



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>



École doctorale IAEM Lorraine

Doctorat THÈSE

pour obtenir le grade de docteur délivré par

Université de Lorraine
Spécialité doctorale “Mathématiques”

présentée et soutenue publiquement par

Zhanhao Liu

le 22 Avril 2021

**Méthodes statistiques et variationnelles de modélisation
préalable au contrôle de procédés industriels**

Directeur de thèse : **Thomas Chambrion**

Co-directeur de thèse : **Radu S. Stoica**

Co-encadrante de thèse : **Marion Perrodin**

Jury

David Brie,

Saïd Moussaoui,

Samuel Vaiter,

Marianne Clausel,

Claire Boyer,

Vincent Duval,

Diane Bienaimé,

Professeur à Université de Lorraine

Professeur à Ecole Centrale de Nante

Chargé de recherche à CNRS

Professeure à Université de Lorraine

Maîtresse de conférence à Sorbonne Université

Chargé de recherche à INRIA

Ingénieure de recherche à Saint-Gobain Research Paris

Président

Rapporteur

Rapporteur

Examinatrice

Examinatrice

Examineur

Invitée

Résumé

Dans cette thèse, nous cherchons à proposer une méthodologie pour analyser des procédés industriels à partir des données procédé collectées à l'aide de méthodes statistiques et variationnelles. L'objectif est d'identifier les facteurs clés garantissant le bon fonctionnement des procédés industriels en vue de la proposition de lois de contrôle de ceux-ci.

La première partie présente les résultats d'analyses statistiques appliquées aux paramètres d'un procédé de Saint-Gobain. Nous avons d'abord analysé les données procédé à l'aide d'outils statistiques classiques (l'analyse en composantes principales, la classification non-supervisée, etc) et cherché à relier les informations ainsi obtenues au fonctionnement du procédé. Puis, nous avons analysé une mesure de qualité du produit final (appelée cible) en fonction des paramètres opératoires. La cible étant faiblement corrélée avec les paramètres enregistrés, l'hypothèse du modèle linéaire est rejetée. Un ensemble restreint des paramètres contribuant à l'explication de la cible a été identifié par les méthodes statistiques utilisées, ce qui a été d'ailleurs validé auprès de l'expert procédé. Ensuite, nous avons testé les modèles additifs généralisés (GAM) en introduisant la non-linéarité dans la modélisation, ce qui a amélioré la performance de nos modèles. Mais les modèles proposés restent insuffisants pour les futures applications. D'après l'intuition des opérateurs et de l'ingénieur process impliqués, un des signaux clés du fonctionnement du procédé était fortement bruité, et une des pistes développées pour améliorer la performance de nos modèles a été de reconstituer les informations manquantes de celui-ci.

Pour cela, dans la deuxième partie de la thèse, nous avons développé des méthodes (hors ligne et en ligne) de restauration par la régularisation de variation totale avec la détermination automatique de l'hyper-paramètre. Notre méthode d'estimation de l'hyper-paramètre a une performance similaire aux méthodes existantes dans les littératures scientifiques. Notre méthode pour estimer à la fois la restauration et l'hyper-paramètre est adaptée au traitement d'une grande quantité de données en temps réel. Des applications d'analyse de motifs et de détection de ruptures ont ensuite été développées pour plusieurs procédés industriels.

Abstract

In this thesis, we aim to propose a methodology for analyzing industrial processes based on a large quantity of process data with statistical and variational methods. The objective is to identify the key factors which ensure the good functioning of industrial processes. It is a preliminary step for the automatic generation of process control law from the collected data.

In the first part of this thesis, we presented the statistical analysis of a process of Saint-Gobain. At first, we applied some classical statistical tools (Principal component analysis, clustering and so on) to the process data, and linked the obtained information with the functioning of the process. Then, we analyzed a product quality measure (called target) with the collected process parameters. The target is weakly correlated with the parameters, so the hypothesis of linear model is rejected. A restricted list of parameters which contribute to the explanation of the target was identified by the statistical methods and validated by our industrial interlocutors. Afterwards, we tested a non-linear model method : the generalized additive model (GAM). The introduction of the non-linear terms improved the performance of our model, but it remained insufficient for the future applications. Following the intuition of the process engineers and operators, we focused on a noisy signal, tracked regularly in the plants, characterizing the good functioning of the process, and restoring the missing information of this signal may improve the model.

In the second part of this thesis, we developed a total variation restoration method with an automatic choice of hyper-parameter. Furthermore, our proposition of hyper-parameter has a similar performance as the existing methods, and our estimation method of both hyper-parameter and restoration is well fitted for the real time processing of a large quantity of data. Based on the proposed method, we have developed some applications of pattern restoration and discontinuities detection for several industrial processes.

Remerciements

Tout d’abord, je souhaite remercier chaleureusement Marion, Radu et Thomas, mes directeurs de thèse, grâce à qui j’ai énormément appris sur le plan technique comme sur le plan humain. Je leur suis également reconnaissant pour le temps conséquent qu’ils m’ont accordé, pour toutes leurs aides, leurs conseils et leur écoute qui ont permis la réussite de cette thèse.

Je tiens également à remercier Saïd Moussaoui et Samuel Vaiter, qui ont accepté de rapporter ma thèse, ainsi que tous les autres membres du jury Davie Brie, Marianne Clausel, Claire Boyer, Vincent Duval et Diane Bienaimé de m’avoir fait l’honneur d’examiner ce travail.

Mes remerciements vont aussi à mes collègues de Datalab de Saint-Gobain Recherche : Frédéric, Tristan, Antoine, Amaury, Christopher, Camille, Alessandro, Sami, Yuliya, Loïc, Matthieu, Simon, Paul, Mojdeh, Adélaïde, Yasmina, Julie, Lydia, Quentin, Vincent, Mathilde, Alexandra, Wendy, Jacques, Paul-henri. Mes remerciements s’adressent aussi à tous les collègues avec qui j’ai collaboré durant la thèse : notamment Jean-Sébastien David, Thierry Capla, Ludovic Hericher, Christian Fourel, Baptiste Menges, Diane Bienaimé, Yann Desailly et Laure Cossalter pour leurs aides à la compréhension des procédés industriels bien complexes.

Je tiens également à remercier le laboratoire IECL, et notamment l’équipe Pobra-stat, ainsi que les deux équipes INRIA dont j’ai pu faire partie (Sphinx et Pasta) pour l’accueil, les séminaires scientifiques et les bonnes conditions de travail.

