont labellisées. Les exemples non-labellisés peuvent tout de même apporter des informations exploitables ;
— Classification non-supervisée : dans ce cas, aucun exemple n'est labellisé. Les classes doivent alors être caractérisées à l'intérieur des données.

Dans le cadre de la détection de fautes appliquée aux données FDC, nous avons vu qu'il n'était pas possible de prévoir les multiples fautes pouvant apparaître dans le système : il faut donc se contenter d'exemples de bon fonctionnement uniquement. On parle d'apprentissage supervisé mono-classe. Parmi les deux classes possibles, les données d'apprentissage ne caractérisent que le bon fonctionnement. Il s'agit alors de délimiter leurs caractéristiques communes, et ainsi définir négativement les données de la classe "mauvais fonctionnement", comme étant les données ne correspondant pas aux caractéristiques communes des données d'apprentissage.

2.3.2 Machines à vecteurs de support mono-classes

Les algorithmes de machines à vecteurs de support (SVM pour "support vector machine") représentent un grand nombre de cas d'utilisation en classification [3, 9, 35, 46, 52]. L'objectif de ces algorithmes est de créer un séparateur des données, c'est-à-dire de délimiter l'espace des données entre différents sous-espaces qui caractérisent les classes des individus.

Les SVM s'appuient sur le théorème de Cover [13] : la probabilité que deux classes représentées dans un jeu de données fixé soient linéairement séparables croît avec la dimension dans laquelle sont représentées les données. Pour une dimension n, la notion de séparabilité linéaire désigne le fait de pouvoir trouver un hyperplan, c'est-à-dire une surface de dimension $n-1$, tel que d'un coté de l'hyperplan se trouve la première classe, et de l'autre se trouve la seconde classe. Ce théorème se comprend intuitivement du fait que pour n valeurs prises dans $\mathbb{R}$, il suffit de projeter les n points sur un $n-1$-simplexe (un simplexe est la généralisation aux dimensions supérieures du triangle qui est un 2-simplexe) pour s'apercevoir que toute partition de ces n points est linéairement séparable dans cet espace de dimension $n-1$.

Algorithme SVM avec deux classes

L'algorithme des SVM est basé sur deux actions : la projection dans un espace de plus grande dimension, et la recherche dans ce nouvel espace d'un hyperplan séparant les données. L'hyperplan est calculé en maximisant

la marge entre les deux classes à séparer, et il s'agit d'une courbe non-linéaire lorsqu'il est projeté dans l'espace initial.

Nous présentons d'abord le principe général de cet algorithme lorsque les données sont labellisées avec deux classes, que l'on note -1 et 1, avant de traiter des deux propositions d'apprentissage dans le cas où une seule classe est disponible.

Soit $\{(x_1, y_1), \dots (x_n, y_n)\} \in \mathbb{R}^N \times \{-1, 1\}$ l'ensemble initial des données labellisées. Si on considère les données projetées dans un espace F de dimension supérieure à N, alors l'équation d'un hyperplan dans cet espace pour une observation $x \in F$ est $w^T x + b = 0$ où $w \in F$ et $b \in \mathbb{R}$. La projection se fait avec une fonction $\phi : \mathbb{R}^N \mapsto F$. Pour éviter le sur-apprentissage, ce qui voudrait dire proposer un hyperplan correspondant trop spécifiquement aux données avec de faibles capacités de généralisation, des paramètres de relâchement ξ_i sont ajoutés, avec une constante de pondération C. Le problème à résoudre pour trouver l'hyperplan de séparation est alors :

$$\min_{w,b,\xi_i} \frac{||w||^2}{2} + C \sum_{i=1}^{n} \xi_i, \tag{2.2}$$

sous les contraintes $\xi_i \geq 0, y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$ pour tout $i \leq n$.

Lorsque ce problème est résolu avec la méthode des Lagrangiens, le classifieur f, c'est-à-dire la fonction qui à une entrée x associe une valeur dans $\{-1, 1\}$ est :

$$\begin{aligned} f(x) &= sgn\left(\sum_{i=1}^{n} \alpha_i y_i \phi(x)^T \phi(x_i) + b\right) \\ &= sgn\left(\sum_{i=1}^{n} \alpha_i y_i K(x, x_i) + b\right) \end{aligned}. \tag{2.3}$$

Dans cette équation, "sgn" désigne la fonction qui à un nombre réel associe son signe, les α_i sont les multiplicateurs de Lagrange et K désigne le produit scalaire de deux éléments de F.

On observe ainsi que le classifieur obtenu au final ne dépend pas de ϕ : il n'y a pas besoin d'effectuer réellement la projection, ni même de connaître l'espace F, il suffit de connaître le produit scalaire K, nommé "noyau". C'est ce que l'on nomme l'astuce noyau, et cela permet de faire indirectement des calculs en très grandes dimensions sans être limité par la complexité informatique pouvant en découler. La dimension de F peut même éventuellement être infinie.

Il existe une très grande variété de fonctions noyaux, et l'adaptation à un jeu de données particulier n'est pas garantie [5]. Un exemple de noyau

couramment utilisé est le noyau radial gaussien, avec $\sigma \in \mathbb{R}$ et $||.||$ une distance :

$$K(x, x') = \exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right). \tag{2.4}$$

Réduction à une seule classe d'apprentissage

Il y a deux approches classiques pour transformer cet algorithme nécessitant des exemples des deux classes en un algorithme mono-classe :

— La première, proposée par Schölkopf [65] est de séparer les données d'apprentissage de l'origine dans l'espace F, en maximisant la marge prise par l'hyperplan avec l'origine. L'idée est ainsi que l'origine de l'espace F est le représentant de la classe "mauvais fonctionnement". La projection dans l'espace initial de cet hyperplan donne un contour non-linéaire des données initiales.

— La seconde, proposée par Tax et Duin [69] est de construire une hyper-sphère dans l'espace F qui entoure les données d'apprentissage. L'idée est ici que, dans l'espace F, les données de mauvais fonctionnement entourent les données de bon fonctionnement, et ce dans toutes les directions. Cette méthodologie est nommée Support Vector Data Description (SVDD). Le résultat obtenu est également un contour non-linéaire des données initiales.

Dans les deux cas, les modifications à effectuer dans l'algorithme présenté Section 2.3.2 sont mineures, et on retrouve donc une expression du classifieur final très similaire, même si ces deux algorithmes font chacun apparaître des paramètres spécifiques.

L'application des algorithmes SVM mono-classes dans l'industrie du semiconducteur est soumise à deux grandes problématiques. D'une part, le choix d'un noyau permettant un contour correct des données est une étape difficile. En effet, potentiellement tout ensemble d'apprentissage requiert un noyau différent pour une performance optimale, et la diversité des équipements rend la paramétrisation complexe [4]. La deuxième contrainte est celle que l'utilisation des SVM pour localiser les fautes est limitée par l'aspect boîte noire de l'algorithme [37]. La projection dans l'espace initial de l'hyperplan calculé dans l'espace F n'est qu'une potentialité, puisque l'espace F n'est pas connu en général. Il n'est pas toujours possible de déterminer a posteriori les raisons pour lesquelles une observation a été classée comme étant extérieure à la classe d'apprentissage, ce qui rend plus difficile la prise de décision a posteriori : savoir réagir à une détection de mauvais fonctionnement est très difficile sans indice quant à la nature de la faute.

2.3.3 Algorithmes dérivés de l'Analyse en Composantes Principales

L'Analyse en Composantes Principales (ACP [26]) est une méthodologie de réduction de dimensions. De nouveaux axes de représentation sont calculés à partir des données d'apprentissage, et ceux-ci sont déterminés relativement à un critère de variance expliquée dans les données. Représenter les données par une sélection des premiers axes issus de l'ACP permet généralement de retenir une grande partie de la variance observée tout en réduisant la dimension. Une fois les données représentées par ce modèle, la détection de fautes se fait à travers des indices de détection tirant partie de cette compression de l'information.

Analyse en composantes principales en deux dimensions

On présente tout d'abord l'algorithme classique de l'ACP, c'est-à-dire dans le cas statique où les variables n'ont pas d'évolution temporelle.

On considère n observations et m variables, on a ainsi une matrice de données de cette forme :

$$X = [x(1), ..., x(n)]^T \in \mathbb{R}^{n \times m} \tag{2.5}$$

où $x(i) = [x_1(i), ..., x_m(i)]^T \in \mathbb{R}^m$ pour $i \in \{1, ..., n\}$ et $x_j(i)$ est la valeur de la j^e variable pour la i^e observation.

L'ACP propose de déterminer les matrices T et P telles que $X = TP^T$ et $T = XP$ sous la contrainte de maximiser la variance expliquée par chaque axe, et que chacun des nouveaux axes calculés soit orthogonal aux précédents. Cela signifie d'abord trouver la solution v_1 parmi les vecteurs v de norme 1, de :

$$\max_{v \neq 0} \frac{v^T X^T X v}{v^T v}; \tag{2.6}$$

pour ensuite itérativement trouver la solution v_2 vérifiant la même relation, mais située dans l'orthogonal de v_1, et ainsi de suite jusqu'à v_m, qui est orthogonal à tous les précédents. L'espace de projection des données est ainsi de dimensions identiques à l'espace initial, seuls les axes ont été modifiés.

Le résultat principal de l'ACP est que $P \in \mathbb{R}^{m \times m}$ est la matrice des vecteurs propres de la matrice de covariance $\Sigma = \frac{1}{n-1} X^T X \in \mathbb{R}^{m \times m}$. S'agissant d'une matrice symétrique réelle, elle est diagonalisable en base orthonormée. La matrice $T \in \mathbb{R}^{n \times m}$, qui est la matrice des composantes, est ainsi déduite directement de la détermination de P.

Les valeurs propres de la matrice de covariance des données caractérisent la quantité de variance expliquée dans les données par la projection sur les vecteurs propres associés. Ainsi, choisir de projeter les données initiales sur les l premières composantes, que l'on nommera dans ce cas là composantes principales, permet une modélisation rendant compte de $(\frac{\sum_{i=1}^{l}\lambda_i}{\sum_{i=1}^{m}\lambda_i} \times 100)$ % de la variance observée dans l'ensemble d'apprentissage. Les $m-l$ composantes restantes sont nommées composantes résiduelles. De très nombreuses méthodes existent pour fixer le nombre de composantes principales, et ce paramètre joue un rôle important dans les modèles de détection [47].

Prise en compte de l'évolution temporelle

L'ACP classique s'applique à deux dimensions, et ne permet donc pas de tenir compte des données temporelles. En effet, la présence du temps dans les données rajoute une troisième dimension : on dispose de I individus, J variables et K instants de mesure. Il n'est ainsi pas immédiat d'appliquer l'ACP à de telles données. Trois principales méthodologies ont été proposées pour pallier à ce problème.

L'ACP par blocs : Cette ACP est conçue pour les procédés industriels découpés en étapes. Une pondération spécifique des données correspondant aux différentes étapes est calculée par un algorithme itératif. Une étude comparative de différentes versions de cet algorithme est disponible dans [78]. Les procédés du semi-conducteur sont en général effectivement séparés en steps distincts, et correspondent donc aux procédés visés par cette méthodologie. Cependant, le principal obstacle d'application de cette méthode dans le cas des données du semi-conducteur est la présence de steps de courte durée : la collecte de données s'avère incertaine pour les étapes dont la durée est comparable à la fréquence d'échantillonnage.

L'ACP Dynamique : Cette proposition consiste en la création de variables ne représentant qu'une portion du procédé, basée sur un paramètre de décalage. Dans sa version initiale [36], l'ACP dynamique ne traite que d'une seule trajectoire multivariée (les données d'un seul run) : $X = [X_1 ... X_p] \in \mathbb{R}^{n_t \times p}$ où p est le nombre de variables et n_t le nombre d'observations temporelles pour chaque variable. La supposition est que pour tout i, X_i est une série temporelle auto-corrélée, supposée stationnaire, c'est-à-dire que l'espérance est la variance sont constantes au cours du temps, et que l'auto-corrélation entre deux observations ne dépend que de l'écart de

temps entre celles-ci. Cette méthode nécessite le choix d'un paramètre de décalage w, pour construire les matrice de trajectoires X_i^w définies ainsi :

$$X_i^w = \begin{pmatrix} X_i(1) & X_i(2) & ... & X_i(w) \\ X_i(2) & X_i(3) & ... & X_i(w+1) \\ . & . & ... & . \\ . & . & ... & . \\ . & . & ... & . \\ X_i(n_t - w + 1) & X_i(n_t - w + 2) & ... & X_i(n_t) \end{pmatrix}$$

Il s'agit alors d'appliquer l'ACP classique à $X^w = [X_1^w ... X_p^w]$. Cette première proposition permet de gérer la temporalité du procédé, mais n'est pas tri-dimensionnelle, puisque la dimension du nombre de runs a été supprimée. Cette dimension peut être réintroduite en utilisant la moyenne des matrices de corrélations calculées à partir de X^w pour chacun des runs [6]. L'inconvénient principal de cette approche est que les résultats obtenus dépendent grandement du paramètre w. Dans le cadre de l'utilisation aux procédés du semi-conducteur, le choix de ce paramètre est rendu difficile par la grande variabilité de la durée des différents steps. Les changements de steps ont une grande influence sur la fonction d'auto-corrélation : il n'est en général pas vrai que l'auto-corrélation est la même entre deux observations prises à l'intérieur d'un même step et deux observations prises sur des steps distincts.

L'ACP Multi-way : Cette méthodologie propose une projection entière d'une dimension sur une des deux autres. On parle de déroulement d'une dimension. Ainsi, partant de données de dimension $I \times J \times K$, où I désigne le nombre de runs, J le nombre de capteurs et K le nombre de temps, il y a 6 possibilités pour la dimension obtenue après déroulement (sachant que l'ordre des observations et des variables n'a pas d'influence sur la matrice de corrélation) : $I \times JK$, $IK \times J$, $IJ \times K$, $JK \times I$, $J \times IK$, $K \times IJ$. La dimension de gauche désigne l'unité décrivant un individu. Par exemple dans le cas où cette dimension est IK, un individu dans l'ACP est un instant particulier pour un run donné. La dimension de droite désigne ce qui constitue une variable pour l'ACP. Par exemple dans le cas où cette dimension est JK, une variable est un temps particulier pour un capteur donné. Toutes ces possibilités constituent des interprétations différentes de l'influence du temps sur le système. S'agissant de détecter des fautes sur un équipement qui est représenté par les données récoltées pour chaque run, il est évident que la dimension des runs doit se trouver parmi les individus. De plus, les capteurs ne peuvent pas entrer en ligne de compte dans le champ des individus. Si la matrice est déroulée dans le sens $IJ \times K$, alors sous une même

variable de l'ACP il y a un mélange de grandeurs physiques différentes ne s'exprimant pas dans des unités comparables. Les deux propositions restantes sont ainsi les deux seules qui se retrouvent dans la littérature sur la détection de fautes.

Le choix est alors d'individualiser chaque temps pour chaque capteur sous des variables distinctes, ou alors de n'étudier que les corrélations globales des capteurs sur l'ensemble du procédé. Dans le cadre de l'application aux procédés du semi-conducteur, la première approche n'est pas toujours réaliste informatiquement. En effet, parmi les différents procédés réalisés par les équipements du semi-conducteur, certains sont très longs, comme par exemple ceux de diffusion (voir Section 1.1.1), et il en résulte des données de très grandes dimensions. Un run typique pour un procédé de diffusion est constitué des données d'une cinquantaine de capteurs pour 3000 instants de mesure. Appliquer le déroulement du type $I \times JK$ dans cette situation nécessite alors la diagonalisation d'une matrice de dimensions 150000 $\times$ 150000, rendant les calculs très coûteux. L'ACP multi-way avec le déroulement du type $IK \times J$ est donc un choix permettant des temps de calculs réduits tout en restant cohérent relativement à l'objectif de détection. Pour la suite, lorsque nous désignons l'ACP multi-way, nous parlons de ce déroulement particulier.

Indices de détection

Pour effectuer de la détection de fautes avec un modèle d'ACP, on détermine un indice de détection, qui permet de classer les observations représentatives d'un défaut : l'indice est un indicateur statistique calculé pour chaque observation, et dont on encadre la valeur en fonctionnement normal relativement à l'ensemble d'apprentissage. Nous présentons les deux indices les plus couramment utilisés avec l'ACP. Ceux-ci appartiennent à la catégorie des indices quadratiques, car dépendant du carré des observations projetées : on peut constuire d'autres indices similaires ainsi que des combinaisons de ces indices [28, 84]. On reprend les notations définies en début de section, et on applique ces indices dans le cas de l'ACP multi-way décrite au paragraphe précédent.

Indice SPE : L'indice le plus couramment utilisé avec l'ACP est l'indice SPE (Square Prediction Error) [50]. Celui-ci porte uniquement sur les composantes résiduelles. La définition de celui-ci est la suivante, pour la i^{e} observation :

$$SPE(i) = \tilde{x}^T(i)\tilde{x}(i). \tag{2.7}$$

Ici $\tilde{x}$ désigne la projection du vecteur de données sur l'espace résiduel (défini par le choix du nombre de composantes principales). Dans le cas de l'ACP multi-way, cet indice de détection est ainsi un vecteur, constitué d'une valeur par temps, puisque la i^{e} observation représente un seul temps du run. On dispose donc pour chaque run d'autant de valeurs de l'indice que d'instants de mesure dans le procédé. On définit alors la limite du SPE également comme un vecteur :

$$\left(\delta_i^2\right)_{i=1\ldots K} = \left(\frac{v_i}{m_i}\chi^2_{\frac{2m_i^2}{v_i},\alpha}\right)_{i=1\ldots K}, \tag{2.8}$$

où m_i et v_i représentent la moyenne et la variance du SPE pour l'ensemble des runs d'apprentissage à l'instant i. Le paramètre α est l'erreur de Type-I choisie. À chaque instant de temps, un retour sur les valeurs du SPE dans l'ensemble d'apprentissage permet d'adapter la limite. L'objectif est de tolérer une marge d'erreur au-dessus des valeurs observées lors de l'apprentissage, cette marge d'erreur étant calculée avec la loi du χ^2 sous l'hypothèse que les composantes résiduelles suivent une loi normale centrée. L'hypothèse soutenue pour utiliser le SPE est que l'espace principal capte l'essentiel du processus, et que donc il ne reste que du bruit dans l'espace résiduel. Une perturbation des données doit alors engendrer une augmentation importante du bruit, que l'on peut détecter. Dans le cadre de l'ACP multi-way, avec le déroulement évoqué, cette hypothèse n'est pas évidente. Dans la Section 3.4.2, nous montrons que l'hypothèse selon laquelle un bruit est observé sur l'espace résiduel dans ce cas n'est que rarement vérifiée.

Indice $\mathbf{T^2}$: Un autre indice utilisé fréquemment dans la littérature est l'indice T^2 de Hotelling [50]. Celui-ci porte uniquement sur l'espace principal. Sa définition est identique à celle du SPE, sauf qu'au lieu de prendre les projections des données sur les composantes résiduelles, ce sont les projections sur les composantes principales qui sont utilisées. Pour un run i, sa définition est :

$$T^2(i) = \hat{x}^T(i)\hat{x}(i). \tag{2.9}$$

Ici $\hat{x}$ désigne la projection du vecteur de données sur l'espace principal. Dans le cas de l'ACP multi-way, comme pour le SPE, on dispose donc pour chaque run d'autant de valeurs de l'indice que d'instants de mesure dans le procédé. De façon identique, on définit la limite :

$$\left(\delta_i^2\right)_{i=1\ldots K} = \left(\frac{v_i}{m_i}\chi^2_{\frac{2m_i^2}{v_i},\alpha}\right)_{i=1\ldots K}, \tag{2.10}$$

où m_i et v_i représentent la moyenne et la variance du T^2 pour l'ensemble des runs d'apprentissage à l'instant i. Le paramètre α est l'erreur de Type-I choisie.

2.3.4 Une méthodologie de détection basée sur l'ACP multi-way

Dans la thèse de Thieullen [70], une méthodologie basée sur l'ACP multi-way appliquée aux données du semi-conducteur est proposée. Elle est conçue en deux temps, avec d'abord la construction d'un modèle sur un ensemble d'apprentissage donné, puis ensuite une phase de mise à jour permettant d'intégrer la notion de maintenance des équipements.

La méthode Exponentially-weighted Hybrid Multi-way Principal Component Analysis (E-HMPCA) [85] est une méthodologie dérivée de l'ACP Multi-way. Les étapes de cette méthode sont les suivantes :

— Déroulement de la matrice des données, suivant la dimension décrite Section 2.3.3 ;
— Application d'un filtre EWMA (Exponentially Weighted Moving Average) sur les composantes : pour chaque run, une dépendance temporelle spécifique est créée. Une fois les données projetées sur les composantes, chaque observation est filtrée relativement aux précédentes. On utilise la notation T pour désigner la matrice des données projetées, et on désigne par T_k les données de la matrice T associées au temps k. La matrice filtrée T^f est définie par $T_1^f = T_1$ et $T_k^f = \lambda T_k + (1-\lambda)T_{k-1}^f$ pour $k > 1$. Le paramètre λ qui contrôle l'intensité du filtrage est à fixer par l'utilisateur de la méthode, et susceptible d'être modifié pour chaque type d'équipement ou de recette.
— Choix d'un nombre l de composantes principales.
— Utilisation de l'indice SPE, qui est donc filtré puisque cet indice est calculé à partir des composantes résiduelles qui ont toutes été filtrées.

Une fois le modèle construit suivant l'E-HMPCA, la méthodologie de l'ACP récursive est appliquée à chaque maintenance de l'équipement. La maintenance perturbe l'équilibre qui avait été observé dans l'ensemble d'apprentissage, ce qui fait que le modèle appris n'est plus utilisable tel qu'il est : l'indice SPE détecte les runs dont les données ont été récoltées après la maintenance. Il faut mettre à jour le modèle pour tenir compte de la maintenance. L'algorithme de l'ACP récursive permet pour un modèle ACP fixé d'introduire l'influence de nouvelles observations distinctes. Ces nou-

velles observations sont pondérées par un paramètre d'oubli μ entre 0 et 1, choisi par l'utilisateur, qui indique l'importance à accorder aux nouvelles observations relativement à l'ensemble d'apprentissage initial. Tout le modèle est mis à jour : les paramètres de normalisation de la matrice de covariance et la matrice de covariance. Ensuite, les autres paramètres du modèle sont recalculés : la matrice de projection dans l'espace des composantes, le nombre de composantes principales à sélectionner et l'indice SPE ainsi que ses limites. Cette méthodologie permet de faire appel à d'anciennes données sans avoir besoin de les stocker, ce qui peut s'avérer utile pour des données volumineuses dans le cas d'une utilisation pour un grand nombre d'équipements.

Le défaut principal de cette méthode réside dans le fait qu'aucun test préalable n'est fait avant de mettre à jour le modèle après une maintenance. En effet, dans la Section 1.2.2, nous évoquons le fait que l'intervention de maintenance peut être source d'erreur humaine. Il n'est pas sûr que les données récoltées juste après la maintenance correspondent à un bon fonctionnement de l'équipement. Il s'agit là de l'hypothèse fondamentale de la détection par apprentissage statistique mono-classe : si des données de mauvais fonctionnement sont introduites dans le modèle, alors la faute est apprise et ne pourra pas être détectée. Le fonctionnement correct pourrait même être vu comme défectueux avec un tel modèle.

2.4 Conclusion

Les données récoltées dans l'industrie du semi-conducteur sont des trajectoires temporelles multi-variées, représentant des paramètres physiques caractéristiques du procédé réalisé par un équipement. Une des contraintes importantes pour le traitement de telles données est la variabilité de la dimension temporelle : deux réalisations d'un même procédé ne sont que rarement synchrones. La démarche la plus employée dans la littérature est d'aligner temporellement ces données sur une base commune, afin de les rendre comparables à tout moment du procédé. Les méthodologies basées sur l'algorithme Dynamic Time Warping sont les plus efficaces pour réaliser cette tâche.

Le problème de la détection de faute relève de la classification par apprentissage statistique supervisé mono-classe. Le principe est de caractériser statistiquement des données de référence. Lorsque des données correspondant à un nouveau run sont étudiées, elles sont classées comme fautives lorsqu'elles ne partagent pas ces caractéristiques : il s'agit d'une définition négative des fautes.

Les approches les plus observées dans la littérature sont celles basées sur des méthodologies non-linéaires, comme les SVM, et celles, linéaires, basées sur l'Analyse en Composantes Principales. Les premières présentent le désavantage d'un aspect boîte noire : il est difficile d'identifier la source des détections. Ces deux types d'algorithmes n'intègrent pas de gestion de la maintenance des équipements. Il s'agit d'un point crucial pour l'étude de données issues de l'industrie du semi-conducteur, car les maintenances sont fréquentes, leurs effets ne sont pas toujours prévisibles, et elles peuvent faire l'objet d'erreurs humaines. Il faut mettre à jour le modèle conçu, mais il faut également tenir compte du risque que les nouvelles données récoltées ne soient pas nécessairement représentatives d'un bon fonctionnement.

Nous proposons, dans les Chapitres 3 et 4, une méthodologie de détection de fautes tenant compte des contraintes s'exerçant dans l'industrie du semi-conducteur, et tout particulièrement de celles imposées par la maintenance des équipements.

Chapitre 3

Gaussian Time Error : une méthodologie de détection de fautes en temps réel

Dans ce chapitre, nous présentons une méthodologie de détection des fautes en temps réel intitulée Gaussian Time Error. Il s'agit de la principale contribution de cette thèse. Nous avons d'abord introduit cette méthodologie dans [40], et apporté des améliorations dans [41].

3.1 Introduction

Nous définissons tout d'abord les enjeux de l'élaboration d'une méthodologie de détection adaptée aux équipements de l'industrie du semiconducteur, et présentons les étapes de la méthodologie Gaussian Time Error que nous proposons. La route de fabrication d'un wafer est constituée de plusieurs centaines d'exécution de recettes. Les mesures sur produits, effectuées entre les opérations, s'avèrent contraignantes : elles ralentissent la production et peuvent dans certains cas être destructrices. Seule une sélection de produits est ainsi mesurée, et ce de façon irrégulière, rendant parfois difficile la décorrélation de la mesure obtenue avec les étapes précédentes de la route. En effet, la mesure récoltée est le résultat de plusieurs opérations successives. Les données FDC permettent une mesure en temps réel de l'équipement pendant qu'il traite un ou plusieurs wafers, et donnent ainsi des informations précises sur le déroulement complet du procédé. Il est alors possible d'isoler les modifications observables dans ces données. Les défaillances d'un équipement peuvent alors potentiellement être identifiées en termes de paramètres physiques et de steps des recettes concernées.

Il n'y a pas de courbes théoriques de référence qui pourraient être associées aux différentes recettes d'une stratégie donnée. Dans un tel cas, une étude relative à ces courbes pourrait être menée, mais il faut ici découvrir par les données ce que constitue un procédé convenable. L'apprentissage statistique sur des jeux de données de référence est un moyen de permettre une modélisation en l'absence de référence absolue. Cette approche permet d'extraire les informations en termes de paramètres physiques associés à un état stable de l'équipement sélectionné : l'objectif est alors de maintenir les présentes caractéristiques de l'équipement proches des caractéristiques relevées dans l'ensemble d'apprentissage stable. L'inconvénient principal de cette approche est que l'on rejette la possibilité d'un état de fonctionnement correct, c'est-à-dire qui produit des wafers fonctionnels, mais distinct de l'état de fonctionnement observé dans l'apprentissage.

L'apprentissage statistique appliqué aux données FDC se heurte à un obstacle majeur : l'instabilité du système. La première instabilité est le reréglage perpétuel de l'équipement, aussi bien via la régulation interne que via la régulation appliquée par l'industriel (par les boucles de régulation). La consigne pour certains paramètres physiques peut ainsi évoluer au cours du temps. Cette instabilité peut de façon générale être estimée dans l'apprentissage de façon indirecte, par la variabilité observée. La deuxième instabilité, la plus importante, est due à la maintenance des équipements. En effet, les équipements de l'industrie du semi-conducteur sont fréquemment mis à l'arrêt pour effectuer des nettoyages, des changements de pièces, ou différents réglages. Ces maintenances peuvent être dites préventives, dans le cas où la maintenance est planifiée relativement à une durée de fonctionnement ou un nombre de runs effectués, ou correctives, dans le cas où une irrégularité a été observée qui nécessite réparation. Dans les deux situations, l'impact sur les données FDC est peu prévisible. En effet, le changement de conditions induit par les modifications faites lors de la maintenance, peut se manifester d'un grand nombre de façons, à travers les différents paramètres physiques et au cours des différents steps. Ces changements rendent difficile voire impossible le mélange de données issues de deux périodes de temps séparées par une maintenance : les écarts entre ces données peuvent s'avérer tout aussi larges que les écarts observés dans le cas d'une faute nécessitant réparation. Mélanger ces données dans un même ensemble d'apprentissage pourrait amener à surestimer la variabilité d'un run à un autre. L'emploi d'un modèle fixe, où aucune maintenance n'a lieu pendant la période d'apprentissage, peut amener à détecter des changements normaux dus à une maintenance comme étant des fautes du procédé. Ainsi, il est nécessaire de concevoir une méthodologie de détection qui gère la maintenance de façon automatique, car les maintenances

sont fréquentes et ne sauraient être gérées au cas par cas. De plus, parmi les changements observés suite à une maintenance, seuls les changements entrainant une défaillance de l'équipement doivent être signalés. La détection de faute doit pouvoir se poursuivre suite à la maintenance, et en même temps il faut se prononcer sur une éventuelle erreur lors de celle-ci : il n'est pas garanti que l'équipement redémarre correctement après celle-ci, du fait d'éventuelles erreurs humaines (voir Section 1.2.2).

La démarche employée ici s'effectue en quatre temps :

— Pré-traitement des données : les mesures industrielles brutes sont temporellement alignées avant d'être exploitées ;

— Élaboration d'un modèle statistique : nous utilisons une technique issue de l'analyse en composantes principales pour proposer un modèle représentatif des données ;

— Détection de défauts par l'utilisation d'un indice : on élabore un critère, que nous nommons Gaussian Time Error, permettant de chiffrer les écarts au modèle déterminé ;

— Mise à jour de l'indice de détection lorsqu'une intervention de maintenance a lieu : la maintenance des équipements (aussi bien prédictive que corrective) apporte fréquemment des modifications dans les données, empêchant de conserver un modèle appris avant celle-ci. Chaque maintenance est testée, et si aucune faute n'est détectée sur la base des premiers runs, certains paramètres du modèle sont ajustés.

On se propose de travailler par réunion de recettes ne différant que par la longueur d'un ou plusieurs steps, ce que nous avons défini comme étant une stratégie de collecte à la Section 2.1. L'alignement temporel des données permet cette réunion, et évite ainsi la multiplication du nombre de modèles de détection au delà de ce qui peut être géré en termes de logistique. De plus, cela permet de faciliter la phase d'apprentissage, puisque certaines recettes peu fréquentes peuvent alors être surveillées, alors qu'il serait difficile de réunir un ensemble d'apprentissage convenable pour celles-ci. Un résumé de la méthodologie de détection de fautes proposée est présenté sur la Figure 3.1.

3.2 Constitution d'un ensemble d'apprentissage

Pour créer un modèle, il est nécessaire d'employer des données correspondant à un fonctionnement correct de l'équipement. Dans l'industrie du

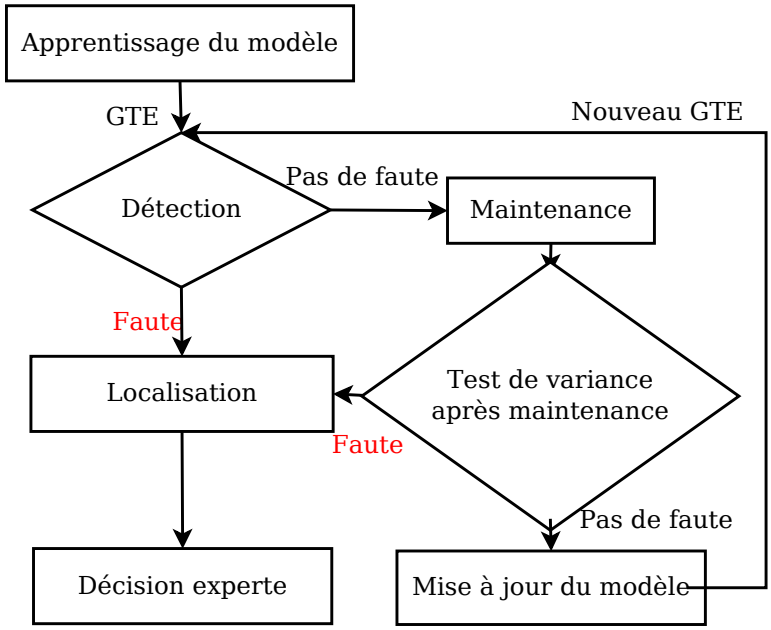

FIGURE 3.1 – Schéma global du fonctionnement de la méthodologie.

semi-conducteur, il est en général difficile d'être certain que l'équipement effectue parfaitement la tâche qu'il exécute : il est plus simple de tirer négativement des conclusions, à savoir de dire si l'on a observé ou non un impact de l'équipement sur les produits. Nous proposons alors d'employer des données correspondant à des runs lointains, plusieurs mois dans le passé, dont les wafers associés ont été testés en sortie d'usine et fonctionnant normalement. Il est important de s'assurer qu'aucune discontinuité de production n'a eu lieu dans la période correspondante : pas d'intervention de maintenance, pas de faute de l'équipement repérée. La période couverte doit être la plus large possible, afin de pouvoir tenir compte des régulations internes de l'équipement qui interviennent pour contrer sa dégradation entre deux maintenances. Dans les exemples étudiés, voir Section 3.6, nous employons les données de 100 runs pour constituer l'ensemble d'apprentissage. Les contraintes pour les choix de taille sont doubles : il faut tenir compte de la représentativité de tels ensembles, mais également de la contrainte industrielle. En effet, pour certains équipements et certaines stratégies, le nombre total de runs effectués entre deux maintenances consécutives ne permet pas de choisir de plus grande quantité pour l'apprentissage. Nous montrons dans la Section 3.6 que ce choix est pertinent du point de vue

des résultats de détection : sélectionner plus de runs n'apporte pas de meilleures performances sur l'ensemble de test.

3.3 Le pré-traitement des données

Une étape préalable à l'analyse des données est l'alignement temporel. Les contraintes techniques de la collecte effectuée par les capteurs peuvent générer de nombreux problèmes, les deux principaux étant que le nombre de points collectés au total est variable pour chaque plaque, et ensuite que les instants de collecte sont rarement strictement identiques. La notion de temps exact dans la recette n'est pas un apport souhaitable à l'analyse : dans notre cas, nous mélangeons des recettes dont les steps peuvent varier en durée. Rajouter la notion de temps complexifierait l'analyse, là où d'autres instances de détection peuvent mieux s'appliquer à cette problématique, comme la régulation interne de l'équipement par exemple. Travailler à remettre les observations d'une même stratégie à une échelle commune de déroulement du procédé s'avère une solution à ce problème.

3.3.1 Méthodologie d'alignement et détermination d'une trajectoire de référence

Dans la Section 2.2, nous traitons des méthodes de synchronisation existantes. Dans ces travaux, nous appliquons l'algorithme DDTW, sans modification par rapport à ce qui est présenté à la Section 2.2.3. Le choix de la trajectoire de référence est effectué de manière plus simple qu'avec l'algorithme DBA. En effet, parmi les limitations de l'algorithme DBA décrites, son importante dépendance envers la trajectoire d'initialisation de l'algorithme fait que dans notre cas d'application, aucun gain n'est observé à utiliser DBA pour calculer une référence plutôt qu'en choisissant directement cette trajectoire d'initialisation comme référence. L'application des méthodologies de détection proposées dans ces travaux donne en général des résultats strictement identiques entre l'utilisation de DBA pour calculer la référence et l'utilisation de la trajectoire d'initialisation de DBA comme référence. Le gain de temps est substantiel, DBA étant un algorithme présentant un coût important en termes de calcul.

On se propose d'employer l'étape d'alignement pour élaborer une méthodologie associant un modèle de détection unique à des recettes ne différant que par la durée d'un step, c'est-à-dire les recettes regroupées sous la notion de stratégie. C'est un cas très fréquent par exemple pour les ma-

chines de dépôt, où déposer une couche plus épaisse se fait avec la même recette que pour une couche moins épaisse, à ceci près qu'un des steps est plus long.

En pratique, nous proposons la solution suivante : dans le cas de l'alignement DDTW avec les contraintes décrites, une donnée récoltée pendant le traitement d'une plaque ne peut être supprimée que dans le cas où le nombre total d'observations pour la plaque dépasse le nombre total d'observations de la référence. Sinon, on ne peut avoir que des dédoublements d'observations, sans suppression. Il suffit alors de déterminer la trajectoire de référence en se basant sur les données de la recette la plus longue. Ainsi un défaut sera toujours visible : sa localisation exacte risque d'être plus floue du fait des dédoublements, sans que cela soit rédhibitoire puisque le step de la recette pourra encore être correctement identifié. Notre proposition est ainsi de prendre comme trajectoire de référence la trajectoire la plus intègre, c'est-à-dire ayant le plus d'observations relativement à l'échantillonnage prévu, parmi les trajectoires associées à la recette la plus longue de l'ensemble d'apprentissage. Il est alors important de choisir spécifiquement des données d'apprentissage recouvrant la recette la plus longue de la stratégie pour laquelle le modèle de détection est construit.

Nous ne proposons pas ici de solution pour regrouper sous un même modèle toutes les recettes d'un équipement sans hypothèse sur leur similarité. Dans le cadre de la méthodologie présentée ici, cela n'est pas possible, car différentes recettes peuvent être associées à différentes variables physiques, et à des trajectoires temporelles très différentes pour les variables communes : l'alignement perdrait toute sa pertinence.

3.3.2 Alignement en présence de variables sans forme caractéristique

Proposition méthodologique

Parmi les variables étudiées, certaines peuvent ne pas avoir de lien direct avec le déroulement du processus : c'est le cas par exemple d'une pression qui doit rester stable tout au long de la recette sauf en cas de fuite. De telles variables gênent la précision de l'alignement. En effet, leur valeur intervient dans les calculs de l'algorithme DDTW alors que les variations observées d'une plaque à l'autre ne doivent pas nécessairement être synchrones. Au lieu d'exclure complètement ces variables, ce qui pourrait amener à ne pas détecter certains types de fautes, nous proposons de les repérer pour les exclure seulement de la procédure d'alignement. Ces variables sont alors réintégrées une fois l'alignement effectué, en utilisant

l'association temporelle à la référence qui a été calculée.

Étudier la relation d'une variable au temps directement n'est pas immédiat dans le cas des procédés du semi-conducteur. Pour la plupart des recettes, une faible proportion des steps représente la majeure partie du temps de la recette. Étudier le temps en faisant abstraction de la notion de step peut donc revenir à privilégier certains steps par rapport à d'autres. De plus, un même temps de recette peut ne pas toujours correspondre à un même step de la recette, car un step peut se prolonger jusqu'à l'obtention de certaines conditions. Dès lors, nous proposons d'étudier la relation entre les valeurs des variables non alignées, et le step de la recette correspondant : si une variable n'a pas de trajectoire temporelle caractéristique alors elle ne dépend pas de la variable qui indique le step de la recette. Réciproquement, une variable dont les valeurs ne dépendent pas du step n'aura a priori pas de forme caractéristique.

La démarche proposée ici est d'employer l'Analyse de Variance (ANOVA), pour étudier l'impact du step, qui est une variable qualitative, sur la valeur de chacune des variables récoltées pour une plaque donnée. L'ANOVA est une méthodologie statistique très classique, dont le détail peut être trouvé par exemple dans [64].

Soit $(X_k)_{k\leq K}$ une variable de longueur K récoltée pour un wafer, et soit $(S_k)_{k\leq K}$ les steps associés à chaque mesure. On dispose de $s \leq K$ steps distincts dans la recette étudiée. Dans la pratique, on n'observe pas nécessairement les s steps pour chaque plaque, certains steps très courts étant parfois absents dans le cas d'une ou plusieurs valeurs manquantes. Pour supprimer l'influence de la durée du step, on remplace le vecteur X par le vecteur Y, d'au plus s valeurs, correspondant aux moyennes de X sur chacun des steps disponibles. Pour I plaques, on réunit alors les informations de la variable dans un vecteur d'au plus sI valeurs, toutes associées à un step, et on effectue l'ANOVA de ce vecteur contre la variable "step". La p-valeur du test obtenue nous donne la probabilité pour une variable effectivement indépendante du step de produire les valeurs observées.

Pour les données étudiées, les p-valeurs de la majeure partie des variables sont extrêmement faibles (l'exemple ci-après illustre ce fait). Nous proposons ainsi d'exclure de l'alignement toute variable dont la p-valeur est au-dessus de 10%. D'après notre expérience sur les données du semi-conducteur, la conséquence liée à l'exclusion d'une variable qui aurait pu être conservée est en général très faible voire inexistante. En effet, ce sont les synchronisations des variations déterministes les plus larges qui définissent un bon alignement. Les variables soumises à ce type de variations ne sont pas exclues dans cette procédure, étant donnée la plage de p-valeurs qui leur est associée.

Exemple

On étudie ici l'exemple d'un équipement de gravure doté de 77 capteurs. L'application sur l'ensemble de ces 77 capteurs de la méthodologie d'alignement proposée fournie des résultats médiocres d'alignement : le déroulement du procédé sur les différents runs n'est pas du tout synchronisé. Pour illustrer cela, on applique l'algorithme d'alignement tel que décrit, et on observe le résultat avec le graphique Figure 3.2, pour 50 runs superposés, représentant une variable de flux de gaz en fonction du temps dont les différents steps sont censés être synchronisés d'un run à l'autre.

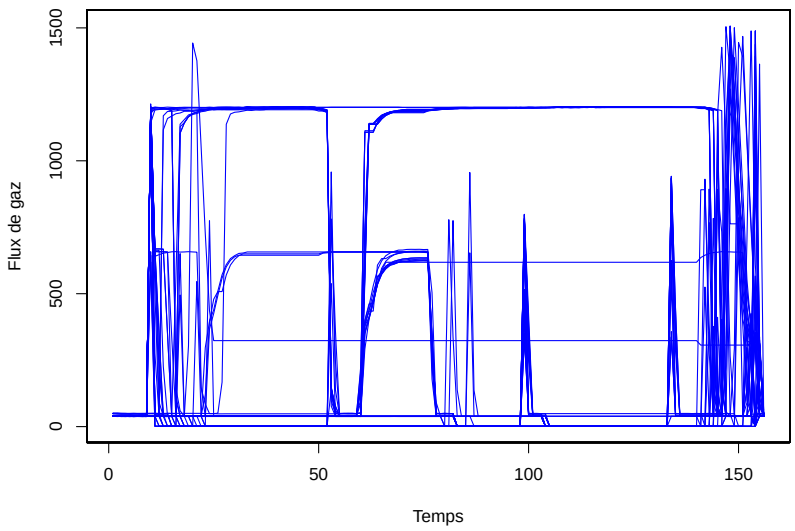

FIGURE 3.2 – Alignement d'une variable dépendante des steps en présence de variables indépendantes des steps, pour 50 runs.

On applique alors la méthodologie décrite, en effectuant pour chaque capteur une ANOVA basée sur ces 50 runs, où la variable à expliquer est la valeur moyenne sur le step, et la variable explicative est le numéro du step. Le test basé sur les p-valeurs nous permet d'exclure 36 variables. La Figure 3.3 représente un exemple de capteur exclu : on observe bien l'absence de changement de valeurs pour chaque run à travers le temps, l'objectif sur la pression récoltée par ce capteur étant de maintenir une valeur stable. Les capteurs qui sont conservés correspondent en général à des p-valeurs très

faibles. La p-valeur médiane pour ceux-ci vaut 0.001, et le quantile à 25% est de 10^{-6}. La Figure 3.4 montre le résultat obtenu, pour le même flux de gaz que pour la Figure 3.2, en appliquant le même algorithme d'alignement (sans changer de trajectoire de référence ni aucun autre paramètre) en ayant exclu les 36 capteurs : les différents steps sont correctement synchronisés.

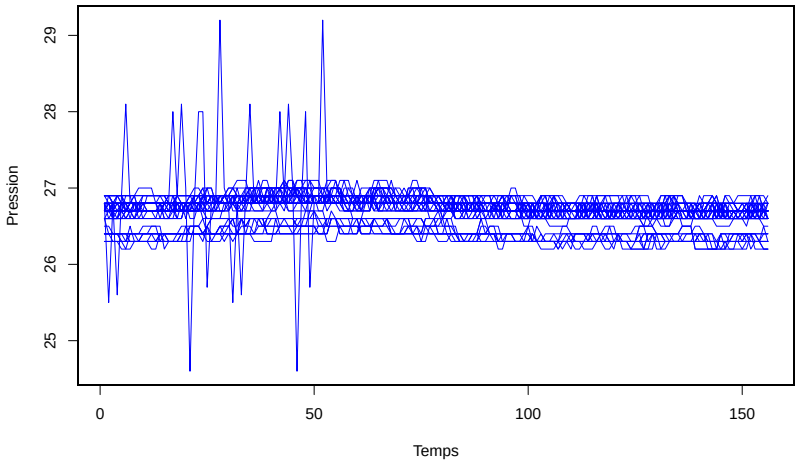

FIGURE 3.3 – Variable de pression indépendante des steps, pour 50 runs.

3.3.3 Particularité des données d'équipements d'implantation ionique

Le cas étudié dans la Section 3.3.2 ne peut s'appliquer qu'à des recettes disposant de steps, et où il est attendu d'une partie non-négligeable des variables d'avoir une forme caractéristique. Dans les procédés d'implantation ionique, la majeure partie des recettes ne disposent pas de la notion de step, et une très grande majorité des variables n'ont pas de forme spécifique attendue. Cela est dû au fait que le procédé vise à créer un état stable permettant l'implantation : les grandeurs physiques observées sont très fréquemment des perturbations a priori aléatoires, empêchant d'utiliser la méthode d'alignement DDTW proposée à la Section 2.2.3.

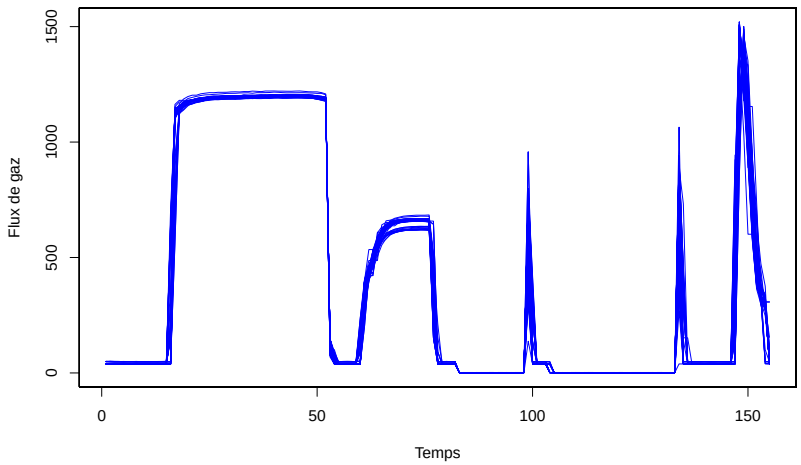

FIGURE 3.4 – Alignement d'une variable dépendante des steps en l'absence des variables indépendantes des steps, pour 50 runs.

Pour aligner ces données, nous proposons d'utiliser l'algorithme DTW, avec la distance euclidienne, appliqué non pas de façon multivariée avec l'ensemble des grandeurs physiques, mais de façon mono-variée : le seul paramètre que l'on aligne est le pourcentage de complétion du processus d'implantation. Ainsi, les données observées dans la matrice obtenue après alignement sont liées directement à la progression du processus : les données observées à un temps k donné, sont liées au pourcentage de complétion que la trajectoire de référence a au temps k, et donc à l'observation de la trajectoire à aligner correspondant au pourcentage de complétion le plus proche. Le temps est un paramètre variable d'une plaque à l'autre, et il introduit ici un bruit dans l'observation des données.

Les méthodologies présentées par la suite ne sont pas applicables pour de tels équipements, car les hypothèses faites dans la Section 3.4.1 ne sont pas vérifiées : les données ne décrivent pas un unique processus sous-jacent que l'on peut caractériser par des lois normales. Des re-réglages automatiques de l'équipement, intervenant en général plusieurs fois par jour, permettent de maintenir le résultat sur les produits constant. Ceux-ci empêchent une caractérisation des données récoltées sur l'équipement, car les variations observées ne sont pas suffisamment prévisibles, et aucun temps défini ne peut être accordé pour apprendre les changements observé, qui sont parfois immédiatement corrigés.

3.4 Modélisation statistique des données

On considère un ensemble d'apprentissage pour une stratégie de collecte fixée, récolté pendant une phase où l'équipement est stable : aucun défaut connu, aucune intervention de maintenance. Cette ensemble est constitué des données de I runs, associées à J capteurs, et à K temps, une fois le pré-traitement effectué tel que décrit Section 3.3. On note X la matrice tri-dimensionnelle, de dimensions $I \times J \times K$, contenant l'ensemble de ces données. On note X_k la matrice de dimensions $I \times J$ contenant les données collectées au temps $k \leq K$.

3.4.1 Démarche de construction d'un test d'hypothèse

On fait l'hypothèse suivante : pour tout k, on a un vecteur moyen $m_k \in \mathbb{R}^J$ et une matrice de covariance $S_k \in \mathbb{R}^{J \times J}$ tels que X_k suive la loi gaussienne $N(m_k, S_k)$. Dès lors, la loi de probabilité de X en tant que mélange de Gaussiennes peut s'écrire : $X \sim \sum_{k=1}^{K} \frac{1}{K} N(m_k, S_k)\delta_k$ où δ_k est le delta de Dirac. Cette hypothèse pose des contraintes faibles sur le système : dans le cas d'un fonctionnement normal, les données récoltées du procédé correspondent au suivi d'une trajectoire idéale, avec des écarts à cette trajectoire tirés de façon indépendante dans le temps. Les contraintes temporelles sont fixées par l'évolution des moyennes et des covariances à travers le temps d'avancement de la recette.

On se propose de construire un test statistique permettant de décider si les données récoltées pour un nouveau run de dimensions $J \times K$ sont issues ou non de la distribution. Pour construire ce test, nous nous proposons de diagonaliser les matrices S_k, afin de ramener ce test multi-dimensionnel à une combinaison de tests uni-dimensionnels. Il y a alors 3 grandes possibilités :

1. La possibilité la plus simplificatrice est d'ignorer les corrélations entre les différents capteurs : on suppose que, pour tout k, S_k est diagonale ;
2. Une deuxième possibilité est de supposer que les S_k soient diagonalisables dans une base commune à l'ensemble des temps k : il s'agit de déterminer une unique matrice de changement de base, caractérisant les corrélations globales entre les capteurs ;
3. La dernière possibilité est de déterminer une diagonalisation indépendante pour chaque S_k, ce qui correspond à déterminer une matrice de changement de base différente à chaque temps k.

La possibilité 1 est irréaliste dans notre cas. On vérifie dans nos données que les coefficients hors diagonale des différentes matrices de covariance

calculées à chaque instant ne sont pas négligeables : il existe clairement une corrélation entre les données récoltées par les différents capteurs d'un même équipement. Intuitivement, il s'agit des corrélations physiques entre les différents capteurs choisis : par exemple introduire un flux de gaz modifie la pression et la température de la chambre.

La possibilité 3 correspondrait à une situation de sur-apprentissage, où on appliquerait une ACP différente à chaque temps de la recette. En effet, chaque matrice de changement de base est constituée de J^2 coefficients, et J^2 est dans la pratique proche de I, voire supérieur. Par exemple, si les données d'apprentissage disponibles correspondent à 100 runs, et que l'équipement dispose de plus de 10 capteurs, ce qui est un cas très fréquent, alors on détermine plus de coefficients pour effectuer les changements de bases qu'il n'y a de valeurs en entrée du modèle. De plus, un problème fréquemment rencontré est celui de la présence d'écarts-types inter-runs nuls pour certains temps de certains capteurs. Ce problème est détaillé à la Section 3.4.4. Pour tous les temps où au moins un capteur présente un écart-type nul, on ne peut pas calculer de matrice de corrélation représentative des données, ce qui est un obstacle à l'application de l'ACP. Enfin, les problèmes de collectes observés dans l'ensemble d'apprentissage, pouvant engendrer des alignements imparfaits, affectent grandement l'estimation des matrices de changement de base. Les capacités de détection du modèle sont alors réduites. Nous montrons dans l'exemple d'illustration ci-après, basé sur les mêmes données que l'exemple de la Section 3.6.1, que ces deux approches ne sont pas efficaces dans la pratique.

On se propose ainsi d'élaborer un test à partir de la deuxième possibilité, qui est de diagonaliser les matrices de covariance à chaque temps à partir d'un changement de base commun. C'est la démarche que l'on suit en appliquant l'ACP multi-way avec le déroulement proposé à la Section 2.3.3. Il s'agit d'appliquer l'ACP à la matrice tri-dimensionnelle déroulée sur l'espace de dimension $IK \times J$. Pour des raisons métier, variant d'une recette à l'autre, l'approximation faite en se basant sur cette hypothèse peut varier. Par exemple, certaines parties de l'équipement sont isolées les unes des autres, impliquant une très faible dépendance entre les capteurs qui y sont respectivement associés. Cependant, une corrélation entre ces variables peut dans certains cas être globalement observée par l'intermédiaire de la variable implicite rendant compte de l'avancée du procédé. De plus, certains steps se font en l'absence totale de certains paramètres influant sur le procédé. C'est le cas par exemple pour une recette où les premiers steps sont une préparation de l'équipement avant introduction d'un gaz : tant que le gaz n'est pas introduit, il n'a strictement aucune influence sur le reste des variables physiques qui sont récoltées.

Exemple d'illustration sur des données industrielles

Nous prenons un exemple basé sur un équipement de dépôt, pour illustrer l'écart de modélisation fait en prenant ces 3 approches. On considère les données de 100 runs, correspondant chacun à 15 capteurs pour 115 temps de mesure, après alignement. Il s'agit des données employées pour l'apprentissage du modèle dans l'exemple de la Section 3.6.1.