Merci à tous mes amis que j’ai eu le plaisir de côtoyer durant ces dix années en France, à savoir Hugo, Aurélien, Martin, Bastien, Quentin, Nicolas, Paul, Damien, Xiao, Yutao, Kou, Nassim, Youri, Vincent, Rodolphe, PA, Yiming, Valentin, Philippe, Clémence, ...

Enfin, mes plus grands remerciements vont bien évidemment à mes parents, ma sœur et toute ma famille. Je ne pourrais pas conclure mes remerciements sans mentionner ma copine Siqui, pour tout son soutien illimité et sa confiance permanente, sans lesquels cette thèse n’aurait jamais vu le jour.

Table des matières

Table des matières	vii
Table des figures	ix
1 Introduction	1
I Modélisation statistique d'un procédé de centrifugation	3
2 Modélisation statistique d'un procédé de centrifugation : aperçu de la méthode suivie	5
2.1 Modélisation statistique d'un procédé de centrifugation	6
II Restauration par la régularisation de Variation Totale et ses applications	11
3 Méthode automatique de restauration d'un signal 1D par la régularisation de Variation Totale	13
3.1 Méthode de restauration par la régularisation de variation totale	14
3.2 Implémentation	30
3.3 Performance des méthodes proposées	32
3.4 Conclusion	39
4 Applications industrielles de la restauration automatique du signal	43
4.1 Applications pour un signal localement lisse	44
4.2 Applications autour d'une mesure de qualité en 2D	60
4.3 Applications autour d'une mesure de qualité en temps réel	65
4.4 Application : surveillance de la dérive de la variance du bruit	71
5 Conclusions générales et perspectives	77
5.1 Conclusions	77
5.2 Perspectives	78
Références	79

Table des figures

2.1	Projection des individus. (a) : chaque couleur correspond à un poste. (b) : chaque couleur correspond à une tranche de basket. Plus la couleur est claire, plus le numéro de tranche est grand.	8
3.1	Illustration des paliers et des jonctions. Le dernier point du dernier palier n'est pas une jonction de paliers.	17
3.2	Exemple simulé : comment choisir un bon λ à partir du nombre d'extremums locaux($g(\lambda)$).	23
3.3	Illustration de l'implémentation en ligne : les parties changées et inchangées de $\Lambda^{\circ, n+1}$ avec l'introduction d'un nouvel échantillon.	27
3.4	Illustration d'un arbre binaire de recherche. Un nœud est représenté par un cercle, et l'étiquette d'un nœud correspond au chiffre affiché dans le cercle.	30
3.5	Illustration des 2 types de signal périodique simulé (u_{net}). Exemples de $z = u_{net} + \epsilon$ avec $\epsilon \sim \mathcal{N}(0, 4)$ sont affichés en rouge.	34
3.6	Les caractéristiques statistiques de $d(\lambda, \lambda_{op})$ sur 500 simulations de deux types de signaux périodiques avec des bruits gaussiens de différentes variances. Nous avons testé notre approche avec $\log_{10}(q) \in \{0,5; 1\}$ et le choix automatique de q . Ligne continue : choix automatique de q ; ligne de points : $\log_{10}(q) = 0.5$; ligne pointillée : $\log_{10}(q) = 1$; ligne tiret-point : 10-fold cross-validation . Marqueur rond : 25% quantile ; Marqueur triangle : médiane ; Marqueur carré : 95% quantile.	35
3.7	Exemple d'un signal aperiodique simulé. $z = u_{net} + \epsilon$ avec $\epsilon \sim \mathcal{N}(0, 4)$. Couleurs : signal original u_{net} et échantillons simulés z	36
3.8	Temps d'exécution moyen des implémentations hors ligne et en ligne avec différents points de découpe $\hat{\lambda}$. Le temps d'exécution est moyenné sur 20 simulations.	38
4.1	Illustration des VT-restaurations avec les valeurs de λ estimées par différentes méthodes. Couleurs : signal brut t_c , SURE , AUT et notre méthode automatique (3.18) avec le choix automatique de q . Données : ensemble F1.	47
4.2	Illustration des VT-restaurations avec les valeurs de λ estimées par différentes méthodes. Couleurs : signal brut t_c , SURE , AUT et notre méthode semi-automatique avec $\log_{10}(q) = 1$ et $p = 0,03$. Données : ensemble F1.	48
4.3	Illustration des actions de maintenance détectées à partir d'un signal restauré. Les maintenances correspondent au dernier point des grandes descentes du signal restauré.	49
4.4	Illustration de maintenances détectées. Couleurs : maintenances détectées avec $\alpha_1 = 5/12$ et $\alpha_2 = \alpha_1/1,5$, signal brut t_c et restauration t_c proposée. Données : ensemble F1.	50
4.5	Pourcentages des produits anormaux en fonction de la durée d_g . La durée correspond au nombre de produits réalisés depuis la dernière maintenance pendant le poste, ou depuis le démarrage du poste au début du poste. Données : ensemble B, inclus dans F1.	53
4.6	Pourcentages des produits anormaux en fonction des résidus de restauration. Données : ensemble B, inclus dans F1.	53

4.7	Pourcentages des produits anormaux en fonction de la durée d_g . La durée correspond au nombre de produits réalisés depuis la dernière maintenance pendant le poste, ou depuis le démarrage du poste au début du poste. Données : ensemble B, inclus dans P1.	55
4.8	Pourcentages des produits anormaux en fonction des résidus de restauration. Données : ensemble B, inclus dans P1.	55
4.9	Histogramme de l'écart-type de y dans un poste de production. Données : ensemble B, inclus dans F1.	56
4.10	Classification des résidus de restauration : ce dendrogramme utilise la distance Kolmogorov-Smirnov et le saut moyen. Les groupes identifiés sont représentés par les branches ayant une même couleur dans le dendrogramme. Chaque feuille de l'arbre binaire affiché correspond à un poste de production. Données : ensemble B, inclus dans F1.	57
4.11	Illustration des fonctions de répartition du résidu de restauration pour les différents postes de production considérés. Une courbe correspond à un poste de production. Chaque couleur représente un groupe identifié par la méthode CAH (c.f. Figure 4.10). Données : ensemble B, inclus dans F1.	57
4.12	Histogramme normalisé de l'écart-type de l'écart de qualité (y) d'un poste dans les groupes verts et rouges identifiés par la CAH (c.f. Figure 4.10). Données : ensemble B, inclus dans F1.	57
4.13	Histogramme normalisé de l'écart-type des résidus de restauration. Couleurs : bon postes et mauvais postes. Données : ensemble B, inclus dans F1.	58
4.14	Histogrammes normalisés de l'écart-type des résidus de restauration. Couleurs : bon postes et mauvais postes.	58
4.15	Mesure brute de qualité : exemple de 200 lignes de mesure. Plus la couleur est claire, plus la valeur de la mesure est grande. L'objectif serait d'homogénéiser tout le produit à la valeur de mesure de zéro, qui est la qualité visée.	60
4.16	Illustration des groupes de capteur. Le numéro de capteur est affiché dans la cellule de tableau. Chaque groupe contient 3 capteurs successifs.	61
4.17	Illustration de la découpe des parties dans une ligne de mesure. La ligne noire correspond au profil moyen d'une production d'une heure. Les valeurs moyennes dans chaque partie sont représentées par une ligne avec la couleur correspondante. Abscisse : le numéro de capteur, ordonnée : la mesure de qualité. Couleurs : partie gauche, partie centrale et partie droite.	62
4.18	Illustration des caractéristiques statistiques de 6000 lignes de mesures lors d'un poste de production. (a) : la médiane des lignes de mesure. (b) : la restauration des médianes proposée par notre méthode automatique et la moyenne mobile sur 600 points. Données : ensemble E2.	62
4.19	Illustration des restaurations des quantiles de mesure de qualité pour une longue production. Couleurs : quantile 87,5%, quantile 75%, médiane, quantile 25% et quantile 12,5%. Données : ensemble E2.	63
4.20	Analyse de Fourier des valeurs moyennes des lignes de mesure ($\text{moyenne}(z^j)$ pour $j = 1, \dots, n$) sur une longue production. Trois composantes périodiques se détachent. Données : ensemble E2.	63
4.21	Un exemple de signal collecté durant plusieurs jours de production. Données : ensemble E3.	65
4.22	Estimation de ψ fondée sur les fenêtres glissantes w_i^m avec différentes tailles de fenêtre (m). Abscisse : index i de la fenêtre w_i^m . Données : ensemble E3.	67
4.23	Histogramme de ψ avec les fenêtres glissantes de la taille $m = 200$. Le signal brut est affiché sur la Figure 4.21a. Données : ensemble E3.	68
4.24	Résultats de détection avec la méthode locale robuste. Le signal brut est affiché sur la Figure 4.21a. Couleurs : VT-restauration globale avec notre proposition de λ , moyenne mobile sur une fenêtre de 120 points et ruptures détectées par la méthode locale robuste (4.4) avec $m = 200$, $\alpha = 40$ et $p = 20$. Données : ensemble E3.	69

4.25	Comparaison entre les détections par la méthode locale et celles par la méthode globale. Les quatre fenêtres détectant les changements brusques sont affichées. L'ensemble de détections par la méthode locale est affiché sur la Figure 4.24. Couleurs : ruptures détectées par la méthode locale (4.4), ruptures détectées par la méthode globale, VT-restauration sur l'ensemble d'échantillons avec notre proposition de λ et restauration par la moyenne mobile de 120 points (points noirs). Données : ensemble E3.	69
4.26	Un exemple de l'estimation de la variance du bruit avec une fenêtre de 400 points. . .	73
4.27	Résultats de 100 simulations sous les différents types de bruit. (1) : $\mathcal{N}(0, \sigma_a^2(t))$, (2) : $\mathcal{N}(0, \sigma_b^2(t))$, (3) : $\mathcal{U}(-d(t), d(t))$ et (4) : $\mathcal{N}(0, \sigma_a^2(t)) + \mathcal{U}(-1, 1)$	74
4.28	Score RVE de 100 simulations de notre méthode et MAD avec $\epsilon \sim \mathcal{N}(0, \sigma_c^2(t))$ et $\sigma_c(t) = 2 + 0.0005t$	74
4.29	Score RVE de 100 simulations de notre méthode avec les différentes tailles de fenêtre (m). $\epsilon \sim \mathcal{N}(0, \sigma_a^2(t))$	74

Chapitre 1

Introduction

Aujourd'hui, les industries se modernisent et se digitalisent, et cette évolution est connue sous le nom d'*Industrie 4.0*. Un des objectifs est d'améliorer la performance de procédés à l'aide de données collectées issues des capteurs installés sur ceux-ci. Les questions clés sont les suivantes :

- Comment capturer les facteurs clés du bon fonctionnement du procédé à partir des données collectées ?
- Comment mieux réguler le procédé à partir des facteurs clés identifiés ?

Il n'existe cependant pas une recette unique pour répondre à ces questions en raison de la multitude des procédés existants, ce qui nécessite l'accompagnement, étape par étape, de la transformation numérique du procédé considéré avec la collaboration des experts métier, et l'application d'outils de statistiques et d'analyse de contrôle. Pour répondre à la première question, plusieurs étapes préliminaires sont nécessaires :

- La collection des données : il s'agit d'enregistrer les paramètres procédé de la manière adéquate (fréquence, précision, etc) pour capturer les phénomènes importants et de bien structurer les données issues de différentes étapes de fabrication.
- L'analyse statistique classique : les données sont usuellement d'abord analysées à l'aide de méthodes statistiques classiques (comme par exemple l'analyse en composantes principales, la classification non-supervisée, etc). Les objectifs de cette étape sont de comprendre la structure des données collectées et d'identifier si possible les paramètres procédé contribuant à l'explication de la cible définie par les experts métier.

Dans cette thèse, nous avons développé une telle approche d'accompagnement de transformation numérique pour un procédé particulier de Saint-Gobain. Des données procédé issues de différentes usines ont d'abord été analysées par des outils statistiques classiques, puis nous avons cherché à modéliser une mesure de qualité en fonction des paramètres procédé accessibles avec des méthodes linéaires. L'hypothèse linéaire entre la mesure de qualité et ces paramètres procédé a été rejetée, mais cette analyse nous a permis d'identifier une liste restreinte de paramètres contribuant à l'explication de la cible. En se basant sur les paramètres sélectionnés, des analyses plus fines ont ensuite été effectuées : les modèles additifs généralisés ont amélioré la performance de modèle prédictif. Pour aller plus loin, nous avons envisagé des méthodes plus complexes, ayant le plus souvent plusieurs termes impliqués, comme par exemple un terme d'attachement aux données et plusieurs termes de régularisation concernant les informations a priori imposées aux modèles construits. Les différents termes sont pondérés par des hyper-paramètres dont le choix joue un rôle important sur la performance de la méthode.