On calcule d'abord un modèle du type $X \sim \sum_{k=1}^{K} \frac{1}{K} N(m_k, S_k)\delta_k$. Dans ce cas, sur 100 des 115 temps de mesure (87% de la recette), au moins un capteur présente un écart-type nul d'un run à l'autre. Pour certains de ces temps, il y a jusqu'à 6 capteurs en même temps présentant cette caractéristique, il n'est donc pas possible de tous les ignorer car la perte d'information serait trop grande. Ainsi, on ne peut pas déterminer les modèles d'ACP basés sur la possibilité 3 proposée.

Pour comparer les possibilités 1 et 2, on se propose de calculer pour chaque temps k le rapport en norme de Frobenius entre la matrice des éléments non-diagonaux de S_k (c'est-à-dire la matrice S_k à laquelle on a soustrait sa diagonale) et la matrice constituée de la diagonale de S_k. Pour rappel, pour une matrice $M = (M_{ij})$ définie dans $\mathbb{R}^{a\times b}$ la norme de Frobenius est la norme euclidienne de cette matrice vue comme un vecteur de $\mathbb{R}^{ab}$, c'est-à-dire $\sqrt{\sum_{ij} M_{ij}^2}$. Pour les matrices de covariance initiales S_k, en moyenne cette valeur est d'environ 30%. Si on remplace S_k par sa projection $P^T S_k P$ où P est la matrice de projection calculée avec l'ACP multiway (voir Section 2.3.3), alors cette valeur moyenne descend à environ 5%. Ainsi, la perte d'information en négligeant les termes de covariance est nettement plus faible en faisant une diagonalisation sur une base commune, ce qui justifie le choix de cette solution.

3.4.2 Déroulement des données et obtention de la matrice de covariance

Nous appliquons ainsi un déroulement de la matrice tri-dimensionnelle issue de la récolte initiale des données. On note I le nombre de runs, J le nombre de capteurs et K le nombre d'instants de mesure. On construit une matrice X de J colonnes et KI lignes : la Figure 3.5 illustre ce déroulement.

On considère KI observations collectées selon J capteurs. On construit une matrice de la forme :

$$X = (x_{i,j,k}) \in \mathbb{R}^{KI\times J}, \tag{3.1}$$

où $x_{i,j,k}$ pour $i \in \{1, ..., I\}$, $k \in \{1, ..., K\}$ et $j \in \{1, ..., J\}$ est la valeur du j^{e} capteur pour le i^{e} run à l'instant de mesure k.

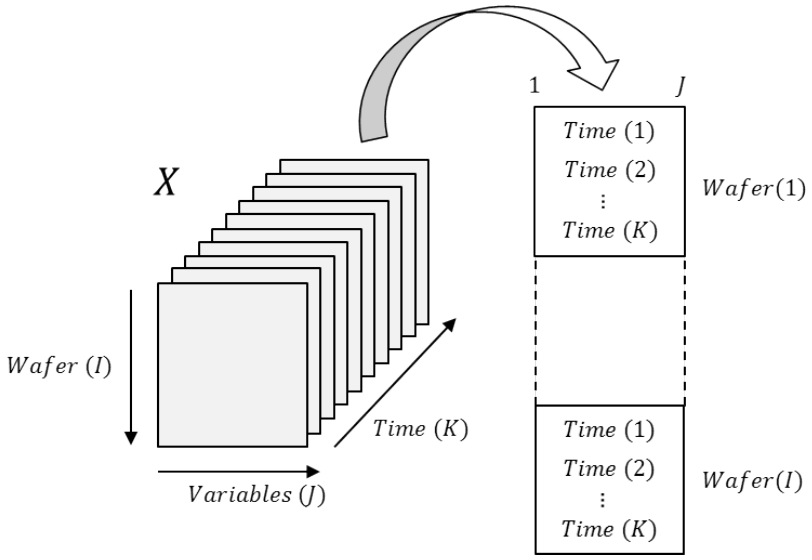

FIGURE 3.5 – Déroulement de données, voir [71].

Dès lors, une variable dans la matrice après déroulement est la combinaison de toutes les valeurs prises par un capteur donné, indépendamment de l'instant de mesure associé. Ce déroulement a été proposé dans [80].

On note X_k la matrice de dimensions $I \times J$ contenant les données collectées au temps k, avec toujours X la matrice complète de dimensions $I \times JK$.

On fait l'hypothèse suivante dans la Section 3.4.1 : $X \sim \sum_{k=1}^{K} \frac{1}{K} N(m_k, S_k)\delta_k$, avec ainsi une loi normale multi-variée à chaque temps du processus. Sous cette hypothèse, la matrice de covariance est la suivante, en utilisant la loi de la variation totale :

$$var(X) = \mathbb{E}(var(X|k)) + var(\mathbb{E}(X|k)) \tag{3.2}$$

$$= \frac{1}{K} \sum_{k=1}^{K} S_k + var(M), \tag{3.3}$$

où

$$M = \begin{bmatrix} m_1 \\ \vdots \\ m_K \end{bmatrix} = \begin{bmatrix} m_{11} \dots m_{1J} \\ \vdots \\ m_{K1} \dots m_{KJ} \end{bmatrix}, \tag{3.4}$$

en utilisant la notation $m_k = (m_{k1}, ..., m_{kJ})$. Pour k un temps et j un capteur, m_{kj} est la moyenne sur tous les runs au temps k sur le capteur j.

Il s'avère dans la pratique que cette matrice de covariance ne dépend quasiment que des moyennes des variables : l'influence des matrices de covariance à chaque temps est très réduite. En effet, en termes d'amplitudes, la variation déterministe du procédé, c'est-à-dire les écarts d'amplitude attendus entre les différentes étapes, est très largement supérieure à la variation aléatoire qu'on peut observer en moyenne à un instant donné. Si ce n'était pas le cas, alors les différentes étapes ne pourraient pas être garanties distinctes, ce qui mettrait en péril la réalisation du procédé. Si on reconsidère les données de l'exemple d'illustration pris à la Section 3.4.1, la norme de Frobenius de $var(M)$ est 335 fois plus élevée que celle de $\frac{1}{K}\sum_{k=1}^{K} S_k$. Dès lors, l'ACP multi-way appliquée ainsi revient à rechercher une projection qui ordonne les axes suivant la variation globale du procédé, qui est déterministe car relative aux moyennes qui décrivent le déroulement des recettes, et non pas relativement aux variations aléatoires qui sont celles que l'on recherche dans l'espace résiduel de l'ACP classique.

La matrice de corrélation est obtenue comme $\Sigma = Dvar(X)D$, où la matrice D est définie comme :

$$D = diag(var(X))^{-\frac{1}{2}} = \begin{bmatrix} \frac{1}{\sigma_1} & & \\ & \ddots & \\ & & \frac{1}{\sigma_J} \end{bmatrix}, \tag{3.5}$$

où σ_j est l'écart-type du j^{e} capteur.

3.4.3 Application de l'Analyse en Composantes Principales : ACP multi-way

Dans cette section, on applique l'Analyse en Composantes Principales [26] à la matrice déroulée.

Comme décrit dans le Chapitre 2, à la Section 2.3.3, l'objectif de l'ACP est de déterminer les matrices T et P satisfaisant $X = TP^T$ et $T = XP$, avec les contraintes suivantes : P est une matrice de changement de base, calculée de telle façon que la première colonne de T maximise la variance parmi toutes les projections uni-dimensionnelles possibles de X. Les autres colonnes sont calculées récursivement : elles maximisent la même variance, mais basée sur la matrice X à laquelle on a soustrait les données déjà projetées. La matrice T est la réécriture de X dans la base fournie par P.

La matrice $P \in \mathbb{R}^{J \times J}$ est la matrice des vecteurs propres de la matrice de corrélation de X. Cependant, pour éviter une influence trop grande des

variables dont la variance est élevée, nous employons une métrique pour la diagonalisation différente de la métrique euclidienne standard, permettant de pondérer chaque colonne par son écart-type. Cette métrique est définie par la matrice D employée dans la Section 3.4.2. Il s'agit d'une démarche classique dans le cadre de l'application de l'ACP [63].

Pour déterminer P, on effectue alors la diagonalisation de $\Sigma = DX^TXD \in \mathbb{R}^{J\times J}$, et la matrice des composantes obtenue est $T = XP = (t_{i,j,k}) \in \mathbb{R}^{KI\times J}$.

Dans la pratique, l'ACP est généralement utilisée pour trouver des modèles linéaires de dimension inférieure à celle des données initiales, [30]. La démarche que nous proposons ici est de l'utiliser uniquement dans le but de trouver une base diagonalisant Σ avec des valeurs propres ordonnées. En particulier, cela signifie que chaque paire de composantes est orthogonale (voir [30]). On ne décompose pas le modèle entre composantes principales et résiduelles, en s'appuyant sur l'argument donné à la Section 3.4.2. Il s'agit pour nous alors de proposer une diagonalisation, avec une base unique, des matrices de corrélation déterminées à chaque temps. La réduction de dimensions se fait ainsi en supposant que les matrices S_k, définies à la Section 3.4.1, sont toutes diagonales par rapport à la même base.

3.4.4 Reconstruction de la dépendance temporelle

En déroulant les données, le lien temporel qui les unissait est essentiellement perdu, puisque la modélisation faite est une diagonalisation globale des covariances des données indépendamment du temps. Nous reconstruisons ce lien une fois la projection des données sur les composantes effectuée. Soit k un instant de mesure. On projette $(x_{i,j,k})_{i,j} \in \mathbb{R}^{I\times J}$, la sous-matrice de X contenant toutes les données liées au k^{e} instant, en utilisant la matrice de projection P donnée par l'ACP. Les variables aléatoires $(t_{i,j,k})_i$ avec i variant et k et j fixés sont modélisés par une distribution normale dont la moyenne et l'écart-type sont notés μ_{jk} et σ_{jk}, respectivement. Pour chaque couple (j,k) ces deux paramètres sont appris sur l'ensemble d'apprentissage, utilisant la moyenne et l'écart-type empiriques comme estimateurs. On a ainsi :

$$(t_{i,j,k})_i \sim N(\mu_{jk}, \sigma_{jk}). \tag{3.6}$$

Gérer les écarts-types nuls : Sur les steps où les conditions de réalisation sont régulées, les données observées pour certains capteurs sont très répétables. Ainsi, lorsque la précision de mesure de ces capteurs est

faible comparativement à la précision de la régulation, alors il est possible de n'observer qu'une seule et unique valeur pour ce capteur à un instant donné du procédé, et ce pour tout l'ensemble d'apprentissage. Il en va de même par exemple pour le flux d'un gaz qui n'a pas encore été introduit. Ce phénomène fait que les écarts-types appris pour les composantes peuvent dans certains cas être extrêmement faibles, et engendrer des problèmes de fausses alarmes seulement dus à la sensibilité du capteur.

Ce que nous proposons ici est d'évaluer la sensibilité de chacun des capteurs, afin de proposer un écart-type minimal se substituant à certaines valeurs trop faibles observées. En effet, si un écart à la valeur nominale observée est situé au même niveau que la sensibilité du capteur, alors aucune conclusion ne peut en être tirée.

Si l'information de sensibilité des capteurs est connue, elle peut être entrée comme paramètre du modèle. Dans notre étude, ce n'est pas le cas. Pour évaluer la sensibilité des capteurs, on utilise l'ensemble d'apprentissage initial de I runs.

On se place dans le cas d'un capteur fixé j'. On note $a_{j'}$ la valeur de précision de ce capteur. Si on observe une mesure x, alors on estime que cette mesure est un tirage issu de la loi uniforme sur l'ensemble $[x - a_{j'}, x + a_{j'}]$. Pour évaluer $a_{j'}$, on construit l'ensemble $A_{j'} = \{x$ tel que x soit une mesure mesure observée sur au moins un des I runs$\}$. On définit alors $\tilde{a}_{j'}$, estimation de $a_{j'}$ comme étant :

$$\tilde{a}_{j'} = min\{\ x - y \mid (x, y \in A_{j'}) \text{ et } (x \neq y)\}. \tag{3.7}$$

L'estimation $\tilde{a}_{j'}$ est ainsi la différence entre les deux valeurs les plus proches observées sur l'ensemble du traitement des I wafers. La modélisation avec la loi uniforme nous permet de déduire l'écart-type de mesure $s^m_{j'}$ associé au capteur :

$$s^m_{j'} = \frac{a_{j'}}{\sqrt{3}}. \tag{3.8}$$

En prenant la notation $P = (p_{jj'})_{j,j' \leq J}$ pour la matrice de projection, pour chaque composante j on peut écrire un écart-type minimal σ^m_j :

$$\sigma^m_j = \sum_{j'=1}^{J} |p_{jj'}| s^m_{j'}, \tag{3.9}$$

correspondant ainsi au pire cas de projection de la variabilité induite par la précision des capteurs.

L'écart-type de mesure est ainsi l'écart-type instantané par défaut pour chacune des composantes. Si jamais un écart-type calculé sur l'ensemble

d'apprentissage pour un instant donné est inférieur à l'écart-type de mesure de la composante, alors c'est ce dernier qui est employé comme écart-type dans la modélisation. Il s'agit d'une situation fréquente, comme évoqué à la Section 3.4.1 : cette vérification est donc employée de façon systématique pour les calculs d'écarts-types dans la modélisation proposée.

3.4.5 Gaussian Time Error : un indice de détection pour chaque run

La démarche que nous proposons est de simplifier les dimensions de modélisation des données. L'espace très complexe où chaque temps est modélisé par un vecteur de moyennes et une matrice de corrélation quelconque, comme décrit à la Section 3.4.1, devient l'espace décrit à la Section 3.4.4. Dans celui-ci, chaque temps est modélisé par un vecteur de moyennes et un vecteur d'écarts-types après diagonalisation de la matrice de corrélation globale. Mesurer l'écart à la distribution multivariée complexe initiale, devient alors plus simple sous cette modélisation : on peut mener des tests indépendants sur chaque dimension de chaque temps. Nous proposons ici un indice de détection, que nous nommons Gaussian Time Error (GTE), en raison du fait qu'il décrit les écarts à des distributions gaussiennes au cours du temps du procédé. Celui-ci est un score entier pour chaque run, qui décrit le nombre de temps de mesure où le run est vu comme atypique.

Dans le cadre de notre étude, nous proposons une définition d'un processus sans défaut :

Définition 1. *Un processus sans défaut est un processus qui présente en chaque instant des valeurs acceptables vis-à-vis des lois normales estimées à partir de l'ensemble d'apprentissage.*

On peut faire une évaluation de la position relative d'une observation par rapport à une distribution mono-variée donnée en utilisant la notion de p-valeur [53]. Pour n'importe quelle distribution initiale, les p-valeurs sont uniformément distribuées entre 0 et 1.

Définition 2. *La p-valeur se définit comme la probabilité d'obtenir une valeur au moins aussi extrême que la valeur observée.*

La notion de valeur extrême peut être bilatérale, ou bien unilatérale dans le cas où seules les valeurs trop grandes (respectivement trop petites) sont considérées comme extrêmes. Nous prenons ici la définition bilatérale de la p-valeur. On définit alors pour chaque $(t_{i,j,k})$ une p-valeur $pval_{i,j,k}$

correspondante, où i, j, et k sont respectivement un run, une composante et un instant de mesure. On a :

$$pval_{ijk} = 2 \cdot \Phi\left(-\left|\frac{t_{i,j,k} - \mu_{jk}}{\sigma_{jk}}\right|\right), \tag{3.10}$$

où μ_{jk} et σ_{jk} sont la moyenne et l'écart-type de la distribution étudiée (j^{e} composante) pour l'instant k (voir Section 3.4.4), et Φ est la fonction de répartition de la loi normale centrée et réduite. On rappelle la définition de Φ : $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \mathrm{e}^{-\frac{1}{2}t^2} \mathrm{d}t$, pour tout $x \in \mathbb{R}$.

Pour tout run, on a ainsi une matrice de p-valeurs, dont les dimensions sont identiques à celles des données initiales : $K \times J$. Chaque p-valeur est liée au test bilatéral déterminant si une valeur donnée est tirée de la distribution normale étudiée, avec un paramètre d'erreur de Type-I α_d à fixer. Lorsqu'une p-valeur est plus petite que α_d, l'hypothèse selon laquelle l'observation est issue de la distribution est rejetée. On définit donc une observation $t_{i,j,k}$ comme acceptable lorsque $pval_{i,j,k} \geq \alpha_d$.

L'indice de détection Gaussian Time Error

En s'appuyant sur les p-valeurs décrites, on peut déterminer un grand nombre d'indices de détection différents, tenant compte du rejet ou non des tests statistiques associés. L'indice Gaussian Time Error (GTE) que nous proposons, consiste à compter le nombre d'instants où au moins une p-valeur est invalide. Ainsi toutes les composantes ont le même poids dans la détection, et un écart à la loi normale est toujours compté avec le même poids, que celui-ci soit juste au-dessus du critère (dépendant de α_d) ou très éloigné de celui-ci.

Cet indice, que l'on note GTE_i pour la i^e plaque, vaut :

$$\mathrm{GTE}_i = \#(\{k \mid \exists j,\ pval_{ijk} < \alpha\}), \tag{3.11}$$

où $\#$ désigne le cardinal de l'ensemble.

Cet indice est pertinent sous les hypothèses suivantes, valables dans le cas d'un équipement en fonctionnement normal :

Hypothèses pour le GTE

(H1) Les composantes données par l'ACP multi-way suivent une loi normale pour chaque instant de mesure individuel ;

(H2) A chaque instant, l'évènement "avoir une composante dont la p-valeur est inférieure à α_d" est indépendant du fait d'avoir le même évènement sur une autre composante ;

(H3) Avoir une composante dont la p-valeur est inférieure à α_d à un instant donné est indépendant du fait d'avoir le même évènement à un autre instant de mesure.

On détermine une limite pour l'indice GTE, en déterminant une loi de probabilité décrivant son comportement dans le cas d'un processus en fonctionnement normal. On fixe un instant k et un run i. Avec $A_{i,j,k}$ l'évènement "$pval_{ijk} < \alpha_d$", on a :

$$\mathbb{1}_{A_{i,j,k}} \sim B(1, \alpha_d), \tag{3.12}$$

où $B(n, p)$ est la loi binomiale de paramètres n et p, et $\mathbb{1}_{A_{i,j,k}}$ la variable aléatoire prenant la valeur 1 si $A_{i,j,k}$ est réalisé et 0 sinon. Sous l'hypothèse (H2), le nombre de composantes ayant une p-valeur dépassant la limite suit une loi binomiale :

$$\sum_j \mathbb{1}_{A_{i,j,k}} \sim B(J, \alpha_d). \tag{3.13}$$

Ainsi, la probabilité d'avoir au moins une composante au delà de la limite correspond à l'approximation suivante :

$$\mathbb{P}\left(\bigcup_j A_{i,j,k}\right) \simeq J\alpha_d. \tag{3.14}$$

Puisque $J\alpha_d$ est négligeable devant 1, alors :

$$\mathbb{P}(B(J, \alpha_d) \geq 1) = 1 - (1 - \alpha_d)^J \simeq J\alpha_d.$$

Enfin, du fait de l'hypothèse (H3), on déduit que :

$$\mathrm{GTE} = \#\left(\{k | \bigcup_j A_{i,j,k}\}\right) \sim B(K, J\alpha_d), \tag{3.15}$$

On peut fixer une limite suivant les quantiles de la loi binomiale, en prenant le plus petit $k \in \{1, ..., K\}$ tel que :

$$\mathbb{P}(B(K, J\alpha_d) \geq k) \leq \alpha'_d, \tag{3.16}$$

où α'_d est l'erreur de Type-I choisie. On a ainsi deux paramètres à fixer pour utiliser le GTE : l'erreur de Type-I des tests individuels α_d à laquelle les p-valeurs sont comparées, et l'erreur de Type-I α'_d qui définit la limite du GTE.

La valeur de la limite dépend des deux paramètres : elle est proportionnelle à α_d, et inversement proportionnelle à α'_d. L'erreur de Type-I du GTE est α'_d, alors que α_d n'intervient que dans le calcul de l'erreur de Type-II.

Erreur de Type-II du Gaussian Time Error

Soit W la matrice des données alignées (de dimensions $J \times K$) d'un run illustrant une faute de l'équipement. On détermine l'erreur de Type-II β'_d que ces données soient classifiées comme correctes alors qu'elles ne le sont pas, sachant les deux paramètres du GTE α_d, erreur de Type-I des tests faits à chaque instant sur chaque composante, et α'_d, erreur de Type-I du GTE. On note $F \subset \{1, ..., K\}$ l'ensemble des temps où le système est en faute, de taille L. Du point de vue du test statistique décrit Section 3.4.1, le système est en faute lorsqu'il existe au moins un temps k pour lequel la distribution dont les données sont tirées ne correspond pas à la loi normale $N(m_k, S_k)$ attendue (en reprenant les mêmes notations).

La matrice W est d'abord projetée dans l'ensemble des composantes, pour obtenir $Z = WP = (z_{jk})_{j,k}$. Pour j et k donnés, on suppose que z_{jk} suit la loi suivante :

— si $k \notin F$, $N(\mu_{jk}, \sigma_{jk})$;
— si $k \in F$, $N(m_{jk}, s_{jk})$, où $m_{jk} \neq \mu_{jk}$ ou $s_{jk} \neq \sigma_{jk}$.

Cela signifie que nous supposons que **si le système est en défaut au temps k, alors la faute est visible sur toutes les composantes, à travers la moyenne ou la variance**. En effet, si la distribution $N(m_k, S_k)$ n'est pas à l'origine des données observées, alors soit m_k a changé, et la répercussion sur les moyennes des composantes est immédiate par combinaison linéaire, soit S_k a changé. La variance de chaque composante dépend de toutes les variances et toutes les covariances des variables initiales. En effet, on n'observe pas de variables strictement indépendantes dans la pratique, celles-ci étant toutes corrélées à la variable implicite décrivant le déroulement du procédé. La variance de chaque composante dépend ainsi de chacune des variables : il y a alors un changement des écarts-types observés sur les composantes peu importe les variables sur lesquelles la faute se manifeste.

Tout d'abord, on détermine l'erreur de Type-II $\beta_d(j, k)$, qui est l'erreur de Type-II du test individuel fait à chaque temps pour chaque composante. Si $k \notin F$, alors $\forall j, \beta_d(j, k) = 0$, puisque l'hypothèse alternative du test est fausse. On considère $k \in F$ et une composante j. On fixe $\alpha_d < 0.5$, ainsi $\Phi^{-1}(\alpha_d) < 0$, et on note ϕ_α la valeur $\Phi^{-1}(\frac{\alpha_d}{2})$ (on rappelle que Φ est la fonction de répartition de la loi normale centrée et réduite).

$$\begin{aligned}
\beta_d(j,k) &= \mathbb{P}\left(pval_{ijk} > \alpha_d | t_{ijk} \sim N(m_{jk}, s_{jk})\right) \\
&= \mathbb{P}\left(\frac{t_{ijk}-\mu_{jk}}{\sigma_{jk}} \in [\phi_\alpha, -\phi_\alpha] | t_{ijk} \sim N(m_{jk}, s_{jk}))\right) \\
&= \mathbb{P}\left(\frac{s_{jk}}{\sigma_{jk}} \frac{t_{ijk}-m_{jk}+m_{jk}-\mu_{jk}}{s_{jk}} \in [\phi_\alpha, -\phi_\alpha] | t_{ijk} \sim N(m_{jk}, s_{jk}))\right) \\
&= \mathbb{P}\left(\frac{t_{ijk}-m_{jk}}{s_{jk}} \in \left[\frac{\sigma_{jk}\phi_\alpha + \mu_{jk} - m_{jk}}{s_{jk}}, \frac{-\sigma_{jk}\phi_\alpha + \mu_{jk} - m_{jk}}{s_{jk}}\right] | t_{ijk} \sim N(m_{jk}, s_{jk}))\right) \\
&= \Phi\left(\frac{-\sigma_{jk}\phi_\alpha + \mu_{jk} - m_{jk}}{s_{jk}}\right) - \Phi\left(\frac{\sigma_{jk}\phi_\alpha + \mu_{jk} - m_{jk}}{s_{jk}}\right),
\end{aligned}$$

puisque $\frac{t_{ijk}-m_{jk}}{s_{jk}} \sim N(0,1)$.

L'erreur de Type-II du GTE β'_d est la probabilité que $GTE(W)$ soit inférieur à la limite l alors que les données sont en faute. Le GTE est inférieur à la limite quand un nombre de temps de faute inférieur à la limite est détecté, et qu'il n'y a pas assez d'autres temps de faute détectés par erreur pour que la limite soit franchie. Cette situation s'exprime différemment selon que la taille de la faute soit supérieure ou non à la limite l. Pour un entier n donné, on note $A(n)$ l'ensemble des arrangements de F de taille n.

$$\begin{aligned}
\beta'_d &= \mathbb{P}\left(GTE(W) <= l\right) \\
&= \mathbb{P}\left(L \leq l, \text{"}L_d \leq L \text{ non détectés", "nombre de faux positifs} \leq l - L + L_d\text{"}\right) + \\
&\mathbb{P}\left(L > l, \text{"au moins } L - l + n \text{ non détectés, } n \geq 0\text{", "nombre de faux positifs} \leq n\text{"}\right) \\
&= \mathbb{1}_{L \leq l} \sum_{L_d=0}^{L} \left(\left(\sum_{a \in A(L_d)} \prod_{k \in a} \prod_{j=1}^{J} \beta_d(j,k) \right) \times \mathbb{P}(B(K-L, J\alpha_d) \leq l - L + L_d) \right) + \\
&\quad \mathbb{1}_{L > l} \sum_{n=0}^{l} \left(\left(\sum_{a \in A(L-l+n)} \prod_{k \in a} \prod_{j=1}^{J} \beta_d(j,k) \right) \times \mathbb{P}(B(K-L, J\alpha_d) \leq n) \right) \quad (3.17)
\end{aligned}$$

3.4.6 Filtrage des fausses alarmes

Le GTE a une erreur de Type-I α'_d : cela signifie donc qu'un équipement en condition normale produira un run hors des limites du GTE en moyenne tous les $\frac{1}{\alpha'_d}$ runs. L'utilisation en temps réel sur les équipements de plusieurs ateliers risque alors de produire un très grand nombre de fausses alarmes relativement aux vraies détections de fautes.

Prenons un exemple d'équipement typique : l'exécution de la recette de production dure environ 2 minutes, et la production se fait quasiment sans discontinuer entre deux maintenances, qui est le seul arrêt de la production, d'environ un jour complet tous les mois. Ainsi, on peut compter

environ 600 runs par jour soit 18000 runs par mois. Cet équipement appartient à un atelier de production comportant 9 autres équipements similaires, pour un total d'environ 180000 runs par mois. Si on veut réduire le nombre de fausses alarmes à une seule par mois sur cet atelier, on s'expose à une incertitude importante en choisissant un α'_d très faible, car les seules fautes détectables sont alors celles dont l'amplitude est extrêmement élevée. En gardant une erreur de Type-I plus élevée, on s'expose, dans l'environnement industriel, au risque d'avoir une grande majorité de fausses alarmes parmi les détections, ce qui peut créer un risque de baisse de l'importance attribuée au traitement des alarmes par le personnel de production. Il s'agit d'un problème déjà observé avec l'utilisation des cartes de contrôle.