Pour cela, nous avons travaillé sur le problème d'estimation d'hyper-paramètre dans la deuxième partie de la thèse, et nous nous sommes concentrés sur une méthode de restauration du signal 1D par la régularisation de variation totale. Des algorithmes en temps réel et en ligne pour estimer la restauration avec un choix automatique d'hyper-paramètre ont été proposés. Des applications d'analyse de motifs et de détection de ruptures ont ensuite été développées pour plusieurs procédés industriels.

Une méthodologie générale pour la modélisation de procédés industriels est encore en cours de développement. À ce stade, le travail concernant l'analyse statistique et la restauration montre déjà

son applicabilité pour plusieurs procédés de Saint-Gobain.

Le manuscrit est construit de la manière suivante :

- La partie [I](#) décrit les travaux de modélisation statistique d'un procédé de Saint-Gobain. Cette partie est soumise à l'obligation de confidentialité. Un résumé public de ces travaux est toutefois présenté dans ce document.
- La partie [II](#), contenant les chapitres [3](#) et [4](#), présente la méthode de restauration en temps réel par variation totale avec la détermination automatique de l'hyper-paramètre développée au cours de cette thèse.
 - Dans le Chapitre [3](#), nous proposerons d'abord des méthodes (hors ligne et en ligne) d'estimation de la restauration et de l'hyper-paramètre, puis les comparerons avec les méthodes existantes dans la littérature scientifique.
 - Dans le Chapitre [4](#), plusieurs applications de cette méthode de restauration à des procédés industriels sont montrées et analysées.
- Le Chapitre [5](#) fait le point sur l'état des travaux, tire les conclusions et ouvre sur les perspectives de recherche et d'application de ces travaux.

Première partie

Modélisation statistique d'un procédé de centrifugation

Chapitre 2

Modélisation statistique d'un procédé de centrifugation : aperçu de la méthode suivie

Nous avons dans un premier temps cherché à analyser un procédé industriel de Saint-Gobain (que nous appellerons *Process 1*) avec des méthodes statistiques. Les objectifs étaient de comprendre le fonctionnement du procédé à partir des données collectées et de modéliser une mesure de qualité en utilisant les paramètres procédé enregistrés. L'intégralité de ce travail a donné lieu à un rapport interne confidentiel à Saint-Gobain Recherche [25].

Le *Process 1* est contrôlé par plusieurs opérateurs à l'aide d'algorithmes de régulation sur la qualité des produits finaux. Les paramètres opératoires sont enregistrés durant la production : pour chaque produit réalisé (appelé *individu*), plusieurs paramètres procédé (appelés *variables*) sont enregistrés, et une mesure de qualité (appelée *cible*) est réalisée à la fin de la fabrication. Le versement est une étape cruciale du *Process 1* et consiste à verser des matières premières dans la machine de formage du produit final. Chaque produit donne lieu à un versement. On regroupe l'ensemble des produits fabriqués par une équipe d'opérateurs donnée, sur une plage temporelle donnée, sous l'appellation "poste".

Nous avons d'abord analysé des données issues de deux usines (appelées respectivement "P" et "F") en utilisant des outils statistiques, dont l'analyse en composantes principales (ACP) [21]. Plusieurs informations ont été remontées par les analyses :

- Un paramètre opératoire a_1 concernant le versement des matières premières avait une distribution multimodale dans les données de l'usine P, ce qui indiquait l'existence de plusieurs points de fonctionnement dans l'étape de versement et semblait problématique pour la stabilité de la qualité finale du produit.
- La distribution de la cible n'était pas uniformément indépendante dans un poste, et nous avons constaté une dépendance entre les produits voisins temporellement parlant.
- Les projections des individus proposées par l'ACP ont montré l'existence de clusters d'individus avec une forte proximité des individus d'un même poste. Les clusters pourraient représenter les points de fonctionnement de la machine en raison des différentes conditions de fabrication d'un poste à l'autre.
- Les projections des individus ont aussi mis en évidence les comportements spécifiques des premiers produits réalisés immédiatement après le remplissage des matières premières par rapport aux produits "normaux", ce qui correspond bien aux connaissances métier.
- Les cercles des corrélations proposés par l'ACP ont montré que certaines variables étaient fortement corrélées entre elles. Une étape de sélection de variables était nécessaire afin d'avoir un modèle statistique robuste pour l'explication de la cible.
- La cible était cependant faiblement corrélée avec les paramètres enregistrés, donc les modèles linéaires utilisant ces variables n'auront pas une bonne performance pour prédire la cible.

Ensuite, nous avons analysé les clusters d'individus et les groupes de variables corrélées avec des méthodes de classification non-supervisée. En analysant le paramètre a_1 dans les données de l'usine P, nous avons identifié un changement de méthode de mesure (changement de carte de conversion) non archivé dans la base de données et que deux types d'outils de versement étaient utilisés durant les productions, ce qui a expliqué les points de fonctionnement identifiés dans l'ACP. Nous avons aussi appliqué la méthode de classification ascendante hiérarchique (CAH) pour identifier les groupes de variables suivant la distance de corrélation [12].

Enfin, nous avons testé des méthodes de modélisation classiques, en partant du plus simple et en complexifiant petit à petit [22] [19] [20] [2], pour modéliser la cible en fonction des paramètres opératoires, tout en tenant compte des connaissances sur le fonctionnement du procédé et des informations issues des analyses précédentes. La régression linéaire des produits "normaux" (l'ensemble des produits à l'exception de ceux réalisés immédiatement après le remplissage des matières premières) que nous avons réalisé a obtenu respectivement un score R^2 de 0,127 pour les données de l'usine P et 0,232 pour les données de l'usine F. Ces faibles scores étaient attendus du fait de la faible corrélation entre la cible et les variables constatée dans les analyses précédentes, ce qui nous a permis de rejeter l'hypothèse linéaire. Nous avons ensuite introduit des dépendances entre les produits voisins dans le modèle. Notre meilleur modèle ainsi obtenu avait respectivement un R^2 score de 0,242 dans les données de l'usine P et 0,411 dans les données de l'usine F. Ces modèles sont linéaires avec les variables (simples et quadratiques) ayant permis de produire l'individu concerné, mais également celles des individus voisins.