Nous proposons ainsi de filtrer les runs dont le GTE dépasse la limite, plutôt que d'abaisser de façon trop drastique le paramètre α'_d. Nous revenons alors au concept de variations de causes assignables, traité dans la Section 1.2.1. Si un seul run correspond à un GTE au-dessus de la limite, il n'y a pas nécessairement de conclusion à tirer, et une telle variation est supposée aléatoire. Cependant, la répétition à l'identique des observations ayant amené le dépassement de la limite ne constitue pas un phénomène aléatoire, mais bien un phénomène déterministe dû à un changement. Les runs observés ne correspondent pas aux runs d'apprentissage du fait d'une cause assignable.

Nous proposons un filtre dont le fonctionnement est le suivant : chaque nouvelle alarme est filtrée relativement aux $n-1$ précedentes. Soit une p-valeur inférieure à α_d associée à la dernière alarme. Si parmi les n dernières alarmes, m présentent une p-valeur faible correspondant au même temps et à la même composante, alors cette p-valeur faible est conservée pour le calcul du GTE filtré. Sinon, elle est supprimée. L'algorithme de filtrage se fait en 3 étapes :

1. On considère un run i fixé dont on souhaite filtrer le GTE. Si GTE_i est inférieur à la limite, alors on ne fait rien. Si GTE_i est supérieur à la limite, tout d'abord, la matrice des p-valeurs est transformée en une matrice constituée de 0 et de 1 : pour la composante j et le temps k, on note d_{ijk} l'évènement $d_{ijk} = \{pval_{ijk} < \alpha\}$. On remplace alors chaque p-valeur par la valeur $\mathbb{1}_{d_{ijk}}$ correspondante. On note $D_i = (D_{ijk})_{jk}$ la matrice correspondant au run i.
2. On note A l'ensemble des n dernières alarmes, incluant i. On note D_A la matrice de terme général $(\max(\sum_{a=1}^{n} D_{ajk} - m, 0))_{jk}$. On remplace D_i par sa version filtrée D_i^f qui est $D_i^f = D_A.D_i$ où . désigne le produit matriciel de Hadamard, c'est-à-dire le produit terme à terme des deux

matrices.

3. On calcule le GTE filtré de i, noté GTE_i^f, en se basant sur les valeurs de D_i^f : $\text{GTE}_i^f = \sum_{k=1}^{K} \max_j(D_{ijk}^f)$.

On note qu'il est alors nécessaire d'attendre m alarmes au minimum pour obtenir une vraie alarme. Les premières alarmes pour un modèle donné sont ainsi toujours entièrement filtrées. L'ensemble A peut être réinitialisé après une faute détectée et réparée.

Le choix des paramètres n et m se fait sur la base des rendements d'un atelier de production. La valeur de n décrit la capacité d'oubli de l'algorithme, alors que la valeur m donne le nombre d'alarmes sur le GTE non filtré similaires nécessaires pour déclencher une véritable alarme. Si les runs sont courts et traitent peu de wafers en même temps, un filtre fort assure un minimum de fausses alarmes et un risque de pertes relativement faible en cas de défaut de l'équipement. Par exemple, pour les recettes de dépôt de matériaux, correspondant à un seul wafer traité pendant environ 2 minutes, les valeurs de $n = 5$ et $m = 3$ donnent de très bons résultats selon notre expérience (voir Section 3.6). Pour des runs correspondant à plus de wafers, par exemple pour des équipements de diffusion dont les recettes durent plusieurs heures et traitent une centaine de wafers à la fois, un filtrage moindre est recommandé, car le coût économique de filtrer une alarme à tort risque d'être élevé. Dans ce cas, il faut choisir un filtre avec la valeur $m = 2$, en réduisant la valeur d'oubli du filtrage, en prenant par exemple $n = 10$.

3.4.7 Localisation statistique des fautes

Lorsque le GTE amène à la détection d'une faute associée à un run, l'objectif est alors de pouvoir apporter une aide à la réparation de l'équipement. Cela se fait en identifiant ce qui dans les caractéristiques des données du run a entraîné la détection. En effet, nous avons exclu, dans le Chapitre 2, la possibilité d'utiliser la connaissance experte comme base d'identification des fautes, du fait de leur multiplicité. Recenser précisément et exhaustivement les conséquences sur les données FDC de toutes les fautes possibles de l'équipement pour chaque recette n'est pas concevable. De même, se baser sur l'historique des fautes l'est encore moins puisque, même en considérant plusieurs années de production, on n'observe qu'un échantillon très réduit de toutes les fautes possibles. L'identification réelle de la faute se fait ainsi par la connaissance d'un expert, une fois la faute localisée dans les données, ce que nous nommons "localisation statistique" de la faute.

En suivant la démarche permettant de calculer le GTE, les temps du procédé correspondant à la faute ont déjà été identifiés : ce sont les temps pour lesquels au moins une p-valeur est inférieure à α_d. Cela résout ainsi une part importante du problème, puisque tous les autres temps sont ignorés à partir de là.

Soit i un run correspondant à un GTE hors-limite, et k un instant pour lequel une p-valeur au moins, associée à une composante, est inférieure à α_d. On effectue un retour sur les capteurs initiaux. On note x_{ijk} la valeur du j^{e} capteur à l'instant k, et m_{jk} et s_{jk} la moyenne et l'écart-type respectivement pour les runs de l'ensemble d'apprentissage pour le capteur j à l'instant k. On définit le z-score Z_{ijk} et la contribution C_{ijk} associés à x_{ijk} :

$$Z_{ijk} = \frac{x_{ijk} - m_{jk}}{s_{jk}}, \qquad C_{ijk} = 100 \frac{Z_{ijk}}{\sum_{a=1}^{J} Z_{iak}}.$$

Pour chaque wafer détecté et chaque temps associé à la faute, la plus haute valeur de C_{ijk} correspond à la variable j sur laquelle la faute est la plus visible. Ces formules signifient que la variable la plus responsable est celle qui s'éloigne le plus de la distribution observée parmi les runs d'apprentissage.

Pour l'application dans un cas industriel de la méthode, les temps relevés pour la faute sont ramenés à la notion de step de la recette en utilisant la trajectoire de référence d'alignement : on fait correspondre l'avancée des steps dans le temps pour cette trajectoire avec la localisation de la faute. L'information en termes de steps, qui est fixe pour une recette donnée, est plus importante pour les experts que l'information de temps, qui est variable pour chaque run.

Des exemples de telles identifications sont présentés à la Section 3.6.

Remarque. *Il n'est pas nécessaire de stocker les moyennes m_{jk} des capteurs initiaux, car elles peuvent être calculées directement à partir de la matrice de projection P et des moyennes des composantes, l'espérance étant une application linéaire. En revanche, il faut stocker leurs écart-types, car il n'est pas possible de les obtenir simplement à partir des écarts-types des composantes.*

3.5 La problématique de la maintenance

Nous traitons de la façon dont notre méthode s'adapte aux changements observés dans les données du fait de la maintenance des équipements. Il s'agit là d'un problème majeur pour la détection statistique de fautes dans

l'industrie du semi-conducteur. Pour s'assurer d'avoir un ensemble d'apprentissage correct, il est nécessaire de prendre des données de runs ayant eu lieu plusieurs semaines ou mois dans le passé, et pour rentabiliser ce travail de sélection, il faut conserver le modèle créé le plus longtemps possible. Étant donnée la fréquence des maintenances, cela signifie qu'un modèle donné est en général confronté à une première maintenance immédiatement après création, celle entre les données d'apprentissage et le présent, avant d'être confronté à une nouvelle maintenance à peu près mensuellement. En effet, si l'ensemble d'apprentissage est pris, par exemple, deux mois dans le passé, il y a eu une maintenance postérieure à la période d'apprentissage depuis. La problématique est double : savoir se prononcer, sur la base du modèle appris, quant à la présence d'une faute induite par la maintenance (le plus souvent une erreur humaine lors de l'intervention), et ensuite savoir adapter les paramètres du modèle, dans le cas où le premier test n'a rien détecté, afin de pouvoir reprendre la détection de fautes en temps réel de façon automatisée.

3.5.1 Impact des maintenances sur les données récoltées

L'expérience industrielle à STMicroelectronics Rousset montre que les valeurs attendues des différents capteurs au cours du procédé sont susceptibles d'être modifiées par une intervention de maintenance. Certains paramètres sont réajustés par les techniciens, et d'autres varient naturellement du fait du remplacement de certaines pièces de l'équipement. De telles variations ne sont pas caractéristiques de fautes sur l'équipement. La Figure 3.6 montre l'exemple d'une variable avant (en rouge) et après (en bleu) une maintenance, pour plusieurs runs superposés. On observe deux différences importantes, qui donneraient des fausses alarmes systématiques si le modèle GTE initial était conservé, avec ou sans filtrage des alarmes puisque la différence est répétée à chacun des runs. Il s'agit d'une différence de cause assignable : les valeurs observées avant la maintenance ne doivent plus être répétées, et une partie des paramètres du modèle, concernant les moyennes, est devenue obsolète.

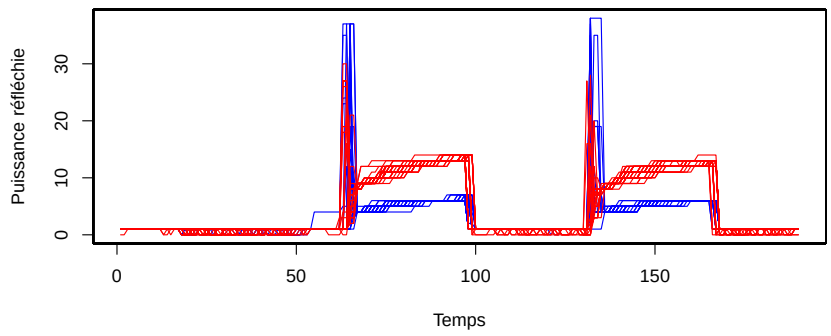

FIGURE 3.6 – Modification significative d'une variable suite à une maintenance.

Puisque nous n'avons plus de certitude concernant les valeurs moyennes dans le modèle, nous ne proposons pas de test les concernant juste après une maintenance. L'ensemble d'apprentissage initial ne permet pas d'élaborer de test pertinent sur ces valeurs : nous proposons une solution faisant intervenir la comparaison entre plusieurs équipements dans le Chapitre 4. En revanche, la variabilité d'un wafer à l'autre reste une donnée pertinente. Celle-ci doit rester de l'ordre de celle observée dans l'ensemble d'apprentissage. L'expertise sur les équipements montre qu'une augmentation de la variabilité indique une erreur liée à une maintenance fautive. Pour cette raison, en utilisant un échantillon des premiers runs effectués après la maintenance, on teste si la variabilité n'a pas significativement augmenté, et ensuite, si c'est le cas, on ajuste les paramètres de moyenne dans le modèle, et seulement eux. La matrice de projection dans l'espace des composantes est conservée, car elle traduit les corrélations physiques globales entre les capteurs, et les paramètres liés à la variabilité des composantes le sont également, car ils correspondent au large ensemble d'apprentissage choisi initialement, ce qui leur confère plus de robustesse. Un schéma de cette méthodologie est présenté sur la Figure 3.7.

Des exemples de tests de variabilité sur les maintenances sont présentés à la Section 3.6.

3.5.2 Test de variance pour les maintenances

On vérifie d'abord que la maintenance n'a pas engendré de faute, en testant les variances empiriques de l'ensemble de mise à jour contre les variances apprises dans le modèle initial. On considère la j^{e} composante au temps k. On rappelle que σ_{jk} désigne son écart-type dans le modèle GTE. L'écart-type empirique, calculé sur l'ensemble de mise à jour de taille

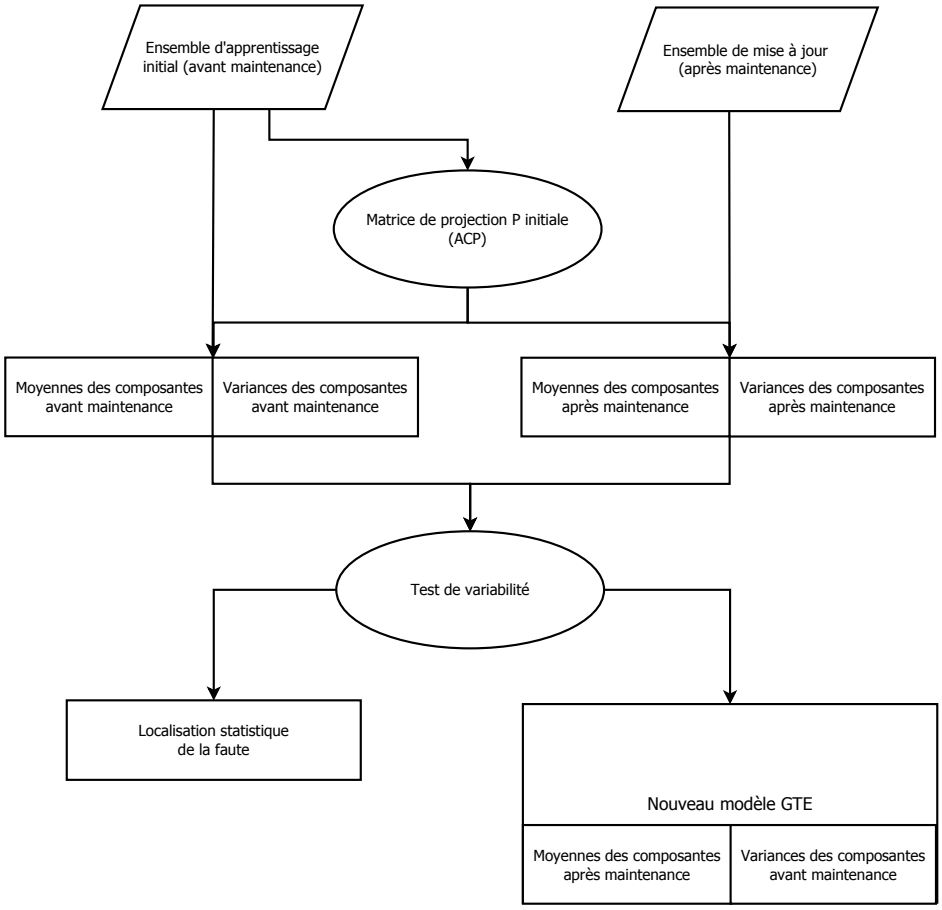

FIGURE 3.7 – L'algorithme de gestion des interventions de maintenance pour les modèles GTE.

n, est noté σ'_{jk}. Sous l'hypothèse que les n nouvelles observations sont tirées d'une loi normale d'écart-type σ_{jk}, on a la relation suivante, en notant χ^2_{n-1} la loi du χ^2 à $n-1$ degrés de liberté :

$$\frac{(n-1)\sigma'^2_{jk}}{\sigma^2_{jk}} \sim \chi^2_{n-1}.$$

Dès lors, on sélectionne la limite l relativement au quantile du χ^2 associés. On choisit une erreur de Type-I α_m. La limite $l \in \mathbb{R}_+$ est le plus petit nombre vérifiant :

$$\mathbb{P}(\chi^2_{n-1} \geq l) \leq \alpha_m. \tag{3.18}$$

On utilise un test unilatéral, puisqu'une baisse de variabilité est parfois

observée après nettoyage de l'équipement, et il s'agit de quelque chose de positif : on ne souhaite détecter que des augmentations de variabilité.

Le test de maintenance proposé se base sur les premiers runs effectués immédiatement après la maintenance. Il n'est pas rare d'observer une variabilité un peu plus élevée sur de tels échantillons, sans aucune conséquence sur la qualité de la production. Seule une augmentation de l'ordre de la variabilité a besoin d'être détectée. On prend alors pour limite $10l$ au lieu de l. Ce choix se justifie dans la pratique sur les exemples étudiés.

Comme pour le GTE, on peut montrer que le nombre de variances empiriques hors-limites dans le cas d'un fonctionnement normal suit la loi binomiale $B(K, J\alpha_m)$. De façon analogue, on détermine une limite avec la paramètre d'erreur de Type-I α'_m. Les rôles de α_m et α'_m sont les mêmes que ceux de α_d et α'_d pour la phase de détection, définis à la Section 3.4.5.

Identification de la cause en cas d'échec du test

Si le test décrit dans la Section 3.5.2 échoue, il est nécessaire de remonter aux variables physiques à l'origine de cet échec. Il n'est pas possible d'utiliser directement les contributions proposées à la Section 3.4.7. En effet, si l'on effectue une mise à jour du modèle, c'est car les moyennes instantanées des capteurs ont changé de façon imprévisible. Si l'on utilisait un modèle de calcul de contributions construit sur des moyennes relevées avant la maintenance, alors l'identification de la source du défaut détecté serait compromise, désignant comme source de défaut des variables physiques ayant évolué de façon naturelle.

La solution proposée ici est d'effectuer les mêmes tests de variance en se basant cette fois-ci directement sur les variables physiques. Le principe de cette démarche est le même que pour les contributions présentées à la Section 3.4.7. Pour chacun des instants où le test de variance sur les composantes a échoué, on utilise les valeurs des tests de variance sur les variables physiques comme indicateurs de la responsabilité de la variable : au plus la valeur du test est élevée, au plus la variable est suspectée d'être à l'origine de la faute.

Soit k un temps correspondant à un test de variance hors-limite. On effectue un retour sur les capteurs initiaux. On note s_{jk} la valeur de la variance du j^{e} capteur à l'instant k sur l'ensemble d'apprentissage initial, et s'_{jk} la variance pour le même capteur au même instant dans l'ensemble de mise à jour. On définit le score ζ_{jk} à partir du rapport des deux variances et

la contribution C_{jk} associés à ζ_{jk} :

$$\zeta_{jk} = \max\left(\frac{s'_{jk}}{s_{jk}} - 1, 0\right), \qquad C_{jk} = 100\frac{\zeta_{jk}}{\sum_{a=1}^{J}\zeta_{ak}}.$$

Seuls les capteurs dont la variabilité a augmenté sont recherchés, et ils sont donc les seuls à avoir une contribution non nulle.

3.5.3 Mise à jour des moyennes dans le modèle

Si le test décrit dans la Section 3.5.2 ne détecte aucune faute induite par la maintenance, alors on effectue la dernière étape dans la gestion de la maintenance dans un modèle GTE : les moyennes du modèle sont mises à jour. Les moyennes μ_{jk} dans (3.6) sont mises à jour en utilisant les moyennes empiriques observées sur les runs ayant servi à tester la variabilité. La matrice de projection P ainsi que les écarts-types σ_{jk} sont conservés du modèle initial. Puisque P est conservée, les moyennes des capteurs servant à la localisation des fautes dans la Section 3.4.7 sont automatiquement mises à jour (voir Remarque dans la Section 3.4.7). Si il n'y a pas de faute à identifier, aucune intervention humaine n'est nécessaire pour mettre à jour le modèle. Le modèle GTE initial est en général appris avec les données de 100 runs, mais nous n'utilisons que les données de 20 runs pour effectuer une mise à jour efficace après maintenance. La contrainte industrielle est importante dans le choix de la taille de l'ensemble de mise à jour : le nombre de runs servant à la mise à jour doit être le plus petit possible, car pendant l'exécution de ces runs, l'équipement est à risque. Nous montrons dans la Section 3.6 que 20 runs suffisent : les tests de variabilité sont effectués avec succès, et la mise à jour des moyennes n'engendre pas de fausses alarmes.

3.5.4 Estimateurs tronqués

En vu d'une utilisation industrielle, une mise à jour après maintenance doit nécessiter le moins de données possible. Il faut ainsi trouver un moyen de prendre une décision correcte avec peu de plaques à disposition pour tester les variabilités (Section 3.5.2). Avec peu de wafers, les données atypiques peuvent devenir un problème important, car une seule valeur aberrante, qui peut apparaître du fait de l'alignement dans le cas par exemple où on a plusieurs données manquantes consécutives, peut entraîner une augmentation significative des variances associées et faire échouer le test de variabilité. Pour se prémunir de ces problèmes, nous proposons d'utiliser

des estimateurs tronqués des moyennes et des variances des composantes, afin de limiter l'impact des valeurs extrêmes. Le choix que nous effectuons est de supprimer le maximum et le minimum de chaque distribution de l'ensemble de mise à jour. Ceci signifie donc que pour chaque instant et chaque composante deux valeurs sont supprimées indépendamment pour l'estimation des paramètres μ_{jk} et S_{jk} correspondants. Ceci permet de tenir compte du fait que l'apparition de valeurs aberrantes est a priori aléatoire, et ainsi une plaque qui présente une ou plusieurs valeurs aberrantes à un certain instant de la recette peut être utilisée pertinemment pour estimer les paramètres du reste de la recette.

3.6 Exemples d'application industrielle

Dans cette section, nous appliquons la méthodologie GTE à 3 exemples issus de cas de fautes sur des équipements de STMicrolectronics. Les produits n'étant testés que plusieurs semaines voire mois après leur passage dans les équipements, l'ensemble d'apprentissage dans chacun de ces cas est constitué de runs d'une période éloignée de la faute, dont la qualité est estimée certaine. Le modèle initial est basé sur cette période. D'autres runs de la même période sont utilisés pour vérifier le taux de fausses alarmes, puisqu'aucune détection de faute n'est censée avoir lieu durant la période ayant servi à l'apprentissage. Après ce premier test, la méthodologie de gestion des maintenances, décrite à la Section 3.5, est utilisée d'abord pour tester la présence d'une faute causée par les interventions sur l'équipement depuis la période d'apprentissage, et ensuite mettre à jour le modèle si aucune faute n'est détectée. On présente d'abord les résultats obtenus sans aucun filtrage des valeurs du GTE, afin de montrer la performance de l'indice GTE en lui-même ; ensuite, on traite des résultats filtrés, qui dans chacun de ces cas éliminent toutes les fausses alarmes, et ne font perdre que les deux premiers runs réellement détectés.

En plus des exemples présentés ici, cette méthode a pu être testée sur une cinquantaine d'équipements, sur environ 6 mois d'historique. Les modèles de détection sont restés valides sur toute la période étudiée.

3.6.1 Détection d'une faute apparaissant après une mise à jour

Le premier exemple traite d'une fuite dans un équipement de dépôt de matériaux. L'atelier de cet équipement est nommé "Chemical Vapor Depo-

sition" (CVD), en raison du fait que ce procédé consiste en l'introduction dans la chambre de production d'un matériau volatil qui réagit avec le wafer en se déposant sous forme de fine pellicule sur celui-ci. C'est une étape clé de la fabrication du produit final, et elle est menée plusieurs dizaines de fois dans les différentes routes de production.

Après le pré-traitement (voir Section 3.3), les données ont la structure suivante : $K = 115$ temps de mesure associés à $J = 15$ capteurs. Les capteurs collectent les données de flux de gaz, de puissances électriques, de pressions et de températures. Les deux paramètres du GTE, α_d et α'_d, sont tous deux fixés à 0.1%. On considère deux périodes où sont collectées les données :

— La première période, d'apprentissage, qui contient 200 runs ;

— La seconde période, de 383 runs, après une maintenance.

Le modèle GTE initial (voir Section 3.4) est appris sur un ensemble de 100 runs choisis aléatoirement dans la première période. Les 100 runs restants sont testés pour vérifier le taux de fausses alarmes. La procédure de gestion des maintenances est alors appliquée sur les 20 premiers runs de la période suivante. On vérifie qu'aucune faute n'est présente, à partir du test de variance, avec les paramètres α_m et α'_m tous deux fixés à 0.1% (voir Section 3.5.2). Dans ce cas, la maintenance n'a pas été classifiée comme fautive, et donc les moyennes du modèle sont mises à jour. Les runs restants de la seconde période sont testés à partir du modèle mis à jour : le précédent est obsolète et ne sera plus réutilisé. La procédure de détection est la même qu'avant la mise à jour.

Détection de la faute

Les résultats complets sont représentés sur la Figure 3.8, qui présente l'indice GTE pour tous les runs. La ligne horizontale représente la limite du GTE, calculée de façon théorique à partir de la loi binomiale (voir (3.16)). La première ligne verticale délimite l'instant de la maintenance. La deuxième ligne verticale délimite le premier run affecté par une faute de l'équipement. Cette date d'apparition a été confirmée par les tests finaux sur les wafers. Le Tableau 3.1 résume les performances du GTE sur cet exemple.

Dans la première période, les runs qui n'ont pas été utilisés pour construire le modèle sont testés. Aucune faute n'est censée être vue, et on obtient un taux de fausses alarmes de 3%, dû à des problèmes de collecte sur les 3 runs détectés. Ce taux est 30 fois supérieur à l'erreur de Type-I α'_d fixée, mais ce paramètre ne tient pas compte de la collecte de données : l'hypothèse n'est vérifiée que si la collecte est parfaite, ou si le pré-traitement compense par-

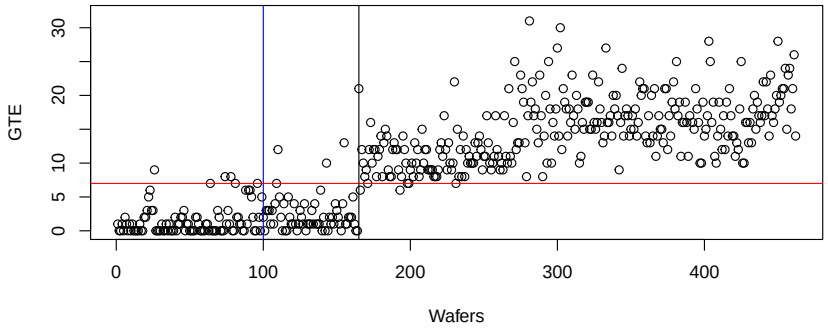

FIGURE 3.8 – Indice GTE pour l'exemple 3.6.1.

	Détections	Total	
Période 1 : Fausses alarmes	3	100	3%
Période 2 : Fausses alarmes	3	65	4.6%
Période 2 : Détection	288	298	97%

TABLE 3.1 – Performance du GTE pour l'exemple 3.6.1.

faitement les pertes de données. L'application du GTE filtré, décrit dans la Section 3.4.6 permet de supprimer ces 3 fausses alarmes, puisqu'elle n'ont pas de cause commune.

Dans la seconde période, les 20 premiers runs servent à tester si une faute a été introduite par la maintenance. Le test, décrit à la Section 3.5.2, ne détecte pas de telle faute. Ces mêmes runs sont alors utilisés pour mettre à jour le modèle (Section 3.5.3), et ce nouveau modèle permet ainsi de tester les runs qui suivent.

Une faute est détectée par le GTE à partir du run 166 dans cet échantillon. Nous considérons qu'une détection avant ce run est une fausse alarme, alors qu'une détection après ce run est valide. On observe un taux de fausses alarmes pour les runs 101 à 166 de 4.6%, et un taux de détection du run 167 au run 464 de 97%. Le filtrage permet de supprimer les 3 fausses alarmes, qui ont des causes distinctes, alors que seules les 2 premières détections (runs 167 et 168) sont ignorées puisqu'il faut au moins 3 détections identiques pour que la détection ne soit pas filtrée (voir Section 3.4.6). Ceci représente un coût négligeable par rapport au coût réel sans application de la méthodologie GTE : 298 runs défectueux ont été traités avant l'arrêt de l'équipement suite au test des wafers, alors que dès le 3^{e} run le GTE filtré apporte une détection.

Cet exemple montre la capacité du GTE à correctement détecter une

faute malgré la présence d'une maintenance entre l'ensemble d'apprentissage et l'ensemble des runs sur lesquels la faute survient.

Localisation statistique de la faute

Sur chaque run détecté, on applique la procédure de localisation permettant d'identifier le temps du procédé ainsi que les variables physiques responsables de la détection, comme décrit dans la Section 3.4.7. L'équipement a continué de fonctionner après le début de la faute, et on dispose de 298 runs pour identifier la faute. On peut ainsi se servir de l'ensemble de ces données pour proposer une identification de faute plus robuste. En utilisation en temps réel, la procédure s'applique sur la base du GTE filtré de la première alarme. La Figure 3.9 montre la contribution moyenne au step où la faute est repérée, le 2^{e}, en pourcentage sur toutes les variables, pour les capteurs les plus pertinents : c'est la variable "Pressure Baratron" qui se distingue comme étant responsable de la détection. Cette localisation apporte une aide à l'expert pour son identification de la cause mécanique de la faute.

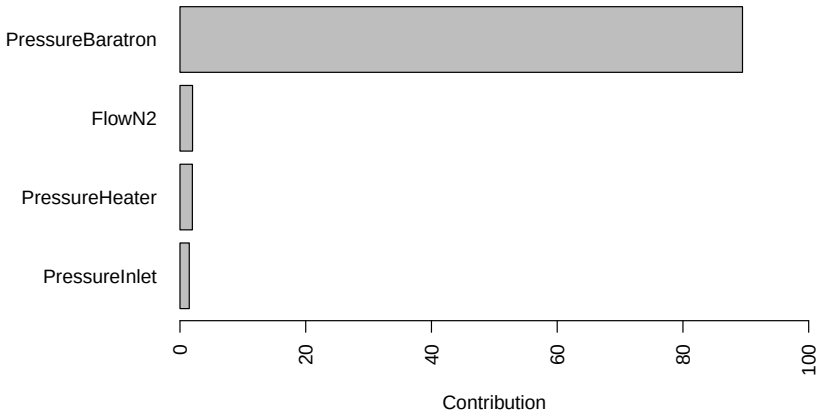

FIGURE 3.9 – Contributions à la faute pour l'exemple 3.6.1.

Taille des ensembles d'apprentissage

Dans les Sections 3.2 et 3.5.3, nous indiquons notre choix de considérer 100 runs pour l'apprentissage initial et 20 runs pour la mise à jour du modèle après une maintenance. Nous proposons ici d'étudier l'impact de ce choix sur cet exemple de données réelles. L'indicateur que nous prenons est la valeur moyenne de l'indice GTE, sans filtrage, sur les données saines

des ensembles de test. Sur de telles données, la valeur idéale du GTE est 0, car il n'y a pas de faute à détecter, et l'espérance est ici 1.7 compte-tenu des paramètres de la loi binomiale.

Pour la phase d'apprentissage initiale, on obtient les valeurs reportées dans le Tableau 3.2.

Nombre de runs d'apprentissage	GTE moyen
20	5
50	3.3
100	2.1
150	1.9

TABLE 3.2 – Valeur moyenne du GTE pour différentes tailles de l'ensemble d'apprentissage, sur la période 1, pour l'exemple 3.6.1.

Pour la phase de mise à jour après maintenance, on obtient les valeurs reportées dans le Tableau 3.3.

Nombre de runs d'apprentissage	GTE moyen
5	3.3
10	2.7
20	2.2
30	2.1

TABLE 3.3 – Valeur moyenne du GTE pour différentes tailles de l'ensemble de mise à jour, sur la période 2, pour l'exemple 3.6.1.

On observe ainsi qu'à partir de respectivement 100 runs et 20 runs, la décroissance de la valeur moyenne du GTE est fortement réduite. Le gain obtenu en employant plus de runs d'apprentissage est ainsi plus faible, alors que le coût d'obtention de données supplémentaires, tout particulièrement dans le cas de la mise à jour après maintenance, est élevé.

3.6.2 Une faute causée par la maintenance

Le deuxième exemple que nous présentons est basé sur une erreur humaine faite durant une maintenance d'un équipement de dépôt, dédié comme dans le premier exemple (Section 3.6.1) au CVD. Pendant la maintenance, un technicien n'a pas suffisamment resserré une pièce, ce qui a permis l'entrée d'une faible quantité d'oxygène dans la chambre de production. Son

impact n'était pas visible dans les steps couverts par des cartes de contrôle, et donc plusieurs centaines de runs ont été effectués avant que l'équipement ne soit arrêté pour une maintenance corrective.

Après la phase de pré-traitement, les données correspondent à $K = 190$ temps de mesure, associés à $J = 13$ capteurs. Comme pour l'exemple de la Section 3.6.1, 4 types de grandeurs physiques sont mesurées : des flux de gaz, des puissances électriques, des pressions et des températures. Les deux paramètres du GTE, α_d et α'_d, sont tous deux fixés à 0.1%. Nous disposons de données couvrant trois périodes :

— 557 runs venant d'une période saine ;
— 379 runs effectués après une première maintenance ;
— 331 runs effectués après une seconde maintenance.

Le modèle GTE initial est appris sur un ensemble de 100 runs choisis aléatoirement dans la première période. Les runs restants sont testés pour vérifier le taux de fausses alarmes. La procédure de gestion des maintenances est appliquée sur les 20 premiers runs des périodes 2 et 3, avec les paramètres α_m et α'_m tous deux fixés à 0.1%.

Détection et localisation de la faute

Les résultats complets sont présentés sur la Figure 3.10, qui montre les valeurs du GTE pour chacun des runs. La ligne horizontale représente la limite du GTE, alors que les lignes verticales délimitent les maintenances. Le Tableau 3.4 décrit les performances du GTE sur les 3 périodes.

	Détections	Total	
Période 1 : Fausses alarmes	6	457	1.3%
Période 2 : Maintenance fautive	132	358	37%
Période 3 : Fausses alarmes	1	311	0.3%

TABLE 3.4 – Performance du GTE pour l'exemple 3.6.2.

Dans la première période, les runs qui n'ont pas servi à l'apprentissage du modèle sont testés afin de mesurer le taux de fausses alarmes. On trouve un taux de 1.3%. Comme pour l'exemple de la Section 3.6.1, ce taux est très élevé devant α'_d, du fait d'une mauvaise collecte de données pour certains runs. Le filtrage du GTE supprime ces 6 fausses alarmes.

Sur la deuxième période, on teste d'abord la maintenance en utilisant les données des 20 premiers runs. Une faute est détectée dans ce cas, et donc l'équipement aurait été stoppé après le traitement du 20$^{\text{e}}$ run dans

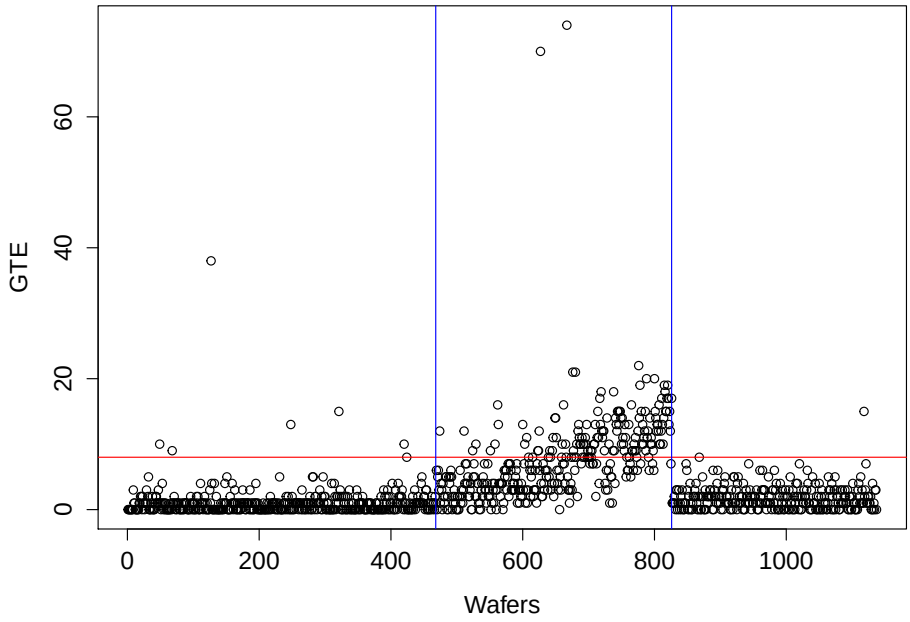

FIGURE 3.10 – Indice GTE pour l'exemple 3.6.2.

le cas d'une utilisation en temps réel du GTE. Cependant, 379 runs ont été effectués dans l'usine, et nous montrons les résultats obtenus en appliquant le modèle mis à jour sur la base des moyennes des 20 premiers runs sur la Figure 3.10. Ces résultats ne sont pas pertinents dans le cadre de la méthodologie GTE : on observe que le modèle a intégré une partie de la faute, ce qui empêche sa détection correcte, montrant ainsi l'intérêt d'effectuer le test de mise à jour.

Sur les 20 premiers runs de la période 2, on applique la procédure de localisation statistique de la faute, décrite à la Section 3.5.2, afin de trouver les steps de la recette ainsi que les capteurs permettant d'identifier la faute de l'équipement. La Figure 3.11 montre les contributions des variables les plus pertinentes au step d'initialisation de la recette : la variable "Back Pressure" est désignée comme responsable de la faute, ce qui permet d'identifier la fuite correspondante.

Pour la troisième période, qui suit la réparation de ce problème, le test de variance sur les 20 premiers runs n'apporte pas de détection. Après la mise à jour, aucun run n'est censé être détecté. On obtient une fausse alarme, soit un taux de 0.3%, et cette unique fausse alarme est donc filtrée,

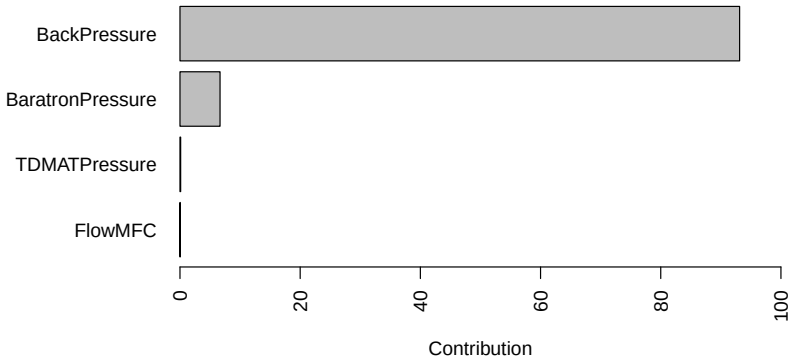

FIGURE 3.11 – Contributions à la faute de l'exemple 3.6.2.

puisqu'au moins 3 alarmes sont nécessaires pour qu'une alarme soit visible via le filtre proposé.

Taille des ensembles d'apprentissage

Dans les Sections 3.2 et 3.5.3, nous indiquons notre choix de considérer 100 runs pour l'apprentissage initial et 20 runs pour la mise à jour du modèle après une maintenance. Nous proposons ici d'étudier l'impact de ce choix sur cet exemple de données réelles. L'indicateur que nous prenons est la valeur moyenne de l'indice GTE, sans filtrage, sur les données saines des ensembles de test. Sur de telles données, la valeur idéale du GTE est 0, car il n'y a pas de faute à détecter, et l'espérance est ici 2.5 compte-tenu des paramètres de la loi binomiale.

Pour la phase d'apprentissage initiale, on obtient les valeurs reportées dans le Tableau 3.5.

Nombre de runs d'apprentissage	GTE moyen
20	1.9
50	1.2
100	1
150	0.9

TABLE 3.5 – Valeur moyenne du GTE pour différentes tailles de l'ensemble d'apprentissage, sur la période 1, pour l'exemple 3.6.2.

Pour la phase de mise à jour après maintenance, on obtient les valeurs reportées dans le Tableau 3.6.

Nombre de runs d'apprentissage	GTE moyen
5	1.4
10	1.3
20	1.3
30	1.3

TABLE 3.6 – Valeur moyenne du GTE pour différentes tailles de l'ensemble de mise à jour, sur la période 3, pour l'exemple 3.6.2.

Par rapport à l'exemple 3.6.1, l'équipement étudié a un comportement plus répétable sur la période observée. Ainsi, les valeurs proposées de 100 runs d'apprentissage et 20 runs de mise à jour pourraient être réduites à 50 runs d'apprentissage et 10 voire 5 runs de mise à jour, sans perte significative d'information dans le modèle.

3.6.3 Un défaut intermittent

Dans les deux exemples précédents (Sections 3.6.1, 3.6.2), nous étudions le cas d'une faute impactant tous les runs consécutifs à son apparition. Ce troisième exemple illustre un cas où une faute ne se manifeste que rarement dans les données, de façon irrégulière et espacée.

Ce troisième exemple porte sur la détection d'une valve défectueuse, entraînant une manifestation intermittente d'une fuite, sur un équipement de retrait résine. Le travail de tels équipements est de brûler la résine décrivant le dessin des circuits imprimés, afin qu'il ne reste que les matériaux déposés sur les wafers.

Après la phase de pré-traitement, les données correspondent à $K = 597$ temps de mesure, associés à $J = 27$ capteurs. Dans cet équipement, 5 types de grandeurs physiques sont mesurées : des flux de gaz, des puissances électriques, des pressions, des températures et des voltages. Comparativement aux deux exemples précédents, il y a un nombre significativement plus élevé de temps et de capteurs. Par conséquent, nous fixons le paramètre α_d à 0.01%, pour permettre une baisse de l'erreur de Type-II sur les fautes de courte durée (voir Section 3.4.5). L'erreur de Type-I α'_d reste fixée à 0.1% comme dans les exemples précédents. Nous disposons de données couvrant deux périodes :

— 280 runs venant d'une période saine ;

— 1516 runs effectués après une maintenance.

Le modèle GTE initial est appris sur un ensemble de 100 runs choisis aléatoirement dans la première période. Les runs restants sont testés pour vérifier le taux de fausses alarmes. La procédure de gestion des maintenances est appliquée sur les 20 premiers runs de la période 2, avec les paramètres α_m et α'_m fixés respectivement à 0.01% et 0.1% (voir Section 3.5.2).

Détection de la faute

Les résultats complets sont représentés sur la Figure 3.12, qui présente l'indice GTE pour tous les runs. La ligne horizontale représente la limite du GTE, calculée de façon théorique à partir de la loi binomiale. La ligne verticale désigne à la fois la maintenance et l'apparition du défaut, puisque les deux évènements sont simultanés. Pour distinguer les fausses alarmes des détections correctes, nous utilisons un critère métier indiquant si le run a été affecté ou non, basé sur la valeur maximale de la pression lors d'un step spécifique. Il y a 51 runs affectés. Le Tableau 3.7 résume les performances du GTE sur cet exemple.

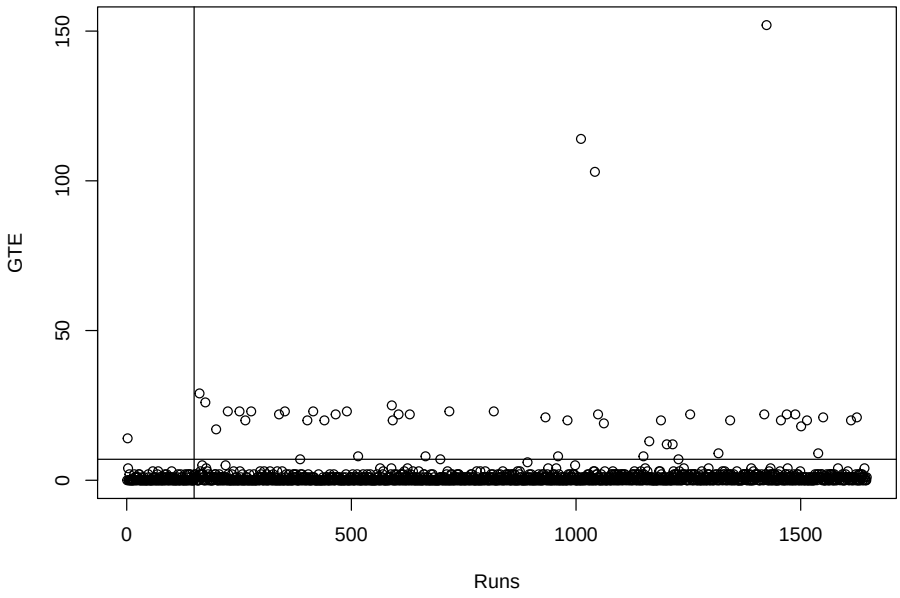

FIGURE 3.12 – Indice GTE pour l'exemple 3.6.3.

	Détections	Total	
Période 1 : Fausses alarmes	1	150	0.6%
Période 2 : Fausses alarmes	5	1446	0.3%
Période 2 : Détection	43	50	86%

TABLE 3.7 – Performance du GTE pour l'exemple 3.6.3.

Dans la première période, les runs qui n'ont pas été utilisés pour construire le modèle sont testés. Aucune faute n'est censée être vue, et on obtient un taux de fausses alarmes de 0.6%. L'application du GTE filtré, décrit dans la Section 3.4.6, permet de supprimer la fausse alarme puisqu'il faut au moins 3 alarmes avec le filtrage appliqué pour obtenir une alarme non filtrée.

Dans la seconde période, les 20 premiers runs servent à tester si une faute a été introduite par la maintenance. Le test, décrit à la Section 3.5.2, ne détecte pas de telle faute. Ces mêmes runs sont alors utilisés pour mettre à jour le modèle (Section 3.5.3), et ce nouveau modèle permet ainsi de tester les runs qui suivent. Une analyse a posteriori montre qu'un des runs affectés se trouve dans cet ensemble de 20 runs. Ce run n'a pas d'influence sur la classification de la maintenance comme mauvaise, ni sur la mise à jour des paramètres, puisque les estimateurs calculés sont tronqués (voir Section 3.5.4). Ainsi, l'équipement n'est pas arrêté, ce qui est une erreur puisqu'il y a une défaillance. Cependant, le run qui présente le défaut n'affecte pas la mise à jour du modèle : la faute n'a pas été apprise et peut encore être détectée par la suite, parmi les 50 runs affectés restants.

Une faute est détectée dans la période 2 de façon irrégulière. Sur les 48 runs détectés, 41 restent après filtrage. Les 5 fausses alarmes ont été filtrées, et les 2 premiers runs affectés par la faute également. Ces 41 runs restants correspondent à 41 des 50 runs affectés dans cette période. Avec une utilisation du GTE en temps réel, la détection aurait eu lieu au 4^e run affecté, puisque le premier run affecté n'est pas détecté car inclus dans l'ensemble de mise à jour. L'impact final de cette faute sur les produits était faible, mais le coût a tout de même été important du fait des semaines de production à risque qui ont suivi, avec la nécessité de faire des tests supplémentaires sur tous les wafers. Après cette maintenance, 1516 runs étaient à risque. Avec le GTE, le 60^e run correspond à la première alarme non filtrée, ce qui représente un gain significatif.

Cet exemple montre ainsi la possibilité de séparer les fausses alarmes des véritables fautes grâce au filtrage du GTE. Il montre également la robustesse du test de maintenance : un seul run semblant être affecté par une faute est une information trop faible pour arrêter un équipement, mais il

est important de s'assurer de ne pas apprendre dans le modèle les caractéristiques de ce run.

Localisation statistique de la faute

Sur les runs dont le GTE filtré est hors-limite, nous appliquons la procédure de localisation statistique de la faute (voir Section 3.4.7). La faute est localisée au 14^{e} step, et 4 capteurs sont désignés comme principaux responsables. Sur la Figure 3.13, nous présentons les valeurs de contributions pour les 6 capteurs ayant les contributions les plus élevées. L'observation des données montre un décalage de la variable "PressureManometer" au début du step, qui est suivi d'un ajustement des trois variables de puissance "PowerForward", "PowerReflected" et "PowerLoad". Un expert a ainsi pu attester la provenance de la faute.

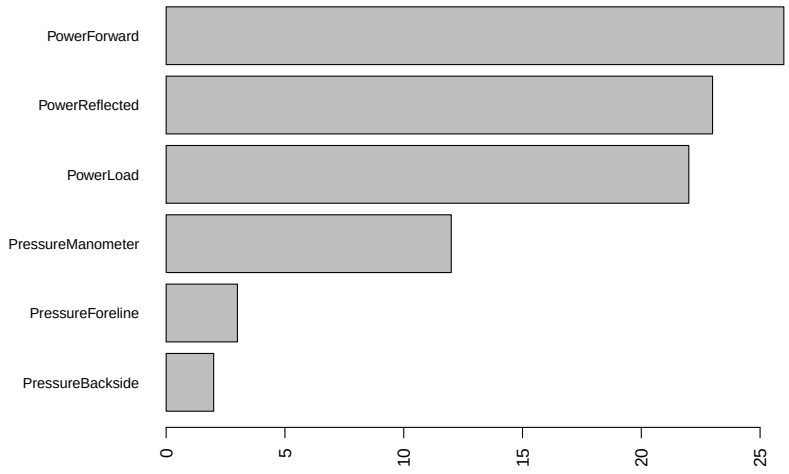

FIGURE 3.13 – Contributions à la faute pour l'exemple 3.6.3.

Taille des ensembles d'apprentissage

Dans les Sections 3.2 et 3.5.3, nous indiquons notre choix de considérer 100 runs pour l'apprentissage initial et 20 runs pour la mise à jour du modèle après une maintenance. Nous proposons ici d'étudier l'impact de ce choix sur cet exemple de données réelles. L'indicateur que nous prenons

est la valeur moyenne de l'indice GTE, sans filtrage, sur les données saines des ensembles de test. Sur de telles données, la valeur idéale du GTE est 0, car il n'y a pas de faute à détecter, et l'espérance est ici 1.6 compte-tenu des paramètres de la loi binomiale. Dans ce cas d'étude, le faute est intermittente : il n'y a qu'une seule véritable période saine. Nous testons donc l'impact de la taille de l'ensemble d'apprentissage initial sur le GTE calculé sur cette période de test. On obtient les valeurs reportées dans le Tableau 3.8.

Nombre de runs d'apprentissage	GTE moyen
20	1.7
50	1.5
100	1.2
150	1.2

TABLE 3.8 – Valeur moyenne du GTE pour différentes tailles de l'ensemble d'apprentissage, sur la période 1, pour l'exemple 3.6.3.

Pour 150 runs d'apprentissage, nous n'observons pas de baisse du GTE moyen par rapport à celui obtenu avec 100 runs d'apprentissage : il n'est pas nécessaire d'obtenir plus de données pour la création du modèle.

3.7 Conclusion

Dans ce chapitre, nous avons présenté la partie supervisée, portant sur les données d'un seul équipement à la fois, de la méthodologie de détection de fautes que nous proposons. Il s'agit de la contribution principale de cette thèse.

Cette méthodologie se nomme Gaussian Time Error, et permet de détecter des fautes sur les équipements, en employant les données de runs individuels. Cette méthodologie est basée sur l'apprentissage d'un modèle à partir de données de référence. Une mise à jour de celui-ci à chaque maintenance, conditionnée par un test sur la répétabilité observée du procédé, est effectuée.

Les données étudiées sont modélisées en utilisant des lois normales indépendantes à chaque instant du procédé. L'indice de détection GTE est basé sur une combinaison de tests mono-variés, effectués grâce à une projection des données sur un espace où celles-ci sont décorrélées. Cet indice compte le nombre d'instants où le procédé est vu comme anormal, et suit une loi binomiale dans le cas d'un procédé sans défaut.

Nous avons présenté dans ce chapitre différents cas industriels, permettant de montrer la capacité de détection de cet algorithme dans des situations diverses. Le premier exemple montre une faute sans rapport avec une maintenance, le deuxième exemple montre une faute directement liée à une maintenance, et le troisième exemple montre une faute intermittente. Dans les trois cas, la détection est efficace et permet à l'industriel de prendre les actions correctives nécessaires dans des délais réduits.

En employant uniquement cette méthodologie, le cas d'une faute lors d'une maintenance correspondant à un procédé ayant une répétabiité similaire à celle observée dans l'ensemble d'apprentissage ne peut pas être détecté. La méthodologie non-supervisée, employant les données de plusieurs équipements, que nous proposons dans le Chapitre 4 permet d'apporter une solution à ce problème.

Chapitre 4

Méthodologie de comparaison d'équipements

Dans ce chapitre, nous présentons notre contribution méthodologique permettant la comparaison non-supervisée, sur la base des données FDC, d'équipements effectuant les mêmes recettes.

Les centaines de recettes appliquées dans une route, éventuellement réalisées plusieurs fois, nécessitent d'avoir plusieurs équipements capables d'effectuer la même tâche. Le temps de fabrication du wafer est de ce fait réduit, mais l'homogénéité des produits en sortie est mise en danger : en effet, s'agissant de haute précision, les spécificités d'un équipement donné risquent d'impacter le produit final.

Il est donc nécessaire d'évaluer la cohérence de la production des équipements effectuant les mêmes tâches, afin de réduire l'écart final entre les produits. Dans le cadre d'une analyse en temps réel de l'écart entre équipements, les données FDC sont les seules à être systématiquement disponibles. Pour ces données, il n'est pas évident d'obtenir des données de référence. En effet, de même que pour la détection de faute, traitée au Chapitre 3, il n'y a pas de certitude quant à l'état instantané d'un équipement donné. Il n'y a pas non plus d'équipement de référence dans chaque atelier : les équipements sont maintenus fonctionnels du mieux possible, et il n'y a en général pas de critère objectif permettant de classer la qualité des produits sortis d'un équipement ou d'un autre.

Ainsi, nous proposons ici une méthodologie de comparaison des données FDC non-supervisée. L'objectif est alors de signaler les équipements dont les données sont différentes du reste de l'atelier. Ces différences signalées doivent être pertinentes, relativement au procédé et aux objectifs de productivité : il n'est pas concevable d'arrêter une grande partie des équipements pour faire des re-réglages mineurs, à cause de la réduction

des capacités de production engendrée. En effet, une telle démarche ne pourrait pas être envisagée, même sur le long terme, car il s'avère que les maintenances fréquentes et régulations permanentes des équipements rendraient ces actions inutiles. Les légères différences initiales entre mêmes types d'équipements impliquent souvent qu'une correspondance exacte du point de vue des données FDC n'est pas toujours possible, ni même souhaitable.

Nous avons présentés les apports introduits dans ce chapitre dans [42].

4.1 Comparaisons entre équipements effectuant un même procédé

Nous avons montré, dans la Section 3.5, les enjeux de la ré-adaptation des paramètres de moyenne dans les modèles GTE : la représentation statistique du bon fonctionnement de l'équipement se doit d'évoluer au fur et à mesure des opérations de maintenance effectuées sur celui-ci. Le modèle GTE peut être mis à jour en termes de moyennes sous la condition que les variances observées tout au long du procédé restent suffisamment proches de celles observées dans l'ensemble d'apprentissage initial.