Afin de faciliter l'interprétation des coefficients estimés, nous avons sélectionné une liste de n_v variables pertinentes avec des méthodes statistiques, et cela a été validé auprès d'expert procédé impliqué. Nous avons testé les méthodes non-linéaires en utilisant les splines sur les variables sélectionnées (sans l'introduction des variables des individus voisins), mais la performance des modèles reste alors faible avec un score R^2 de 0,290 et 44 degrés de libertés pour les données de l'usine F.

L'introduction de la non-linéarité a amélioré le modèle prédictif, ce qui montre une relation non-linéaire entre la cible et les paramètres procédé. Cela indique qu'il est nécessaire d'analyser finement les variables sélectionnées. D'après l'intuition des opérateurs et de l'ingénieur process impliqués, le signal bruité t_c , suivi couramment à l'usine, est l'un des facteurs clés du bon fonctionnement du procédé, et les maintenances réalisées par les opérateurs (souvent non enregistrées) introduisent des changements brusques de la valeur de t_c . Une des pistes développées ici a été d'analyser finement le signal de t_c en reconstituant la variation intrinsèque du signal et les informations concernant les actions de maintenance par une méthode de restauration.

D'ailleurs, la valeur d'auto-corrélation des résidus de modélisation reste significative entre les produits voisins, et les méthodes de type auto-régressif pourraient améliorer la performance du modèle prédictif.

Une des difficultés que nous avons rencontrées lors de la modélisation a été le choix des hyper-paramètres pondérant le termes d'attachement aux données et la régularisation du modèle. La performance des modèles statistiques (par exemple : régression Lasso [39], SVM [10], random forest [6]) dépend en effet sensiblement du choix des hyper-paramètres. De ce fait, nous nous sommes par la suite attelés à l'estimation de ces hyper-paramètres.

2.1 Modélisation statistique d'un procédé de centrifugation

Dans cette section, nous allons présenter les résultats des analyses statistiques effectuées sur le procédé de centrifugation utilisé par Saint-Gobain Pont-à-Mousson. Cela a fait l'objet d'une communication dans un congrès national [26].

Le procédé analysé fabrique des tuyaux en fonte, et les principales étapes sont :

- le *basculement* : de la fonte en fusion est versée dans un récipient, que nous appelons un "basket", contenant l'équivalent en fonte de plusieurs tuyaux. Le moule est ensuite alimenté en fonte par des inclinaisons successives du basket.

- la *translation* : un chariot porte le moule rempli de fonte, en faisant des aller-retour sur une rampe entre un point haut et un point bas. Au point bas, qui est aussi le point de repos, la fonte est déjà transformée en tuyau, et le tuyau est livré. Le chariot remonte à vide pour être rempli à nouveau.
- la *rotation* : la rotation du moule permet la formation des tuyaux par force centrifuge. La rotation démarre en même temps que le versement de la fonte, et elle continue pendant la translation.
- l'*extraction* : quand le chariot est au point bas, un bras mécanique fait sortir le tuyau du moule.

Une mesure d'efficacité de ce procédé de fabrication de tuyaux est la différence, notée y , entre la masse du tuyau réalisée et la masse demandée par le cahier des charges. L'objectif de cette étude est de comprendre le rôle de chaque étape dans l'obtention d'une production de qualité, soit minimiser y sous la contrainte $y \geq 0$. Nous allons étudier les différentes corrélations existantes entre les données enregistrées lors du procédé de fabrication, puis nous proposerons des modèles statistiques pour l'estimation de l'écart de masse y . Dans la suite, nous allons tout d'abord présenter les données enregistrées lors du procédé de fabrication, puis les résultats d'une analyse en composantes principales. Ensuite, plusieurs modèles seront proposés et étudiés, parmi lesquels des modèles linéaires et des modèles de convolution avec des dépendances "spatio-temporelles". Enfin, nous présenterons les conclusions tirées de ces premiers résultats et les perspectives qui en découlent.

2.1.1 Données

Les données sont réparties dans plusieurs fichiers, chaque fichier contenant les informations des tuyaux fabriqués durant un poste de 8 heures.

Variables

Différentes variables sont mesurées pendant les étapes de fabrication. Notamment, sont mesurés durant le basculement, les mouvements angulaires du basket, les durées d'action et les vitesses de basculement ; durant la translation, les distances parcourues par le chariot et les durées ; durant la rotation, les vitesses de rotation du moule et les durées des "régimes" ; et enfin durant l'extraction, les durées de l'action.

Des variables quadratiques sont créées directement dans la base pour caractériser la quantité de fonte versée : ce sont les vitesses de basculement multipliées par les durées appliquées. Quant aux variables qualitatives, il y a par exemple des indicateurs d'anomalies ou la numérotation des tuyaux. Au total, nous avons initialement 61 variables quantitatives et 28 variables qualitatives, qui ont été standardisées avant leur utilisation.

Individus

Nous désignerons désormais un tuyau fabriqué par le terme "individu". Le terme "basket" désigne à la fois l'outil de versement de fonte et les tuyaux réalisés avec la fonte contenue par celui-ci entre deux remplissages. Par exemple, ce que nous appellerons le 4^{ième} basket correspondra aux tuyaux réalisés entre le 4^{ième} et le 5^{ième} remplissage du basket proprement-dit au cours d'un même poste. Le numéro de tranche est quant à lui défini par rapport au niveau de fonte restante dans le basket lorsque la fabrication du tuyau concerné commence. Le nombre de tuyaux réalisés avec un basket n'est cependant pas constant car le remplissage de celui-ci peut fluctuer et certaines tranches peuvent ne pas être présentes pour certains baskets.

Les individus sont caractérisés par 3 facteurs : le poste, le basket et le numéro de tranche. On note le tuyau du i ^{ième} basket, de la j ^{ième} tranche et du k ^{ième} poste comme $T_{i,j,k}$. A chaque $T_{i,j,k}$ correspondent donc des mesures ou des paramètres $X_{i,j,k}$ et une cible $y_{i,j,k}$.

Nous avons initialement 50 225 individus dans notre base de données. Nous avons sélectionné aléatoirement 50% des postes pour le training set, 25% pour le cross-validation set, et 25% pour le test set.