Ainsi, aucun test n'est effectué concernant les valeurs moyennes des signaux récoltés par les capteurs : une faute d'amplitude entraînant l'observation de signaux aussi répétables que ceux observés dans l'ensemble d'apprentissage ne peut pas être détectée par cette méthode. Dans un tel cas, la faute est transmise dans le modèle, et les répercussions peuvent être importantes dans le cas où elle n'est pas vue par d'autres moyens (comme par exemple les mesures sur les wafers). Il est donc nécessaire d'ajouter une strate supplémentaire de surveillance, qui tienne compte des valeurs moyennes ignorées dans le modèle GTE.

La réponse statistique à ce problème ne peut que très difficilement apparaître dans l'historique de l'équipement lui-même. En effet, les maintenances peuvent être espacées de plusieurs mois pour certains types d'équipements. L'historique accessible dans un milieu industriel ne porte que rarement sur de très longues périodes, du fait du volume très conséquent de données collectées : à STMicroelectronics Rousset, les données ne sont conservées que 6 mois. Constituer un ensemble de fonctionnements de référence demande de plus un travail supplémentaire : il faut répertorier les périodes distinctes, vérifier pour chacune l'absence de défauts, et ainsi refaire à chaque maintenance le travail qui n'est fait qu'une seule fois pour le modèle GTE. La modification significative d'une recette de production peut

remettre en cause tout ce travail, forçant ainsi une longue attente avant d'avoir suffisamment de périodes séparées par des maintenances pour pouvoir constituer une référence de taille suffisante.

Dès lors, faire intervenir les données du présent pour juger de la qualité de l'équipement est une solution nettement plus applicable en pratique. En effet, les impératifs de production font qu'un même procédé est réalisé sur un parc d'équipements plutôt que sur un seul : exploiter ceci permet d'avoir des données de comparaison sans aucun travail supplémentaire. Utiliser une comparaison non-supervisée de tous ces équipements présente deux avantages : l'équipement reprenant la production suite à une maintenance est testé face aux autres équipements en cours de fonctionnement, ou ayant fonctionné dans les jours précédents sur le même procédé, et l'homogénéité globale de la production est assurée à travers la comparaison.

Utiliser une méthode non-supervisée est un bénéfice important dans le cas de l'industrie du semi-conducteur. Les produits ne sont réellement testés qu'en bout de chaîne : toute donnée venant d'être récoltée, malgré les nombreuses précautions prises lors de la production, doit par défaut être considérée comme suspecte. On fait alors l'hypothèse que si une chambre se distingue des autres, c'est qu'elle présente potentiellement une faute.

4.2 État de l'art

Nous n'avons pas connaissance de l'existence d'une méthodologie s'appliquant directement à la problématique de la comparaison d'équipements basée sur les données FDC dans le domaine du semi-conducteur, ni sur des données similaires pour d'autres types de procédés. Nous proposons ainsi de distinguer les différents éléments constituants cette problématique pour y apporter une solution.

Dans la Section 2.3.1, nous traitons de la problématique de l'apprentissage statistique. La comparaison de chambres est également un problème d'apprentissage à partir de données. Cependant, là où, dans le cas de la détection de faute, un individu est la matrice représentant les données d'un run, dans le cas de la comparaison de chambres, il s'agit d'un ensemble tridimensionnel, constitué des matrices de données d'un nombre représentatif de runs pour une chambre. Dès lors, à l'échelle du run, nous disposons déjà d'un partitionnement des données, qui nous indique dans quelle chambre le run a été effectué, c'est-à-dire à quelle classe l'individu appartient. Il s'agit alors de déterminer si, parmi les classes dont on dispose, une ou plusieurs se distinguent de l'ensemble des autres. Les classes pour les runs, que sont les chambres, sont séparées en deux ensembles : chambres atypiques,

nécessitant intervention de re-réglage, et chambres typiques, dont la production peut continuer. C'est un problème de classification non-supervisée : il n'y a pas d'a priori sur la qualité des chambres avant analyse.

Notre analyse de la problématique nous amène à scinder la méthodologie que nous proposons en trois parties. Nous menons une étude de la littérature spécifique pour chacun de ces points :

1. Détermination d'une distance entre classes : on détermine un individu représentant la classe, et on se ramène au cas de la distance entre individus ;
2. Détermination d'une distance entre individus : l'individu dans ce cas correspond à une série temporelle multi-variée ;
3. Détection de valeurs aberrantes parmi la distribution des distances entre classes.

Les données étudiées restent les mêmes que celles caractérisées dans les Sections 1.2.3 et 2.1. Une chambre c pour une stratégie donnée est ainsi caractérisée par les données de I_c runs, qui pour chaque run i individuel correspondent à une matrice de dimensions $J \times K_i$ où J est le nombre de capteurs de l'équipement et K_i le nombre de points récoltés pour le run i.

4.2.1 Distances entre classes

Le premier aspect de la méthodologie proposée est la détermination d'une distance entre classes de runs.

S'agissant de comparer des groupes de runs, il est nécessaire de déterminer une distance entre ces groupes sur la base des distances entre runs individuels. Le terme de distance entre classes est ici à considérer dans son sens commun, qui est "l'évaluation d'un écart", et non pas au sens mathématique du terme : les propriétés de séparation et d'inégalité triangulaire ne sont pas vérifiées pour toutes les distances proposées.

La problématique des distances inter-classes est un aspect important de l'algorithme de partitionnement de données nommé "regroupement hiérarchique" [68]. Dans cet algorithme, dont le détail peut être trouvé par exemple dans [23], la distance inter-classes est utilisée dans le but de regrouper, en une unique classe, certaines classes de données initialement distinctes. Ainsi, la littérature traitant de ces méthodes apporte à la fois des exemples de telles distances et une évaluation de leur intérêt dans les applications pratiques.

On note $d(a, b)$ la distance entre un élément a de la classe A et un élément b de la classe B. On note $D(A, B)$ la distance entre les deux classes. Les distances inter-classes les plus utilisées sont les suivantes [21] :

— Le saut maximum, défini par $D_1(A,B) = \max\{d(a,b) : a \in A,\, b \in B\}$, caractérise la distance entre classes par la distance entre les points les plus éloignés ;
— Le saut minimum, défini par $D_2(A,B) = \min\{d(a,b) : a \in A,\, b \in B\}$, caractérise la distance entre classes par la distance entre les points les plus proches ;
— Le saut moyen, défini par $D_3(A,B) = \frac{1}{|A||B|}\sum_{a\in A}\sum_{b\in B} d(a,b)$, caractérise la distance entre classes par l'écart moyen entre les points de chacune des deux classes ;
— La distance entre les centres de gravité, définie par $D_4(A,B) = d(G_A,G_B)$ où G_A et G_B sont les centres de gravité de A et de B, caractérise la distance entre classes par l'écart entre deux représentants de la classe ;
— La distance de Ward [77] pondère la distance entre les centres de gravité en se basant sur le nombre d'individus de chacune des deux classes : $D_5(A,B) = \frac{|A||B|}{|A|+|B|}d(G_A,G_B)$.

Chacune de ces distances définit la proximité de deux ensembles d'une manière différente. Pour illustrer cela, nous prenons un exemple simple. On considère $A = \{1,3,5\}$, $B = \{2,8,9,17\}$ et d la distance euclidienne classique. Dans ce cas, on obtient les valeurs suivantes : $D_1(A,B) = |1-17| = 16$, $D_2(A,B) = |1-2| = 1$, $D_3(A,B) = \frac{10}{3}$, $D_4(A,B) = |3-9| = 6$ et $D_4(A,B) = \frac{12}{7}|3-9| = \frac{72}{7}$. On observe qu'une modification d'un élément de A ou de B n'affecte pas nécessairement chacune de ces distances.

Dans la Section 2.1, nous avons vu que le risque d'erreurs de collecte est élevé dans le cas des données étudiées. Ainsi, l'utilisation du saut maximum est problématique : dans ce cas, les équipements peuvent être comparés sur la base des données de runs les moins représentatives de l'état réel de l'équipement. Le saut minimum est sujet aux mêmes types de problèmes : un run individuel peut ne pas représenter l'équipement et faire conclure sur la base d'une seule observation que deux équipements sont plus proches qu'ils ne le sont en réalité. Le saut moyen est moins sujet aux valeurs extrêmes que les deux distances précédentes, mais sa valeur peut tout de même rester affectée par celles-ci.

L'écart entre les centres de gravité est une proposition satisfaisante dans notre cas. En effet, sous réserve d'une estimation suffisamment robuste des centres de gravité, représenter chaque équipement par un run de référence permet de supprimer l'influence des problèmes de collecte : aucun run individuel ne sert directement à évaluer l'écart entre les équipements. Nous proposons un calcul de centres de gravité spécifique à notre application

dans la Section 4.4.3. La pondération sur le nombre de runs proposée avec la distance de Ward n'a pas de sens dans notre cas d'utilisation. Le nombre de runs collectés dépend de nombreux paramètres, comme par exemple l'activité de l'équipement sur une recette donnée ou encore la date de la dernière maintenance. Il n'y a pas d'intérêt à considérer que deux équipements donnés sont plus ou moins éloignés en fonction du nombre de runs dont les données sont disponibles au moment de l'application de l'algorithme.

4.2.2 Distances entre séries temporelles

La distance entre les équipements, qui sont les classes dans notre application, peut ainsi être ramenée à une distance entre runs, qui sont les individus ici. Dans cette section, nous traitons donc des multiples possibilités pour calculer une distance entre deux runs, représentés par les séries temporelles collectées pendant leur déroulement.

Une première approche peut être de considérer une série temporelle de longueur n comme un vecteur quelconque de $\mathbb{R}^n$. On peut alors utiliser les normes L_p dans ce contexte [83]. Si on considère deux vecteurs X et Y, $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n) \in \mathbb{R}^n$, alors la distance entre X et Y induite par la norme L_p a pour expression $(\sum_{i=1}^{n} |x_i - y_i|^p)^{1/p}$. Les valeurs les plus utilisées de p sont 2, ce qui correspond à la distance euclidienne, et 1, ce qui correspond à la distance de Manhattan. Aucune de ces distances ne s'avère satisfaisante dans notre contexte d'application. En effet, ces distances sont très sensibles au bruit, et tout particulièrement aux problèmes d'alignement de données [1].

Pour pallier à ce problème, il est possible de combiner l'alignement DTW (voir Section 2.2.2) avec la distance euclidienne. On effectue d'abord un alignement DTW entre les deux trajectoires. Ceci permet d'obtenir une dimension commune à ces deux séries temporelles, et également d'ajuster les décalages de durée entre mêmes sous-parties des deux signaux. Il est alors possible de tenir compte des décalages temporels. La distance euclidienne calculée entre les deux trajectoires après alignement est nommée "distance DTW" [2]. Cependant, seuls les décalages et les changements d'échelle temporels sont gérés [24] : si les deux signaux présentent un décalage d'amplitude systématique, il s'accumule à chaque temps. Cette situation est fréquente, par exemple sur les paramètres électriques récoltés, empêchant ainsi l'utilisation de cette distance dans notre cas.

Un autre type de distance tente de pallier au manque de gestion des décalages temporels des distances L_p : les distances basées sur la distance

d'édition. La distance d'édition est une distance utilisée pour la comparaison de chaînes de caractères, dépendant du nombre de caractères en commun et de leur ordre d'apparition. La même idée peut être appliquée à des séries temporelles. Par exemple, la distance LCSS (sigle pour "longest common subsequence"), entre deux séries temporelles X et Y, cherche à déterminer le nombre minimal d'éléments de X qu'il faut transférer vers Y pour transformer Y en X [14]. Cependant, le critère permettant de dire si un élément de Y est suffisamment proche d'un élément de X est basé sur l'amplitude : on retrouve donc le même défaut que pour la distance DTW. Différentes variantes de cette distance existent, comme par exemple la distance EDR ("edit distance on real sequences" [8]) et ERP ("edit distance with real penalty" [7]), mais elles n'apportent pas de solution au problème des écarts d'amplitude.

Enfin, une proposition tente de résoudre en même temps les problèmes liés aux décalages temporels et aux décalages d'amplitudes : la distance SpADE ("spatial assembling distance" [10]). Avec cette distance, chacune des deux trajectoires à comparer est découpée en un nombre fixe de sous-sections, et celles-ci sont résumées sous différents paramètres caractérisant leur position et leur amplitude. Il est alors possible d'assigner ou non des pénalités à tout type de transformations. Cette distance permet ainsi une grande liberté d'utilisation, mais est très soumise au choix des paramètres, qui sont très nombreux [16] : une connaissance experte des séries temporelles à comparer est impérative en vue d'obtenir des résultats satisfaisants. Dans notre cas, la distance à employer doit être générique car la comparaison s'effectue sur tout type de recettes, sans a priori.

Nous proposons ainsi une approche différente, plus adaptée à notre cas d'étude, décrite à la Section 4.4.1. Celle-ci est basée sur la combinaison de l'algorithme DTW et de la régression linéaire.

4.2.3 Détection de valeurs aberrantes

Une fois les distances entre équipements deux à deux déterminées, nous proposons de détecter, de façon non-supervisée, les éventuels équipements atypiques. Cela revient à détecter des valeurs aberrantes dans la distribution des distances entre équipements.

On appelle "valeur aberrante" une observation qui semble se distinguer fortement des autres observations de la distribution dont elle est issue [25]. Effectuer de la détection non-supervisée revient à rechercher des valeurs aberrantes dans une distribution donnée. Dans la pratique statistique, la recherche de valeurs aberrantes est utilisée car, en présence de telles ob-

servations, la modélisation statistique des données est modifiée. Cela engendre un modèle inadapté, ou bien un modèle bien plus complexe qu'il ne le serait sans ces observations. Dans notre cas d'étude, la recherche de valeurs aberrantes est une fin en soi : notre objectif est de déterminer les équipements qui se distinguent des autres, et il n'y a pas d'autre étape suivant cette détection.

De très nombreuses méthodes existent, pouvant être redondantes ou destinées à des usages spécifiques [12, 59, 62, 72]. La présence d'hypothèses fortes, comme par exemple la normalité de la distribution étudiée, peut s'avérer un handicap dans notre application : les parcs d'équipements ne dépassent que très rarement les 15 unités. De plus, s'agissant de déterminer des valeurs aberrantes dans des ensembles de distance entre paires d'équipements, les valeurs étudiées ne sont pas indépendantes.

Les méthodes de détection de valeurs aberrantes sont soumises à deux grandes problématiques, que sont les effets de masquage et d'envahissement :

— On parle d'effet d'envahissement quand la présence d'une valeur aberrante amène à considérer qu'une autre valeur est également une valeur aberrante, alors que ce ne serait pas le cas en l'absence de la première valeur. Par exemple, si on considère la distribution $\{-14, 0, 1, 2, 2, 5\}$, la présence de -14 pourrait amener à considérer que 5 est aberrant. Si, en l'absence de -14, 5 n'est pas considéré comme aberrant, alors 5 est aberrant du fait de l'effet d'envahissement.

— L'effet de masquage est l'effet inverse : la présence d'une valeur aberrante dans la distribution cache une autre valeur aberrante. Par exemple, si on considère la distribution $\{-10, -9, -1, 1, 2\}$, la présence de -10 peut amener à ne pas considérer -9 comme aberrant. Si, en l'absence de -10, -9 est considéré comme aberrant, on parle alors d'effet de masquage.

Pour éviter ces deux effets, la proposition la plus usitée est d'employer des estimations robustes de la dispersion de la distribution étudiée [61]. À partir de telles mesures, il est possible de définir un ensemble de rejet caractérisant les observations trop éloignées du reste de la distribution. S'agissant d'une application pratique en temps réel, partir d'une estimation explicite, c'est-à-dire basée sur une formule dont tous les termes sont connus, est un avantage important en termes de rapidité de calcul. Parmi les estimateurs existant dans la littérature, nous nous intéressons aux L-estimateurs, basés sur des combinaisons linéaires des statistiques d'ordre de la distribution, plutôt qu'aux M-estimateurs, construits par la minimisation d'une fonction dépendant des observations et de paramètres de modé-

lisation [27].

La méthode de l'écart inter-quartile (appelée IQR pour "inter-quartile range") est très utilisée car très simple, et robuste [74]. On définit l'écart inter-quartile d'une distribution comme étant :

$$IQR = Q_3 - Q_1, \tag{4.1}$$

avec Q_3 le troisième quartile et Q_1 le premier quartile de l'échantillon. Toute observation hors de l'intervalle $[Q_1 - 3IQR, Q_3 + 3IQR]$ est considérée comme éventuellement aberrante. Cette méthode est robuste, relativement à la notion de point de casse.

Définition 3. *Pour un estimateur E et une distribution D, le point de casse se définit comme étant la valeur $Pc(E, D)$:*

$$Pc(E, D) = \inf_{1>\epsilon>0} \sup_{D_\epsilon} (|E(D_\epsilon) - E(D)| = \infty), \tag{4.2}$$

où D_ϵ désigne la distribution construite à partir de D dans laquelle une proportion ϵ est remplacée par des valeurs arbitraires, pouvant donc tendre vers l'infini.

L'écart inter-quartile a un point de casse de 25% (il n'est pas modifié en la présence de moins de 25% de valeurs aberrantes) [61]. Par exemple, toute méthode basée sur la moyenne a un point de casse de 0 [17] : la modification de n'importe quelle observation modifie l'estimation.

Cette méthode est originellement basée sur une hypothèse de normalité : en effet, dans ce cas, il y a environ 0.1% des observations qui sont attendues hors de cet intervalle. Dans le cas non normal, la valeur 3 utilisée pour construire l'intervalle n'aura pas nécessairement la même signification, même si cette méthode peut rester efficace.

Une alternative à l'écart inter-quartile est l'écart médian absolu (appelé MAD pour "median absolute deviation"), également très utilisé dans la pratique [38]. Son expression est la suivante, pour un ensemble de n observations $X = (X_1, ..., X_n)$: $\mathrm{MAD} = \text{médiane}\,(\,|X_i - \text{médiane}(X)|\,)$. L'avantage théorique de cet estimateur est son point de casse, qui est de 50% [61]. Cependant, dans notre cas d'étude, cela peut mener à un plus grand taux de fausses alarmes dans la détection d'équipements atypiques. En effet, un sous-groupe d'équipements plus proches entre eux que les autres, par exemple des équipements qui ont été achetés en même temps alors que d'autres l'ont été au fur et à mesure des investissements de l'usine, peut influencer de façon trop disproportionnée l'estimation.

Nous proposons une adaptation de la méthode de l'écart inter-quartile à notre cas d'étude dans la Section 4.4.5.

4.3 Comparaison de formes entre signaux

Nous proposons pour ce problème de procéder à une comparaison de la forme des signaux récoltés par les capteurs, par une analyse menée indépendamment sur chacun d'entre eux.

4.3.1 Vocabulaire du changement de formes

Nous définissons tout d'abord le vocabulaire utilisé pour la détection de formes dans des jeux de données, en se basant sur [67]. Celui-ci est issu de la morphométrie, discipline correspondant à l'étude des formes et dont l'application principale se trouve en biologie [18, 45, 58].

Définition 4. *La **forme** d'une figure est l'information géométrique restant après que les informations de rotation, d'échelle et de localisation aient été supprimées.*

Définition 5. *Un **point d'intérêt** est un point permettant la correspondance entre objets de même forme.*

Les points d'intérêts sont en général de deux types : mathématiques ou anatomiques. Les points d'intérêts anatomiques sont placés relativement à une connaissance experte, alors que les points d'intérêts mathématiques sont placés relativement à des propriétés géométriques.

4.3.2 Fondements de l'approche

Pendant le déroulement du procédé, la forme du signal récolté est le témoin du suivi de toutes les étapes. Notre expérience industrielle nous a amenés à se limiter à une comparaison de forme mono-variée : chaque capteur est étudié indépendamment des autres. En effet, la dépendance faible d'un nombre significatif de capteurs fait que l'étude comparative des corrélations observées pour les différents équipements n'est pas rentable. Il n'est que rarement possible pour l'industriel d'agir sur les corrélations, alors que le re-réglage d'un paramètre donné est en général immédiatement réalisable. Les corrélations observées pour un équipement donné ne doivent pas présenter d'évolutions significatives, et un tel changement serait vu comme une faute pour le GTE (voir Chapitre 3), mais aucune règle similaire n'a pu être déterminée pour la comparaison d'équipements distincts.

Les contraintes s'exerçant sont les suivantes :

1. Dans notre étude, la forme étudiée concerne des courbes continues, par exemple la pression en un certain endroit de l'équipement au cours du procédé, décrites par des capteurs en donnant une représentation échantillonnée. Pour chaque courbe étudiée, la distribution des points d'intérêt est l'ensemble des valeurs échantillonnées récoltées au cours de l'exécution du procédé.
2. Les recettes sont réalisées en temps variable : les signaux peuvent donc être étirés, et ce de façon non nécessairement uniforme. Il s'agit de la même contrainte engendrant la nécessité d'un alignement dans la phase de pré-traitement du GTE (voir Section 3.3).
3. En raison des multiples réglages possibles, les signaux observés peuvent avoir un décalage systématique, voire même une échelle globale variable. Notre hypothèse est qu'un équipement présentant un comportement atypique le fera de façon ciblée dans la recette : ainsi une amplitude anormale sera observée sur une des étapes, altérant alors la forme globale du signal relativement à celle observée sur les autres équipements, indépendamment de l'échelle du signal, et de sa localisation verticale.
4. Dans la définition de la notion de forme, au paragraphe précédent, nous évoquons la possibilité d'effectuer une rotation : ceci n'a pas de sens dans notre application à des courbes temporelles.

4.4 Algorithme de comparaison d'équipements

Dans cette section, nous décrivons la méthodologie que nous avons conçu afin de tenir compte des différentes contraintes d'application décrites dans la Section 4.3.

4.4.1 Description de l'algorithme proposé

L'algorithme de comparaison que nous proposons se déroule en quatre étapes :

1. Il faut tout d'abord définir le champ de l'analyse : quels procédés peuvent être regroupés ? Quels sont les équipements concernés ? L'analyse porte sur des procédés dont les étapes sont communes, mais tolérant des durées d'étapes pouvant être différentes, comme pour les modèles GTE (voir Section 3.1). De la même façon, les périodes de production sur lesquelles sont récoltées les données utilisées ne doivent pas contenir d'interventions de maintenance, au risque de

fausser les résultats par le mélange de données ayant des caractéristiques distinctes.

2. La deuxième étape consiste en l'alignement des points d'intérêts : cette étape se fait à deux échelles, en construisant des courbes représentatives pour chaque capteur de chaque chambre, et enfin en alignant les points d'intérêts de ces courbes entre eux.
3. La troisième étape est la réalisation de régressions linéaires entre capteurs identiques de chambres différentes. Les coefficients de détermination, nommés R^2, de ces régressions sont utilisés ensuite comme distance entre deux chambres pour un capteur donné.
4. Enfin, nous utilisons les distributions des R^2 pour proposer des valeurs limites, différentes pour chaque chambre et chaque capteur, permettant de classifier la chambre comme étant ou non atypique, et ce de façon non-supervisée.

L'algorithme est représenté sur la Figure 4.1.

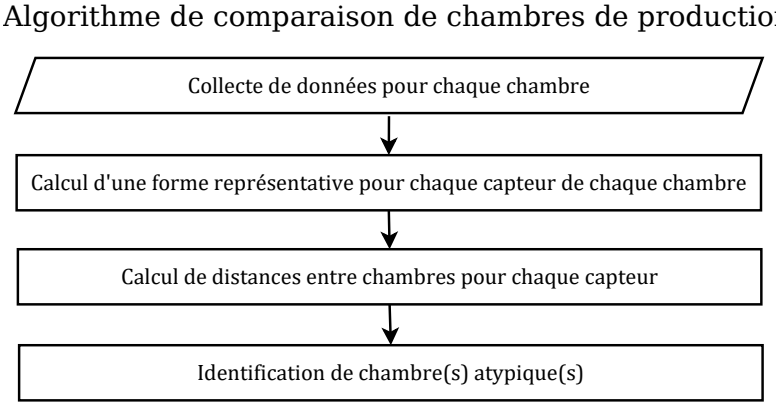

FIGURE 4.1 – Présentation générale de la méthode.

4.4.2 Définition du champ d'étude

La première étape à réaliser pour cet algorithme est de définir quelles chambres seront comparées, et relativement à quel(s) mode(s) de fonctionnement. Comme vu dans la Section 3.3, il est possible de réunir certains procédés qui ne varient que par la durée de certains steps lorsque l'on utilise l'algorithme DTW. Ainsi, les critères sont ici les mêmes que pour créer un modèle GTE : on sélectionne des chambres et des modes de fonctionnement correspondant à des recettes communes qui diffèrent au plus par

des durées de steps. Pour la suite, on utilise la notation $c \in \{1, ..., C\}$ pour désigner une chambre.

Pour chaque wafer traité, un ensemble de J capteurs collectent des données en temps réel, donnant ainsi J séries temporelles. Ces capteurs doivent être communs à toutes les chambres : si un capteur n'est présent que sur une partie des chambres, celui-ci est supprimé de l'analyse. Selon notre expérience, avoir les mêmes capteurs sur des équipements effectuant les mêmes procédés est une pratique industrielle logique, et donc la suppression de capteurs pour une telle analyse reste rare. Pour chaque chambre c, on collecte les données de I_c runs, obtenant ainsi $J \times I_c$ séries temporelles. Les runs doivent être sélectionnés sur une période sans intervention de maintenance, afin que les données représentent un seul état de l'équipement, et non la réunion de plusieurs états.

4.4.3 Pré-traitement de données : Dynamic Time Warping et formes moyennes représentatives pour chaque chambre

Cette étape a pour but d'obtenir pour chacune des C chambres comparées un nombre identique de $J \times K$ (J capteurs et K temps) points d'intérêts, partant de données où le nombre de runs est variable pour chaque chambre et où chaque run a une base de temps individuelle : pour chaque run i, il y a un nombre de temps K_i pouvant être différent. On précise que, comme dans le cas de la création d'un modèle GTE, toute étape de pré-traitement liée à une connaissance métier (par exemple, exclusion de données associées à certaines alarmes de l'équipement) peut s'appliquer au préalable.

Soit c une chambre. On dispose de I_c runs, et pour chaque run i, il y a $J \times K_{ci}$ points d'intérêts. La première étape est d'appliquer directement à cette chambre l'algorithme d'alignement décrit précédemment pour le GTE, à la Section 3.3. Cela signifie que l'on détermine une trajectoire de référence parmi les trajectoires disponibles associées à la recette la plus longue, pour ensuite utiliser DDTW afin d'aligner chaque run à la trajectoire de référence ainsi créée. La base de temps est alors commune à tous les runs de la chambre c, et on dispose de $I_c \times J \times K_c$ points d'intérêts. On construit alors une trajectoire moyenne pour la chambre. On note m_c cette trajectoire : m_c est obtenue en prenant la valeur moyenne sur les I_c runs à chaque instant $k \in \{1, .., K_c\}$. L'estimateur utilisé pour le calcul de la moyenne est la moyenne échantillonnale tronquée, c'est-à-dire la moyenne calculée sur un sous-ensemble de la distribution, dont un pour-

centage des plus grandes et plus petites valeurs ont été retirées. L'objectif ici est d'évaluer le comportement habituel de l'équipement : il faut donc éviter qu'une faible part de valeurs extrêmes influent grandement sur l'estimation. Cependant, l'utilisation d'une médiane peut masquer des défauts intermittents apparaissant sur un nombre significatif de runs. Dans notre pratique industrielle, nous supprimons 20% des données (10 pour chaque côté de la distribution) pour chaque estimation de moyenne. Ainsi, nous obtenons pour chaque chambre une forme constituée de $J \times K_c$ points d'intérêts, obtenue en replaçant sur la même base temporelle les données de chacun des runs étudiés, avant d'effectuer la moyenne de tous les points d'intérêts associés au même temps.