Nous avons considéré les individus ayant une variable dont la valeur standardisée absolue était supérieure à 5 comme des outliers. Les outliers et les individus avec un indicateur d'anomalie actif ont été supprimés de nos différents data sets. Après filtrage, nous avons 21 149 individus dans le training set, 14 562 dans le cross-validation set et 9 611 dans le test set.

2.1.2 Analyse exploratoire

Les données sont traitées via une analyse en composantes principales, qui nous indique qu'en gardant 19 composantes, on préserve 90% de variation globale des données, tout en réduisant la dimension du problème.

Les cercles de corrélation nous montrent des corrélations significatives entre certaines variables caractérisant la même étape du processus. Cela indique que ces variables pourraient être regroupées pour la construction d'un modèle. Toutefois, le fait que la variation de la cible soit faiblement caractérisée par cette nouvelle représentation des données, nous donne de faibles espoirs pour qu'un modèle simple puisse expliquer nos données. Cette hypothèse sera d'ailleurs rejetée dans la section suivante.

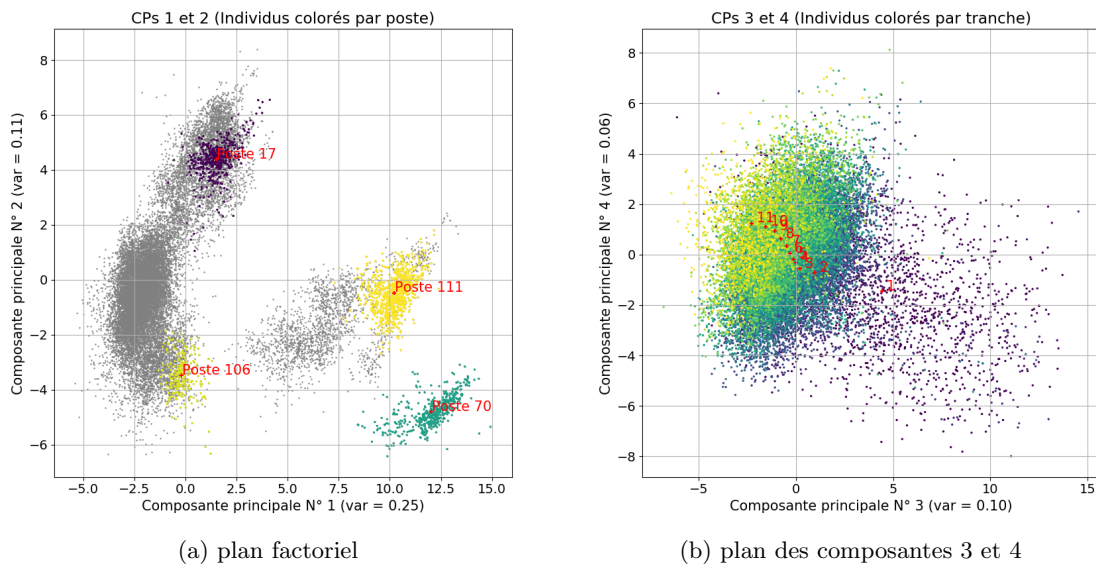


FIGURE 2.1 – Projection des individus. (a) : chaque couleur correspond à un poste. (b) : chaque couleur correspond à une tranche de basket. Plus la couleur est claire, plus le numéro de tranche est grand.

Les projections des individus dans les plans des quatre premières composantes sont présentées dans la Figure 2.1. Nous observons que les individus forment des agrégats en fonction de leur appartenance à une tranche et/ou à un poste. Cela suggère une dépendance temporelle (dans un poste) et spatiale (entre les mêmes tranches) entre les individus. La dépendance temporelle est due aux différentes conditions de travail : la température ambiante, la composition de la fonte, le type de basket utilisé, les habitudes des opérateurs ; tandis que la dépendance spatiale est liée à l'angle de versement du début cycle. Bien que l'information concernant le type de basket ne soit pas disponible sur notre base de données, nous avons pu mettre en évidence son importance à travers des méthodes de classification. Il nous apparaît important que notre modèle final puisse aussi expliquer la répartition en agrégats des individus.

2.1.3 Modélisation

Cette section présente plusieurs étapes, de la plus simple à la plus élaborée, pour la construction des modèles qui pourraient expliquer nos données. Ce travail est en cours.

Suite à nos observations précédentes, les premières tranches ont été supprimées des trois sets du fait de leur comportement spécifique, qui aurait pu interférer dans nos modélisations.

Régression simple

L'hypothèse la plus simple à vérifier concernant la variation de la cible est la suivante : peut-on l'expliquer par la combinaison linéaire des données observées lors de son processus de fabrication, du basculement jusqu'à l'extraction ? Avec les notations introduites, un tel modèle peut s'écrire comme :

$$\hat{y}_{i,j,k} = \theta_0 + \sum_{l=1}^p \theta_l x_{i,j,k,l} \quad (2.1)$$

avec le vecteur des paramètres $\Theta = (\theta_0, \dots, \theta_p)$, les variables explicatives $x_{i,j,k,l}$ avec $l = 1, \dots, p$ et $\hat{y}_{i,j,k}$ la cible du tuyau $T_{i,j,k}$.

Suite à l'ACP, un regroupement des variables est effectué. Ceci nous permet de réaliser sur ces données une régression linéaire avec $p = 76$ paramètres. Après plusieurs expériences, la méthode d'estimation Lasso nous indiquait 28 paramètres significativement différents de zéro, alors que le meilleur score R^2 obtenu était seulement de 0,1. Ce faible score était attendu suite à l'analyse ACP. Il nous a permis d'écarter l'hypothèse la plus simple. Dans ce contexte, nous avons décidé d'introduire plus de dépendances dans le modèle [19].

Introduction de dépendances supplémentaires

La significativité de certains coefficients dans le modèle de régression linéaire simple indique que la variation de y du tuyau $T_{i,j}$ n'est pas indépendante des étapes de fabrication du tuyau lui-même. Le modèle que l'on propose ici introduit, en plus, des dépendances relatives à la fabrication des autres tuyaux sur le même poste $T_{i,j-1}$, $T_{i,j-1}$.