La dernière étape de pré-traitement est d'aligner les C formes obtenues, afin de rendre les points d'intérêts associés comparables. Nous faisons cette étape en sélectionnant comme référence la trajectoire moyenne m_c de la chambre c associée à la valeur K_c la plus grande, avant d'aligner, avec DDTW, les trajectoires restantes sur celle-ci. En effet, dans le cas d'une trajectoire de référence trop courte, ce sont les points d'intérêts les plus éloignés de la référence qui sont exclus pour effectuer l'alignement : ce sont justement ceux-là que l'on doit repérer dans notre cadre d'application.

L'ensemble étudié est désormais l'ensemble des trajectoires moyennes $\{m_1, ..., m_C\}$, contenant une trajectoire représentative pour chaque chambre. La dimension de cet ensemble est donc $C \times J \times K$, où C est le nombre de chambres, J le nombre de capteurs, et K le nombre de mesures correspondant à la plus longue de ces trajectoires avant alignement. Un élément m_{cjk} est la valeur moyenne pour la chambre c et le capteur j au temps k.

4.4.4 Écart entre deux chambres par régression linéaire

L'étape suivante de l'algorithme est de construire un indicateur de l'écart entre deux chambres.

Nous formulons l'hypothèse suivante, basée sur notre expérience des données de l'industrie du semi-conducteur :

Hypothèse. *La forme moyenne des signaux temporels est une représentation pertinente de l'état de fonctionnement des différents équipements.*

La forme moyenne des signaux est obtenue de la façon décrite dans la Section 4.4.3. Les différences de formes sont évaluées à travers des régressions linéaires. Ainsi les différences entre deux signaux couvertes par une transformation linéaire ne sont pas considérées comme caractéristiques de fautes : elles apparaissent du fait de réglages ou de pièces spécifiques, de

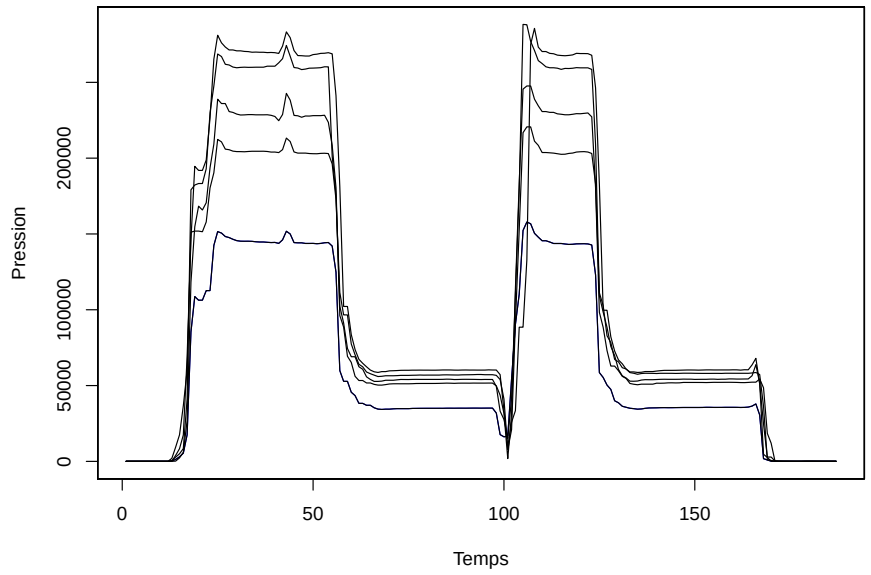

FIGURE 4.2 – Exemple de trajectoire moyenne d'un capteur de pression pour 5 équipements de dépôt : la forme est répétable, mais l'amplitude ne l'est pas.

façon non répétable, et il ne serait en général pas souhaitable de chercher à reproduire de tels ajustements. La Figure 4.2 montre un exemple de telles déformations. Sur ce graphique, la courbe moyenne pour un capteur récoltant la pression en un point précis est représentée pour 5 équipements de dépôt différents. Cette pression étant une résultante du procédé, sa valeur ne peut pas être ajustée directement : on observe la même forme pour ces 5 équipements, mais l'amplitude est différente pour chacune.

Pour un capteur j donné, on considère deux chambres c et d. On note m_{cj} les vecteurs contenant la série temporelle associée au capteur j pour la trajectoire m_c. On choisit m_{cj} comme étant la variable à expliquer de la régression, et m_{dj} comme unique variable explicative. Le résultat de la régression est $\hat{m}_{dj} = am_{cj} + b$, qui est une estimation de m_{dj} basée sur m_{cj}. Calculer $m_{dj} - \hat{m}_{dj}$ nous permet de mesurer des différences de forme. La régression linéaire basée sur les moindres carrés est une méthode très classique et très bien documentée. Les détails peuvent par exemple être trouvés dans [29].

Le R^2 est un indicateur très utilisé pour décrire la pertinence d'un modèle de régression linéaire, c'est-à-dire dans notre cas, pour évaluer si m_{dj}

peut correctement être décrit par une approximation linéaire $m_{dj} \approx am_{cj} + b$. D'un point de vue mathématique, le R^2 décrit la proportion de variance de la variable à expliquer qui peut être décrite par une transformation linéaire de la variable explicative. Sa valeur est entre 0 et 1. Dans notre algorithme, nous comparons la trajectoire d'un capteur j donné entre deux chambres, c et d. Le R^2 évalue alors la similarité des formes du capteur j pour les chambres c et d.

Pour rappeler la définition du R^2, nous définissons d'abord trois valeurs :

— La moyenne de la variable à expliquer : $\bar{m}_{dj} = \frac{1}{K}\sum_1^k m_{djk}$;

— La somme des carrés totale : $SS^t_{jcd} = \sum_1^k (m_{djk} - \bar{m}_{dj})^2$;

— La somme des carrés expliquée : $SS^e_{jcd} = \sum_1^k (\hat{m}_{djk} - \bar{m}_{dj})^2$.

Le R^2 entre les chambres c et d, pour le capteur j est alors défini par :

$$\mathrm{R}^2_{jcd} = \frac{SS^e_{jcd}}{SS^t_{jcd}}. \tag{4.3}$$

Le R^2 est symétrique : si les rôles de c et d sont inversés, sa valeur ne change pas. Cela assure à la fois la cohérence de cet indicateur pour notre utilisation, et une meilleure optimisation des calculs dans la pratique.

Pour une chambre donnée, on calcule le R^2 associé à la régression de chacun des capteurs contre toutes les autres chambres, et ce de façon individuelle. Ainsi, pour chaque capteur j, on construit une matrice symétrique dont la diagonale n'est constituée que de la valeur 1 (car la somme des carrés expliqués est égale à la somme des carrés totale dans le cas d'une régression d'un ensemble de valeurs sur lui-même).

4.4.5 Identification de chambres atypiques à travers la distribution des $\mathbf{R}^2$

À cette étape, une chambre c est définie, parmi les autres, par J ensembles $R_{cj} = \{\mathrm{R}^2_{jcd}\}_{d \neq c}$, chacun de $(C-1)$ valeurs. Pour identifier une chambre atypique, nous comparons ces valeurs, pour chaque capteur j, à la distribution globale des R^2 que nous aurions obtenu sans la chambre c, qui est $R_{\neq cj} = \{\mathrm{R}^2_{jde}\}_{d \neq c, e \neq c}$. Cela signifie que pour chaque capteur, nous considérons les valeurs indépendantes de la chambre c afin de constituer une distribution de référence. Notre hypothèse est qu'une chambre atypique aura au moins un capteur avec une forme moyenne différente, et ainsi une distribution de R^2 présentant des valeurs plus faibles que celles observées dans la distribution globale.

Nous adaptons à notre étude spécifique une méthode classique de recherche de valeurs aberrantes, à savoir la règle de l'écart inter-quartile, pro-

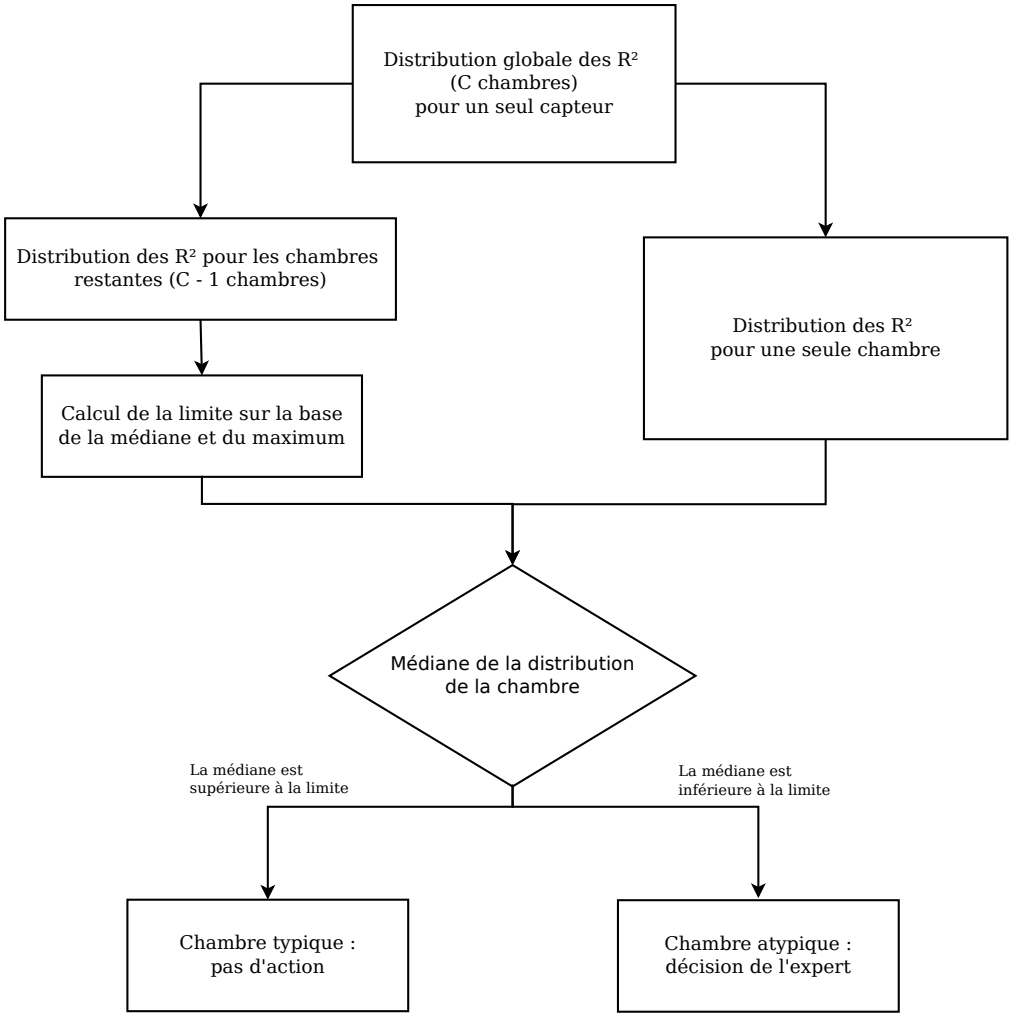

FIGURE 4.3 – L'algorithme de comparaison appliqué à chaque chambre pour chaque capteur.

posée par Tukey [73], détaillée à la Section 4.2.3. Cette méthode classifie comme suspecte tout observation hors de l'intervalle $[Med - 3IQR, Med + 3IQR]$ où Med est la médiane de la distribution, et IQR la distance entre le premier et le troisième quartile. Selon cette règle, une bonne observation se trouve proche de la médiane.

Notre cas est différent : une bonne valeur du R^2 est la plus proche possible de 1, car la valeur 1 signifie une superposition parfaite (à transformation linéaire près) des procédés des deux chambres. La même différence s'observe relativement à l'écart inter-quartile : la meilleure moitié de la distribution est sa moitié supérieure, c'est-à-dire les valeurs observées entre le maximum et la médiane. Ainsi, pour chaque capteur j, nous classifions comme aberrantes les valeurs qui sont en-dessous de $1 - 3(max(R_{\neq cj}) - med(R_{\neq cj}))$.

Un paramètre de limite l peut également être ajouté afin d'éviter un grand nombre de faux positifs. En effet, il n'est pas rare, notamment sur des procédés régulés, que les valeurs observées sur certains capteurs soient quasi-parfaitement répétées d'une chambre à une autre. Dans ce cas, des différences minimes sur une chambre donnée pourraient être détectées avec la limite proposée. Notre expérience industrielle nous a montré qu'une limite plus haute que $l = 0.8$ amène généralement à étudier des cas de différences non pertinentes. Nous définissons alors la limite L_{cj} :

$$L_{cj} = min(1 - 3(max(R_{\neq cj}) - med(R_{\neq cj})), l). \tag{4.4}$$

La limite L_{cj} est différente pour chaque chambre c, mais ne dépend pas directement de c : seule les chambres autres que c ont une influence sur sa valeur.

Pour chaque chambre c et chaque capteur j, on compare la médiane de R_{cj} avec la limite calculée (4.4). On vérifie ainsi que le capteur j pour la chambre c correspond à la même forme que celle observée majoritairement.

On peut ainsi définir l'ensemble A des chambres atypiques :

$$A = \{c \text{ pour lesquels } \exists j \text{ tel que } med(R_{cj}) < L_{cj}\}. \tag{4.5}$$

Remarque. *La limite L_{cj} peut être négative. Dans ce cas, soit la forme du capteur varie très largement d'une chambre à l'autre, ce qui empêche l'identification de chambres atypiques, soit le point de casse de l'estimation est atteint (voir Section 4.4.6). Lorsqu'au moins une chambre correspond à une limite négative, le capteur correspondant est signalé.*

La Figure 4.3 résume la procédure d'identification d'une chambre atypique pour un capteur donné.

4.4.6 Robustesse de l'algorithme

Dans cette partie, nous étudions le rapport entre le nombre C de chambres présentes dans l'analyse, et le nombre dont dépendent effectivement les limites individuelles calculées pour chaque chambre et chaque capteur.

Dans la Section 4.4.5, nous proposons pour une chambre c et un capteur j donnés de déterminer une limite de détection dépendant d'au plus la moitié des valeurs de R^2 indépendantes de c. Cela ne signifie pas pour autant que la limite L_{cj} calculée en (4.4) ne dépend que de la moitié des chambres restantes. En effet, puisque le R^2 est symétrique, et que la distribution étudiée n'est pas redondante mais correspondant à un seul R^2 par couple de chambres, la taille de l'ensemble dont dépend la valeur de la médiane correspond à un pourcentage de la distribution des chambres qui est variable suivant la valeur de C.

La question que l'on se pose est la suivante : combien de chambres atypiques peuvent être présentes dans la distribution sans que cela ne se répercute sur une seule valeur de limite L_{cj} ? Il s'agit ainsi de déterminer à combien de chambre correspond les 50% inférieurs de la distribution $R_{\neq cj}$, c'est-à-dire trouver le nombre maximal de chambres atypiques pouvant être détectées par l'algorithme.

Dans la Section 4.2.3, nous avons défini la notion de point de casse pour un estimateur : il s'agit du pourcentage maximal de valeurs arbitrairement grandes pouvant être présentes dans les données sans que l'estimation ne tende vers l'infini. Cette notion n'a pas directement de sens dans le cas de cet algorithme puisque nous étudions des valeurs situées entre 0 et 1. Pour évaluer la robustesse de l'estimation nous déterminons le nombre de chambres correspondant à des valeurs extrêmes à partir duquel l'estimation d'au moins un L_{cj} est affectée. On note n cette valeur qu'on appelle, par analogie, le point de casse de l'algorithme. Le point de casse pour l'estimation d'une seule valeur de L_{cj} est $n-1$ puisque si la chambre c est atypique alors la valeur L_{cj} n'est pas atteinte.

On a la relation suivante :

$$C-2+...+C-n \geq \frac{1}{2}\left(\frac{C(C-1)}{2}-(C-1)\right) > C-2+...+C-(n-1) \quad (4.6)$$

En effet, la valeur centrale correspond à la moitié de la taille de $R_{\neq cj}$, et donc sa partie entière supérieure correspond au nombre de valeurs susceptibles d'impacter la médiane. Le terme de droite correspond au nombre maximal de valeurs de R^2 qui peuvent être extrêmes (vers 0 dans notre méthodologie) sans que la médiane ne soit affectée : cela correspond donc à $C-2$ valeurs pour la première chambre atypique, $C-3$ valeurs de plus

lorsqu'on rajoute une deuxième chambre atypique, ainsi de suite, jusqu'à $C-(n-1)$ pour la $n-2^{\text{e}}$ chambre atypique. Le terme de gauche correspond ainsi de façon analogue à l'ajout d'une $n-1^{\text{e}}$ chambre supplémentaire au terme de droite de telle façon à dépasser la médiane, définissant ainsi le point de casse de l'estimation. Pour trouver la valeur de n, il suffit ainsi de trouver la valeur de n vérifiant cette inéquation sachant la valeur de C.

On a ainsi le système d'inéquations suivant :

$$\begin{cases} C-n+C-(n-1)+C-2-\frac{C^2}{4}+\frac{3C}{4}-\frac{1}{2}\geq 0 \\ C-(n-1)+C-2-\frac{C^2}{4}+\frac{3C}{4}-\frac{1}{2}<0 \end{cases}$$

Et donc ainsi, en résolvant en n :

$$n=\left\lfloor C+\frac{1}{2}-\sqrt{\frac{C^2}{2}-\frac{3C}{2}+\frac{5}{4}}\right\rfloor, \tag{4.7}$$

où $\lfloor a \rfloor$ désigne la partie entière inférieure de a.

Ainsi, l'introduction de n chambres atypiques dans les données porte atteinte à toutes les estimations de L_{cj} et rend donc la comparaison impossible : c'est le point de casse de l'algorithme. Dans le cas de $n-2$ chambres atypiques, aucune estimation de L_{cj} n'est affectée, alors que dans le cas de $n-1$ chambres atypiques, seules les limites des chambres typiques sont affectées (et donc les chambres atypiques peuvent quand même être détectées).

4.5 Exemples d'application à STMicroelectronics Rousset

Cette section présente trois exemples de comparaison de chambres basés sur des défauts observés à STMicroelectronics. Le premier exemple montre un exemple de défaut d'un régulateur de débit de gaz trouvé parmi un parc de 13 chambres, et le deuxième exemple montre la détection d'une fuite sur une chambre parmi un autre parc de 13 chambres. Le troisième et dernier exemple illustre le cas d'un re-réglage à effectuer, avec impact faible sur les produits, déterminé par la comparaison de 4 chambres.

4.5.1 Panne d'un régulateur de débit

Dans cet exemple, nous traitons de la détection d'une pièce défectueuse d'un équipement par comparaison avec l'ensemble de son atelier. Ces équipements effectuent des procédés de dépôt de type "Physical Vapor Deposition" (PVD). Il s'agit de procédés durant lesquels de la matière condensée

est transformée en vapeur avant d'être à nouveau condensée sous forme de couche mince sur les wafers.

Pour cet exemple, nous avons un ensemble de données correspondant à 13 chambres effectuant la même recette. Ces chambres sont surveillées par 7 ou 8 capteurs : le capteur supplémentaire de certaines de ces chambres est supprimé afin de pouvoir mener une analyse globale commune basée sur l'algorithme présenté à la Section 4.4.1. Un ensemble de données correspondant à 50 runs pour chacune de ces chambres a été récolté, comme expliqué à la Section 4.4.2. Les données des 650 runs sont pré-traitées, afin d'obtenir la matrice des trajectoires moyennes, comme expliqué à la Section 4.4.3. On obtient alors 144 temps de mesure pour chacun des capteurs moyens. En utilisant des régressions linéaires (Section 4.4.4), on détermine toutes les valeurs de R^2 nécessaires, obtenant alors une distribution globale de $\frac{13(13-1)}{2} = 78$ valeurs par capteur, comme vu à la Section 4.4.4 (on étudie des matrices symétriques 13×13 dont on ignore la diagonale).

Pour chacune des 13 chambres et chacun des 7 capteurs, les limites correspondantes sont calculées, en utilisant l'algorithme décrit dans la Section 4.4.5, auxquelles les médianes des distributions des R^2 sont comparées. Le paramètre l est fixé à 0.8 comme évoqué à la Section 4.4.5. Dans le cas de 13 chambres, le point de casse de la méthode, défini dans la Section 4.4.6, est de 5. C'est-à-dire que dans le cas où il y a 3 chambres atypiques ou moins, aucune estimation de limite n'est atteinte.

Seule une chambre montre une valeur atypique, et ce pour un seul capteur : un flux de dihydrogène. La Table 4.1 montre en détail les valeurs de R^2 observées pour toutes les chambres pour ce capteur. La ligne et la colonne correspondant à la chambre anormale sont surlignées. La Figure 4.4 montre en rouge la courbe anormale, qui est comparée aux 12 courbes en bleu, correspondant aux autres chambres. Le flux de gaz était délivré en retard sur une grande partie des wafers traités par la chambre : le régulateur a alors du être remplacé.

La Table 4.2 contient les valeurs de tous les R^2 médians : une valeur par chambre et par capteur, arrondie au centième, avec la valeur détectée en gras. Chaque valeur est comparée à la limite correspondante, qui est ici le paramètre l dans tous les cas : 0.8. On retrouve cette configuration quand les capteurs sélectionnés dans la stratégie de collecte représentent tous un aspect répétable du procédé. Il s'agit ici d'un équipement avec seulement 7 capteurs, du fait de contraintes de réseaux sur ce parc d'équipements : seules les valeurs physiques les plus importantes sont récoltées, il n'y pas de place supplémentaire pour des paramètres annexes au procédé. Il ne s'agit pas du cas le plus fréquent, et on illustre cela dans l'exemple suivant, Section 4.5.2, où les valeurs de limite varient selon les capteurs.

Chamber	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
Chamber													
C1	1.0	.58	.58	.58	.56	.56	.57	.59	.59	.59	.59	.58	.58
C2	.58	1.0	1.0	1.0	.98	.98	.99	1.0	1.0	1.0	1.0	1.0	1.0
C3	.58	1.0	1.0	1.0	.98	.98	.99	1.0	1.0	1.0	1.0	1.0	1.0
C4	.58	1.0	1.0	1.0	.98	.98	.99	1.0	1.0	1.0	1.0	1.0	1.0
C5	.56	.98	.98	.98	1.0	.97	.99	.99	.99	.99	.98	.98	.99
C6	.56	.98	.98	.98	.97	1.0	.99	.98	.98	.98	.98	.98	.98
C7	.57	.99	.99	.99	.99	.99	1.0	1.0	.99	.99	.99	.99	.99
C8	.59	1.0	1.0	1.0	.99	.98	1.0	1.0	1.0	1.0	1.0	1.0	1.0
C9	.59	1.0	1.0	1.0	.99	.98	.99	1.0	1.0	1.0	1.0	.99	1.0
C10	.59	1.0	1.0	1.0	.99	.98	.99	1.0	1.0	1.0	1.0	1.0	1.0
C11	.59	1.0	1.0	1.0	.98	.98	.99	1.0	1.0	1.0	1.0	1.0	1.0
C12	.58	1.0	1.0	1.0	.98	.98	.99	1.0	.99	1.0	1.0	1.0	.99
C13	.58	1.0	1.0	1.0	.99	.98	.99	1.0	1.0	1.0	1.0	.99	1.0

TABLE 4.1 – Exemple 4.5.1 : valeurs du R^2 pour le flux de dihydrogène (capteur S2) pour les 13 chambres.

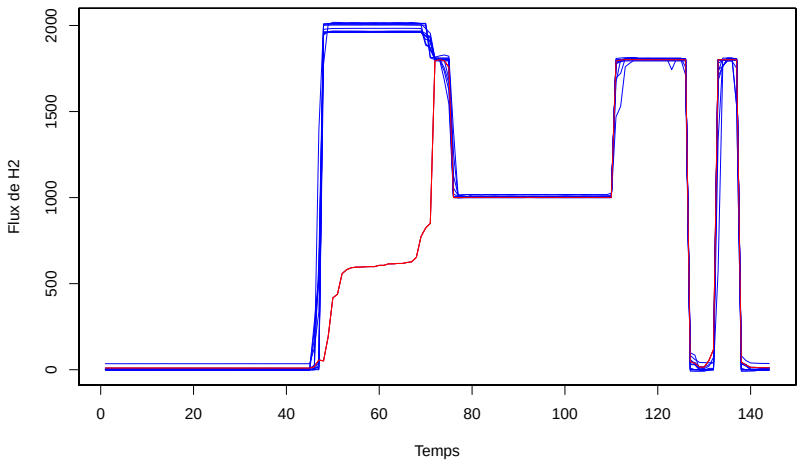

FIGURE 4.4 – Un flux de dihydrogène représenté pour 13 chambres, dont une atypique.

Sensor	S1	S2	S3	S4	S5	S6	S7
Chamber							
C1	1.0	**.58**	.94	.99	.99	.99	1.0
C2	1.0	1.0	.98	1.0	.99	.99	1.0
C3	1.0	1.0	.99	1.0	.99	.99	1.0
C4	1.0	1.0	.98	1.0	.99	.99	1.0
C5	.94	.98	.98	.96	.97	.94	1.0
C6	.96	.98	.94	.97	.99	.82	1.0
C7	.96	.99	.92	.99	.95	.87	1.0
C8	1.0	1.0	.99	.99	.99	.99	1.0
C9	1.0	1.0	.99	1.0	.99	.99	1.0
C10	.98	1.0	.99	1.0	.99	.99	1.0
C11	1.0	1.0	.97	.99	.98	1.0	1.0
C12	1.0	1.0	.98	1.0	.97	.99	1.0
C13	1.0	1.0	.99	1.0	.99	.99	1.0

TABLE 4.2 – Exemple 4.5.1 : R^2 médian pour chaque chambre et chaque capteur. La limite pour chaque valeur est 0.8.

4.5.2 Détection de la fuite de l'exemple 3.6.2 par comparaison de chambres

Dans la Section 3.6.2, nous étudions le cas d'une fuite apparaissant suite à une opération de maintenance, sur un équipement dédié au dépôt CVD. Par la méthodologie GTE, nous détectons la fuite en utilisant les données de 20 runs pour le test de retour de maintenance, portant sur la variabilité inter-runs. Nous proposons maintenant de détecter cette fuite en utilisant cette fois-ci la méthodologie de comparaison de chambres, sur la base des équipements similaires disponibles, c'est-à-dire en ne se basant non plus sur les variabilités, mais sur les valeurs moyennes, ignorées par le GTE. Ainsi, l'algorithme de comparaison de chambres permet de détecter une faute consécutive à une maintenance qui ne se manifesterait pas à travers la variabilité, permettant alors de valider un retour de maintenance suivant deux aspects différents des données.