Dans ce contexte nous proposons d'estimer la variation du y du tuyau $T_{i,j}$ par :

$$\hat{y}_{i,j} = \theta_0 + \sum_{l=1}^p (\theta_l x_{i,j,l} + \theta_{l+p} x_{i-1,j,l} + \theta_{l+2p} x_{i,j-1,l}) \quad (2.2)$$

avec $\Theta = (\theta_0, \dots, \theta_{3p})$ les paramètres du modèle et x les mesures associées pour chaque tuyau considéré. Le modèle estimé par Lasso a un meilleur score R^2 de 0,196 avec 78 variables actives. Cela indique un meilleur comportement de ce modèle par rapport au modèle linéaire simple, et valide l'idée d'étude des dépendances entre tuyaux.

Clairement, le modèle doit encore être amélioré. Nous envisageons des modèles "spatiaux" de la forme :

$$\hat{y}_{i,j} = \theta_0 + \theta_1 y_{i-1,j} + \theta_2 y_{i,j-1} + \sum_{l=1}^p (\theta_{l+2} x_{i,j,l} + \theta_{l+2+p} x_{i-1,j,l} + \theta_{l+2+2p} x_{i,j-1,l}) \quad (2.3)$$

comparables à ceux de [2], [16] et [11].

2.1.4 Conclusion

Nous avons effectué une analyse en composantes principales des données du procédé de centrifugation de Saint-Gobain. Les résultats de celle-ci montrent que l'écart entre la masse réalisée du tuyau et sa consigne est peu corrélé linéairement avec les variables enregistrées lors de chacune des étapes de fabrication. Le meilleur modèle linéaire obtenu a d'ailleurs un score R^2 de 0,196 avec 78 variables actives, dont certaines quadratiques.

Une des pistes pour améliorer la performance de modèle est d'analyser finement les variables ainsi sélectionnées, et notamment leurs signaux temporels. En effet, l'un des facteurs clé du bon fonctionnement du procédé, d'après l'intuition des opérateurs et de l'ingénieur process impliqués, possède un signal fortement bruité, ce qui complique la modélisation. Nous proposons donc de reconstituer la variation intrinsèque de ce signal par une méthode de restauration avec régularisation de variation totale, dont la performance dépend du choix d'un hyper-paramètre λ . De ce fait, nous proposerons également une méthode de détermination automatique d'un bon hyper-paramètre λ en temps réel.

Deuxième partie

Restauration par la régularisation de Variation Totale et ses applications

Chapitre 3

Méthode automatique de restauration d'un signal 1D par la régularisation de Variation Totale

Sommaire

3.1	Méthode de restauration par la régularisation de variation totale	14
3.1.1	Modèle des échantillons	14
3.1.2	États de l'art	14
3.1.3	Analyses théoriques de la VT-restauration	15
3.1.4	Proposition des Algorithmes	21
3.2	Implémentation	30
3.2.1	Arbre binaire de recherche	30
3.2.2	Détails de l'implémentation de PD-VT-2	32
3.3	Performance des méthodes proposées	32
3.3.1	Métriques	32
3.3.2	Méthodes existantes	33
3.3.3	Résultats	34
3.4	Conclusion	39

La restauration à partir des observations bruitées, appelée aussi “problème inverse”, est un domaine de recherche actif. L'idée est de restaurer la variation intrinsèque du signal en éliminant les influences des bruits (comme par exemple le bruit de mesure).

Le bon fonctionnement des lignes de production est surveillé par les capteurs installés dans la ligne. Cependant, les bruits de mesure sont parfois non-négligeables et non-évitables à cause des conditions de travail extrêmes des capteurs ou des techniques de mesure déployées. Les bruits de mesures sont souvent problématiques car ils introduisent des imprécisions sur la surveillance du procédé et la prise de décision : les informations potentiellement intéressantes du signal sont noyées dans des échantillons bruités. Une bonne restauration des signaux mesurés nous permet de visualiser les variations intrinsèques des signaux qui représentent l'état de fonctionnement des lignes. C'est pourquoi une étape de restauration des signaux collectés est indispensable pour garantir l'efficacité des lignes de production. D'ailleurs, une grande quantité de données étant collectée en temps réel sur la ligne de production, une méthode adaptative pour les différentes situations sans intervention humaine faciliterait le déploiement et l'implémentation des méthodes de restauration dans les différents procédés.

Nous avons développé une méthode de restauration par la régularisation de la variation totale (VT) avec la détermination automatique du paramètre de pondération. Cette méthode automatique propose une bonne restauration en temps réel avec une faible charge computationnelle. La rapidité et la bonne performance de cette méthode nous donnent un large champ d'applications sur les procédés de Saint-Gobain.

Le plan de ce chapitre est le suivant : nous présenterons d'abord les analyses théoriques de la restauration par la variation totale et proposerons des algorithmes d'estimation dans la Section 3.1. Ensuite les détails d'implémentation sont présentés dans la Section 3.2. Enfin, nous analysons la performance des méthodes proposées et les comparons avec les méthodes existantes dans la Section 3.3.

3.1 Méthode de restauration par la régularisation de variation totale

3.1.1 Modèle des échantillons

Le signal u est une fonction réelle définie sur un sous-ensemble borné Ω de $\mathbb{R}$: $u : \Omega \rightarrow \mathbb{R}$. Un nombre fini des échantillons bruités de u est disponible. Le vecteur des échantillons bruités est noté $\mathbf{z} = (z_1, \dots, z_n)$ avec z_i l'échantillon à l'instant t_i , et le vecteur des temps d'échantillonnage est noté $\mathbf{t} = (t_1, \dots, t_n)$ où $t_1 < \dots < t_n$. On ne suppose pas que la période d'échantillonnage $\tau_i = t_i - t_{i-1}$ est constante. Le modèle mathématique de z_i est donné par :

$$z_i = u(t_i) + \epsilon$$

où ϵ est un terme de bruit avec $\mathbb{E}(\epsilon) = 0$ et $\mathbb{V}(\epsilon) = \sigma^2$.

Notre objectif est de restaurer le signal original $u(t)$ à partir des échantillons $\mathbf{z}$ en temps réel avec des ressources de calcul limitées.