Dans cet exemple, nous avons collecté les données de 13 chambres utilisant la même recette. Ces chambres ont entre 12 et 15 capteurs, et nous restreignons l'étude aux 12 capteurs communs afin d'appliquer l'algorithme décrit à la Section 4.4.1. Un ensemble de données correspondant à 50 runs est sélectionné pour chaque chambre, comme vu Section 4.4.2. Le pré-traitement de ces données se fait comme expliqué à la Section 4.4.3. On a alors 190 instants de mesure pour chaque capteur moyen calculé.

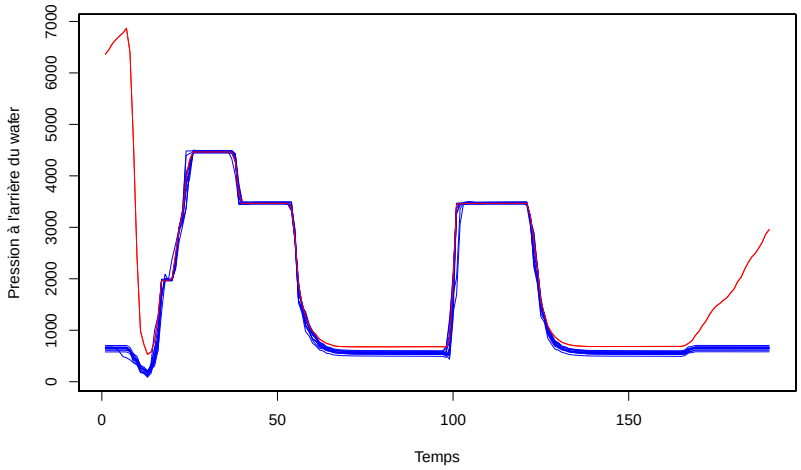

FIGURE 4.5 – La chambre atypique tracée face aux 12 autres, pour une variable de pression.

En utilisant des régressions linéaires (Section 4.4.4), on détermine toutes les valeurs de R^2 nécessaires, obtenant alors une distribution globale de $\frac{13(13-1)}{2} = 78$ valeurs par capteur.

Pour chaque chambre et chaque capteur, la limite est calculée suivant l'algorithme de la Section 4.4.5, à laquelle la valeur médiane des R^2 est comparée. Le paramètre l est fixé à 0.8 comme suggéré à la Section 4.4.5. Seule une chambre présente une valeur atypique, pour un seul capteur : la 3^{e} chambre pour le 1^{er} capteur, qui est une variable de pression. La Figure 4.5 montre en rouge la courbe moyenne anormale, comparée aux courbes bleues qui représentent les 12 autres chambres. La courbe rouge montre une fuite dans la chambre : on retrouve la fuite détectée dans l'exemple 3.6.2, et aucune autre détection n'est faite.

La Table 4.3 contient les valeurs des R^2 médians pour 3 capteurs : une valeur par chambre et par capteur, arrondie au centième, avec la valeur détectée en gras. Chaque valeur est comparée à la limite correspondante. Le premier capteur, "Pression1", est le capteur sur lequel la faute est détectée. Le capteur nommé "Puissance2" montre la possibilité d'avoir une limite sous le paramètre l. Le capteur "Pression3" n'est pas pris en compte dans cette comparaison : les valeurs du R^2 varient de façon très large, entraînant une limite négative suivant la formule 4.4.

Sensor	Pression1	Puissance2	Pression3
Chamber			
C1	1.0 | .8	.90 | .55	.69 | -.3
C2	1.0 | .8	.88 | .56	.46 | .05
C3	**.44** | **.8**	.79 | .59	.86 | -.3
C4	1.0 | .8	.91 | .49	.74 | -.3
C5	1.0 | .8	.83 | .57	.49 | .02
C6	1.0 | .8	.74 | .61	.73 | -.3
C7	1.0 | .8	.86 | .55	.73 | -.3
C8	.99 | .8	.86 | .53	.49 | .02
C9	.99 | .8	.88 | .50	.45 | .02
C10	1.0 | .8	.77 | .57	.85 | -.4
C11	1.0 | .8	.82 | .55	.76 | -.3
C12	1.0 | .8	.87 | .50	.55 | -.2
C13	1.0 | .8	.87 | .56	.78 | -.3

TABLE 4.3 – Exemple 4.5.2 : médiane des valeurs du R^2 et limite pour chacune des chambres pour 3 capteurs.

4.5.3 Nettoyage inefficace d'un équipement

Dans cet exemple, nous étudions le cas de comparaison de 4 chambres, sur une recette de nettoyage d'un équipement de retrait résine : il n'y a pas de wafer traité pendant la recette. L'objectif d'une telle recette est de réduire le dépôt de matériaux qui ont lieu dans l'équipement pendant les réactions chimiques de production qui sont faites sur les wafers. Cet exemple illustre tout d'abord l'utilisation de l'algorithme dans le but d'effectuer un re-réglage qui n'a pas d'impact critique sur les produits. Les deux précédents exemples montrent de véritables fautes de l'équipement détecté, où il est nécessaire d'arrêter l'équipement pour limiter les pertes, ce qui n'est pas le cas ici. Enfin, cet exemple montre également l'utilisation de l'algorithme dans un cas limite, avec un parc d'équipements réduit. En effet, pour 4 équipements, le point de casse défini à la Section 4.4.6 est de 2. La présence d'une seule chambre atypique perturbe une partie des limites calculées, et deux chambres atypiques empêchent le fonctionnement de l'algorithme : dans ce cas on ne dispose pas d'une majorité absolue d'équipements fonctionnant de façon correcte, ce qui est nécessaire dans notre étude non-supervisée.

Chacune des 4 chambres dispose de 27 capteurs récoltant des données pour cette recette de nettoyage. On applique ainsi l'algorithme décrit à la Section 4.4.1 sur un ensemble de données issu de 50 runs pour chaque

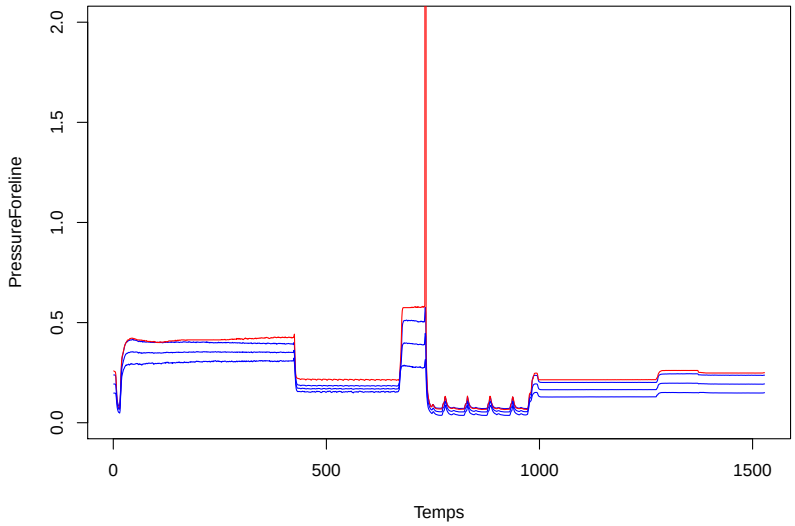

FIGURE 4.6 – La chambre atypique tracée face aux 12 autres, pour une variable de pression.

chambre, comme vu Section 4.4.2. Le pré-traitement de ces données se fait comme expliqué à la Section 4.4.3. On a alors 1528 instants de mesure pour chaque capteur moyen calculé. En utilisant des régressions linéaires (Section 4.4.4), on détermine toutes les valeurs de R^2 nécessaires, obtenant alors une distribution globale de $\frac{4(4-1)}{2} = 6$ valeurs par capteur.

Pour chaque chambre et chaque capteur, la limite est calculée suivant l'algorithme de la Section 4.4.5, à laquelle la valeur médiane des R^2 est comparée. Le paramètre l est fixé à 0.8 comme suggéré à la Section 4.4.5. Seule une chambre présente une valeur atypique, pour un seul capteur : la 2^e chambre pour le 21^e capteur, qui est une variable de pression. La Figure 4.6 montre en rouge la courbe anormale, comparée aux courbes bleues qui représentent les 3 autres chambres. La courbe rouge montre une désynchronisation : la pression augmente nettement avant stabilisation, car un gaz est encore injecté en même temps qu'il est évacué. L'injection du gaz doit normalement être stoppée avant la phase dite de "pompage" permettant la chute de pression. Il en résulte un risque de nettoyage incorrect à cause d'un excès de matériaux réactifs. La distribution des R^2 pour la variable de pression détectée est présentée dans le Tableau 4.4. La ligne et la colonne des valeurs dépendant de la chambre 2, en défaut, est grisée.

	C1	C2	C3	C4
C1	1.00	0.18	0.93	0.99
C2	0.18	1.00	0.14	0.17
C3	0.93	0.14	1.00	0.98
C4	0.99	0.17	0.98	1.00

TABLE 4.4 – Exemple 4.5.3 : distribution des valeurs du R^2 pour le capteur détecté.

4.6 Conclusion

Dans ce chapitre, nous proposons de compléter l'algorithme supervisé décrit au Chapitre 3 en employant les données d'équipements similaires. Il s'agit d'une contribution importante de cette thèse. Cette approche permet de détecter des changements de valeurs moyennes anormaux suite à une maintenance, et également d'améliorer la cohérences de la production en identifiant des différences anormales entre équipements réalisant les mêmes procédés.

Aucune méthodologie appliquée à ce problème particulier n'est disponible dans la littérature. Nous proposons d'effectuer une comparaison de forme entre signaux représentatifs pour les différents capteurs communs des équipements étudiés. Cette étude est non-supervisée, car aucun équipement de référence ne peut être sélectionné. L'algorithme est basé sur l'étude des R^2 obtenus par régression linéaire entre tous les couples d'équipements. Les équipements présentant des R^2 anormaux pour un ou plusieurs capteurs sont identifiés, afin de permettre leur réparation, ou re-réglage dans le cas de différences n'entraînant pas de défaillance.

Nous montrons l'efficacité pratique de cet algorithme à partir de trois exemples d'utilisation en milieu industriel : deux cas de détection d'équipement défaillant, nécessitant réparation, et un cas d'équipement atypique ne nécessitant qu'un re-réglage. Cette méthodologie peut être appliquée indépendamment du GTE, mais seule la combinaison des deux approches permet une vision complète des données.

Conclusion et perspectives

Nous avons présenté dans ces travaux une méthodologie de détection en temps réel pour l'industrie du semi-conducteur, tenant compte des nombreuses contraintes s'y appliquant. En effet, les approches appliquées aujourd'hui dans l'industrie s'avèrent trop limitées pour assurer une surveillance efficace des équipements. Les méthodes disponibles dans la littérature permettent l'amélioration de certains aspects, mais négligent le problème de la maintenance fréquente des équipements.

Nous avons proposé dans cette thèse une méthodologie axée autour de deux formes d'apprentissage : supervisé, à l'échelle d'un seul équipement et non-supervisé, à l'échelle de groupes d'équipements réalisant les mêmes procédés.

Détection supervisée : Gaussian Time Error

Dans le Chapitre 3, nous avons introduit la méthodologie Gaussian Time Error. Cet algorithme a pour but de détecter des équipements défaillants sur la base des runs individuels de celui-ci. Un modèle initial est appris de façon supervisée sur une période de temps où l'équipement fonctionne correctement, et ce modèle est partiellement adapté au fil des maintenances de l'équipement. Ce modèle est constitué par projection des données physiques dans un espace où les variables sont décorrélées ; celles-ci sont modélisées suivant des lois normales avec matrice de covariance diagonale à chaque temps du procédé. Un test est réalisé pour chaque run, avec l'indice de détection GTE, et pour chaque maintenance, sur la base d'un échantillon des premiers runs effectués.

Nous avons montré à travers des exemples industriels la capacité de cet indice à correctement distinguer des fautes, pouvant affecter soit un nombre restreint de runs soit tous les runs à partir d'une certaine date. Les maintenances correctes et incorrectes sont également classifiées avec succès.

Détection non-supervisée : comparaison d'équipements

Dans le Chapitre 4, nous montrons qu'il est possible de compléter la méthodologie GTE en ajoutant une étape parallèle de classification non-supervisée. Un équipement seul ne peut pas constituer une référence pour tester les valeurs moyennes modifiées dans le modèle GTE suite à une maintenance. Faire intervenir les données d'équipements similaires permet alors de distinguer d'éventuelles anomalies. La méthodologie que nous proposons s'appuie sur la comparaison de la forme des courbes observées pour les différents capteurs des équipements d'un même atelier. Les équipements se distinguant des autres pour certains capteurs sont signalés afin de permettre un re-réglage ou une réparation.

Cette méthode permet de valider la cohérence d'un groupe d'équipement ainsi que de compléter la validation des maintenances effectuée avec le GTE. Son efficacité est montrée sur des exemples industriels.

Perspectives

Les nécessités industrielles et les difficultés inhérentes à la détection de fautes font que les perspectives de ce travail demeurent très vastes.

Une première perspective est l'élaboration d'une méthodologie permettant de distinguer les erreurs de synchronisation des signaux des véritables fautes. En effet, nous ne proposons ici qu'un filtrage des détections suivant leur répétition sur plusieurs runs. Une telle approche s'avère limitée pour le cas de fautes ne se manifestant qu'une seule fois dans les données : celles-ci ne sont jamais détectées. Il s'agirait alors de caractériser les erreurs d'alignement, par exemple en comparant les données alignées aux données non-alignées.

Nous n'avons pas proposé de méthodologie de détection pour les données d'équipements d'implantation, dont nous évoquons l'alignement temporel dans la Section 3.3.3. La particularité de ces données est la très grande stabilité de la plupart des signaux à l'échelle de plusieurs runs : l'hypothèse selon laquelle des corrélations physiques globales sont observables à l'échelle d'un run n'est pas vérifiée, car ces corrélations peuvent varier significativement d'un run à l'autre. Plusieurs runs consécutifs présentent une seule et unique valeur pour un paramètre physique donné, et au moment d'un re-réglage automatique de l'équipement, cette valeur change pour les runs suivants, de façon difficilement prévisible. D'autres approches doivent donc être proposées pour de tels équipements, éventuellement par l'emploi de données différentes de celles récoltées actuellement.

La comparaison d'équipements pourrait également être améliorée. En

effet, une comparaison basée seulement sur la forme des signaux fait que certains paramètres, comme l'amplitude globale, sont négligés, et les différences d'amplitude locales faibles relativement à l'amplitude globale ne sont pas détectées. Intégrer ces problématiques a entraîné dans notre cas un nombre extrêmement élevé de fausses alarmes : d'autres approches doivent être élaborées, peut-être en faisant intervenir des données issues d'autres sources.

Enfin, dans ces travaux nous n'avons étudié que la santé des équipements. L'impact sur la production des fautes observées n'est pas directement évalué : les modèles sont définis relativement aux équipements eux-mêmes. L'évaluation de la qualité des produits n'a qu'un impact indirect, à travers le choix de l'ensemble d'apprentissage. Pour l'industriel, c'est la qualité finale des wafers qui est l'information la plus importante : maintenir les équipements n'est qu'un moyen vers cette finalité. Un indice de santé des produits est ainsi, à terme, le véritable objectif de tout travail sur l'évaluation de l'état des équipements.

Bibliographie

[1] J. Assfalg, H. Kriegel, P. Kroger, P. Kunath, A. Pryakhin, and M. Renz. Similarity search on time series based on threshold queries. In *International Conference on Extending Database Technology*, pages 276–294. Springer, 2006.

[2] D. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[3] Y. Cai, P. Ricardo, C. Jen, and K. Chou. Application of svm to predict membrane protein types. *Journal of Theoretical Biology*, 226(4) :373–376, 2004.

[4] G. Cawley and N Talbot. Preventing over-fitting during model selection via bayesian regularisation of the hyper-parameters. *Journal of Machine Learning Research*, 8(Apr) :841–861, 2007.

[5] G. Cawley and N. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(Jul) :2079–2107, 2010.

[6] J. Chen and K. Liu. On-line batch process monitoring using dynamic pca and dynamic pls models. *Chemical Engineering Science*, 57(1) :63–75, 2002.

[7] L. Chen and R. Ng. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 792–803. VLDB Endowment, 2004.

[8] L. Chen, M. Ozsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502. ACM, 2005.

[9] W. Chen, S. Hsu, and H. Shen. Application of svm and ann for intrusion detection. *Computers & Operations Research*, 32(10) :2617–2634, 2005.

[10] Y. Chen, M. Nascimento, B. Ooi, and A. Tung. Spade : On shape-based pattern detection in streaming time series. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 786–795. IEEE, 2007.

[11] L. Chiang, E. Russell, and R. Braatz. *Fault detection and diagnosis in industrial systems*. Springer Science, 2000.

[12] R. Cook. Detection of influential observation in linear regression. *Technometrics*, 19(1) :15–18, 1977.

[13] T. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3) :326–334, 1965.

[14] G. Das, D. Gunopulos, and H. Mannila. Finding similar time series. In *European Symposium on Principles of Data Mining and Knowledge Discovery*, pages 88–100. Springer, 1997.

[15] E. Del Castillo and A. Hurwitz. Run-to-run process control : Literature review and extensions. *Journal of Quality Technology*, 29(2) :184, 1997.

[16] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data : experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2) :1542–1552, 2008.

[17] D. Donoho and P. Huber. The notion of breakdown point. *A festschrift for Erich L. Lehmann*, 157184, 1983.

[18] A. Elewa. *Morphometrics : applications in biology and paleontology*, volume 14. Springer Science & Business Media, 2004.

[19] L. Elshenawy, S. Yi, A. Naik, and S. Ding. Efficient Recursive Principal Component Analysis algorithms for process monitoring. *Industrial & Engineering Chemistry Research*, 49(1) :252–259, 2009.

[20] D. Erdogmus and H. Peddaneni. Recursive Principal Components Analysis using eigenvector matrix perturbation. *EURASIP Journal on Advances in Signal Processing*, 2004(13) :1–8, 2004.

[21] B. Everitt, D. Stahl, M. Leese, and S. Landau. Cluster analysis. 2011.

[22] J. Faucher. Pratique de l'amdec. *Edition DUNOD, Paris*, 2004.

[23] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[24] A. Fu, E. Keogh, L. Lau, C. Ratanamahatana, and R. Wong. Scaling and time warping in time series querying. *The VLDB Journal - The International Journal on Very Large Data Bases*, 17(4) :899–921, 2008.

[25] F. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1) :1–21, 1969.

[26] H. Hotelling. Analysis of a complex of statistical variables into principal components. *J. of educational psychology*, 1933.

[27] P. Huber. *Robust statistics*. Springer, 2011.

[28] S Joe Qin. Statistical process monitoring : basics and beyond. *Journal of chemometrics*, 17(8-9) :480–502, 2003.

[29] R. Johnson and D. Wichern. *Applied Multivariate Statistical Analysis*. Pearson Prentice Hall, 2007.

[30] I. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[31] P. Karl. La logique de la découverte scientifique. *Paris Payot*, 1973.

[32] A. Kassidas, J. MacGregor, and Taylor P. Synchronization of batch trajectories using Dynamic Time Warping. *AIChE J.*, 44(4) :864–875, 1998.

[33] J. Keats and D. Montgomery. *Statistical Applications in process control*. CRC Press, 1996.

[34] E. Keogh and M. Pazzani. Derivative Dynamic Time Warping. In *SDM*, volume 1, pages 5–7. SIAM, 2001.

[35] M. Khelil, M. Boudraa, A. Kechida, and R. Drai. Classification of defects by the svm method and the principal component analysis (pca). *Enformatika*, 9 :226–231, 2005.

[36] W. Ku, R. Storer, and C. Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and intelligent laboratory systems*, 30(1) :179–196, 1995.

[37] F. Lauer and G. Bloch. Méthodes svm pour l'identification. In *Journées Identification et Modélisation Expérimentale (JIME'2006)*, page CDROM, 2006.

[38] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata. Detecting outliers : Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4) :764–766, 2013.

[39] B. Lu. *Improving process monitoring and modeling of batch-type plasma etching tools*. PhD thesis, University of Texas at Austin, 2015.

[40] J. Marino, F. Rossi, M. Ouladsine, and J.Pinaton. Gaussian Time Error : a fault detection index for semiconductor processes. In *American Control Conference (ACC)*. IEEE, 2016.

[41] J. Marino, F. Rossi, M. Ouladsine, and J. Pinaton. A maintenance-adaptive fault detection framework for semiconductor processes using Gaussian Time Error. *Submitted*, 2016.

[42] J. Marino, F. Rossi, M. Ouladsine, and J. Pinaton. Unsupervised semiconductor chamber matching based on shape comparison. In *IFAC World Congress*. IFAC, 2016.

[43] J. Marino, F. Rossi, M. Ouladsine, and J. Pinaton. Gaussian Time Error : a statistical test for time series under time-shifts. *Submitted*, 2017.

[44] J. Marino, F. Rossi, M. Ouladsine, and J. Pinaton. A real-time supervised and unsupervised fault detection algorithm for semiconductor processes. *Submitted*, 2017.

[45] T. McLellan and J. Endler. The relative success of some methods for measuring and describing the shape of complex objects. *Systematic Biology*, 47(2) :264–281, 1998.

[46] L. Meng, W. Miao, and W. Chunguang. Research on svm classification performance in rolling bearing diagnosis. In *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, volume 3, pages 132–135. IEEE, 2010.

[47] B. Mnassri, M. El Adel, M. Ouladsine, and B. Ananou. Selection of the number of principal components based on the fault reconstruction approach applied to a new combined index. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 3307–3312. IEEE, 2010.

[48] L. Mönch, J. Fowler, S. Dauzère-Pérès, S. Mason, and O. Rose. A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. *Journal of Scheduling*, 14(6) :583–599, 2011.

[49] N. Nielsen, J. Carstensen, and J. Smedsgaard. Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping. *Journal of Chromatography A*, 805(1) :17–35, 1998.

[50] P. Nomikos and J.F. MacGregor. Monitoring of batch processes using Multi-way Principal Components Analysis. *AIChE J.*, 40(8) :1361–1375, 1994.

[51] J. Oakland. *Statistical process control*. Routledge, 2007.

[52] E. Osuna, R. Freund, and F. Girosit. Training support vector machines : an application to face detection. In *Computer vision and pattern recognition, 1997. Proceedings., 1997 IEEE computer society conference on*, pages 130–136. IEEE, 1997.

[53] K. Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and J. of Science*, 1900.

[54] F.O. Petitjean, A. Ketterlin, and P. Ganarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44 :678, 2011.

[55] L. Puggini, J. Doyle, and S. McLoone. Fault detection using random forest similarity distance. *IFAC*, 48(21) :583–588, 2015.

[56] G. Ramamoorthy. Gartner says worldwide semiconductor revenue forecast to grow 7.2 percent in 2017. *Gartner Press Releases*, 2017.

[57] P. Ribot. *Vers l'intégration diagnostic/pronostic pour la maintenance des systèmes complexes*. PhD thesis, Université Paul Sabatier-Toulouse III, 2009.

[58] F. Rohlf and D. Slice. Extensions of the procrustes method for the optimal superimposition of landmarks. *Systematic Biology*, 39(1) :40–59, 1990.

[59] B. Rosner. Percentage points for a generalized esd many-outlier procedure. *Technometrics*, 25(2) :165–172, 1983.

[60] D. Rosso. Global semiconductor sales reach 339 billion dollars in 2016. *Semiconductor.org*, 2017.

[61] P. Rousseeuw and C. Croux. Explicit scale estimators with high breakdown point. *L1-Statistical analysis and related methods*, 1 :77–92, 1992.

[62] P. Rousseeuw and A. Leroy. *Robust regression and outlier detection*, volume 589. John wiley & sons, 2005.

[63] G. Saporta. *Probabilités, analyse des données et statistique*. Editions Technip, 2006.

[64] H. Scheffe. *The analysis of variance*, volume 72. John Wiley & Sons, 1999.

[65] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. In *Advances in neural information processing systems*, pages 582–588, 2000.

[66] K. Skinner, D. Montgomery, G. Runger, J. Fowler, D. McCarville, T. Rhoads, and J. Stanley. Multivariate statistical methods for modeling and analysis of wafer probe test data. *IEEE transactions on semiconductor manufacturing*, 15(4) :523–530, 2002.

[67] M. Stegmann and D. Gomez. A brief introduction to statistical shape analysis. *Informatics and mathematical modelling, Technical University of Denmark, DTU*, 15(11), 2002.

[68] G. Szekely and M. Rizzo. Hierarchical clustering via joint between-within distances : Extending ward's minimum variance method. *Journal of classification*, 22(2) :151–183, 2005.

[69] D. Tax and R. Duin. Support vector data description. *Machine learning*, 54(1) :45–66, 2004.

[70] A. Thieullen. *Méthodologie pour la détection de défaillances des procédés de fabrication par Analyse en Composantes Principales*. PhD thesis, Aix-Marseille Université, 2014.

[71] A. Thieullen, M. Ouladsine, and J. Pinaton. Application of PCA for efficient multivariate FDC of semiconductor manufacturing equipment. In *Advanced Semicond. Manuf. Conf.*, pages 332–337. IEEE, 2013.

[72] G. Tietjen and R. Moore. Some grubbs-type statistics for the detection of several outliers. *Technometrics*, 14(3) :583–597, 1972.

[73] J. Tukey. *Exploratory data analysis*. Reading, Mass, 1977.

[74] G. Upton and I. Cook. *Understanding statistics*. Oxford University Press, 1996.

[75] A. Van Nederkassel, M. Daszykowski, P. Eilers, and Y. Vander Heyden. A comparison of three algorithms for chromatograms alignment. *Journal of Chromatography A*, 1118(2) :199–210, 2006.

[76] H. Wang and M. Yao. Fault detection of batch processes based on multivariate functional kernel principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 149 :78–89, 2015.

[77] J. Ward and H. Joe. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301) :236–244, 1963.

[78] J. Westerhuis, T. Kourti, and J. MacGregor. Analysis of multiblock and hierarchical pca and pls models. *Journal of chemometrics*, 12(5) :301–321, 1998.

[79] D.J. Wheeler. *Normality and the Process Behavior Chart*. Statistical Process Controls, Incorporated, 2000.

[80] S. Wold, P. Geladi, K. Esbensen, and J. Öhman. Multi-way Principal Components and PLS analysis. *J. of chemometrics*, 1(1) :41–56, 1987.

[81] S. Wold, M. Sjöström, and L. Eriksson. Pls-regression : a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, 58(2) :109–130, 2001.

[82] K. Yang and C. Shahabi. A multilevel distance-based index structure for multivariate time series. In *Temporal Representation and Reasoning, 2005. TIME 2005. 12th International Symposium on*, pages 65–73. IEEE, 2005.

[83] B. Yi and C. Faloutsos. Fast time sequence indexing for arbitrary lp norms. In *The VLDB Journal - The International Journal on Very Large Data Bases*. VLDB, 2000.

[84] H. Yue and S. Qin. Reconstruction-based fault identification using a combined index. *Industrial & engineering chemistry research*, 40(20) :4403–4414, 2001.

[85] Y. Zhang. *Improved methods in statistical and first principles modeling for batch process control and monitoring*. PhD thesis, University of Texas at Austin, 2008.