3.1.2 États de l'art

Il existe plusieurs familles de méthode pour la restauration des signaux bruités, et nous allons présenter trois familles parmi les plus utilisées :

- Filtres numériques : pour un signal mesuré avec une période d'échantillonnage constante, les filtres numériques sont largement appliqués. Par exemple, le filtre Savitzky-Golay [36] est l'une des méthodes les plus populaires. Il s'agit de faire une régression polynomiale locale dans une fenêtre autour de chaque point (z_i, t_i) . Nous devons choisir en amont deux paramètres : la largeur de fenêtre et le degré maximal de la régression d . Dans le cas $d = 0$, la méthode est en fait la moyenne mobile. Cette méthode est particulièrement rapide si la période d'échantillonnage est constante, car la régression peut être formulée par une convolution avec un noyau pré-calculable. Dans le cas où la période n'est pas constante, le noyau de convolution doit être recalculé pour chaque point, ce qui rend la méthode inefficace.
- Méthodes probabilistes : en introduisant les distributions a priori sur les propriétés du signal et les paramètres du modèle, la restauration est obtenue par la maximisation de la distribution a posteriori. Les lecteurs pourraient retrouver plus de détails dans [42].
- Méthodes variationnelles : cette méthode consiste à définir une fonction d'énergie $F(\mathbf{z}, \mathbf{u}) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ et estimer le signal restauré par :

$$\mathbf{u}^* = \arg \min_{\mathbf{u}} F(\mathbf{z}, \mathbf{u})$$

En général, la fonction d'énergie est une somme pondérée de deux termes :

$$F(\mathbf{z}, \mathbf{u}) = L(\mathbf{z}, \mathbf{u}) + \lambda \Psi$$

avec le terme de *fidélité* $L(\mathbf{z}, \mathbf{u})$ et le *régularisateur* Ψ pondéré par le paramètre de poids λ . Habituellement, nous choisissons la norme L^2 de $(\mathbf{u} - \mathbf{z})$ comme terme de fidélité car il est convexe et dérivable.

La forme du régularisateur doit prendre en compte les connaissances *a priori* du signal. Bien évidemment, les différents régularisateurs donnent des résultats de reconstruction distincts. Nous avons travaillé sur la régularisation par la *variation totale* (VT), définie par :

$$\Psi = D(u) = \int_{\Omega} |\nabla u| \quad (3.1)$$

avec ∇u le gradient de u .

La méthode de restauration par la régularisation de la variation totale (*VT-restoration*) est d'abord introduite dans [33]. Les auteurs de [8] ont montré une équivalence à un problème d'optimisation sans contrainte :

$$u_{VT}^* = \arg \min_u \int_{\Omega} \lambda |\nabla u| + |u - z|^2 \quad (3.2)$$

et aussi une correspondance bijective entre la variance du bruit σ^2 et l'hyper-paramètre λ permettant la meilleure reconstruction du signal. Les méthodes VT-restoration proposent une restauration localement lisse en préservant les contours (i.e. les changements brusques). Elles sont appliquées aux signaux ([17]), aux images ([33], [8]) et plus généralement aux structures de graphe ([31], [24]).

Pour le problème en 1D, la solution exacte de (3.2) à λ fixé peut être estimée efficacement. Nous présentons ici deux familles de méthode :

- Dynamique de λ [40], [32] : ces algorithmes estiment les solutions en suivant la dynamique de u_{VT}^* en fonction de λ . Les auteurs de [40] ont proposé un algorithme *path* pour estimer Λ , la liste de λ qui évoque un changement de la topologie de u_{VT}^* , en faisant varier λ de $+\infty$ à 0. Ce type de méthode est efficace pour estimer les solutions avec plusieurs candidats de λ .
- Dynamique de “nouvel” échantillon : pour une séquence de n échantillons, l'auteur de [9] propose d'ajouter les échantillons un par un dans la considération et de mettre à jour séquentiellement la solution basée sur le comportement local de VT-restoration [29]. Ce type de méthode est particulièrement adapté au traitement en ligne d'un flux de données.

La performance de la restauration dépend du choix de λ . Le choix de λ est l'un des obstacles pour les applications réelles. Les approches traditionnelles consistent à évaluer une liste de candidats par une métrique choisie en amont, et puis sélectionner le meilleur paramètre suivant cette métrique. Ces approches (comme par exemple la validation croisée) sélectionnent aléatoirement une partie de données pour tester les modèles construits sans cette partie. La répétition de la résolution d'un même problème sur les différentes données rend ces méthodes non-adaptées au contexte de temps réel. Les autres approches estiment λ grâce aux connaissances a priori introduites : par exemple, en connaissant le plus grand nombre possible de morceaux constants du signal restauré, un choix de λ pourrait être proposé rapidement [9], [40]. Certaines méthodes efficaces sont basées sur les connaissances a priori du bruit ϵ : les auteurs de [7] et [41] proposent de sélectionner le paramètre sous le principe de contradiction de Morozov [30] : e.g. la norme de L^2 des résidus de restauration ($z - u_{VT}^*$) est égale à la variance du bruit, et les auteurs de [18] proposent de sélectionner le paramètre basé sur les caractéristiques statistiques des résidus de restauration. Si nécessaire, la variance du bruit σ^2 peut être estimée par [14] et [5].

En outre, des méthodes heuristiques sont disponible : les auteurs de [32] proposent d'estimer λ en suivant la quantité $|u_{VT}^*(\lambda_l) - u_{VT}^*(\lambda_{l-1})|^2$ avec λ_l le plus petit λ donnant une restauration avec $l - 1$ morceaux constants.

3.1.3 Analyses théoriques de la VT-restoration

Dans cette section, nous analysons d'abord la dynamique du signal restauré en fonction du paramètre λ . Ensuite, nous nous intéressons aux influences de l'introduction d'un nouvel échantillon et du mouvement des fenêtres glissantes aux signaux restaurés.

Présentation du modèle

Notre objectif est de restaurer le vecteur inconnu $\mathbf{u} = (u_1, \dots, u_n)$ à partir des observations bruitées $\mathbf{z} = (z_1, \dots, z_n)$ avec z_i l'échantillon à l'instant t_i . On introduit le vecteur des “périodes” d'échantillonnage $\tau = (\tau_1, \dots, \tau_n)$ avec :

$$\tau_i = \begin{cases} t_i - t_{i-1}, & i = 2, \dots, n. \\ t_2 - t_1, & i = 1. \end{cases}$$

On considère l'approximation numérique de la fonctionnelle (3.2) :

$$F(\cdot, \mathbf{z}, \tau, \lambda) : \mathbb{R}^n \rightarrow \mathbb{R}$$

$$u \mapsto \sum_{i=1}^n \tau_i (z_i - u_i)^2 + \lambda \sum_{i=1}^{n-1} |u_i - u_{i+1}| \quad (3.3)$$

La fonctionnelle $F(\mathbf{u}, \mathbf{z}, \tau, \lambda)$ est notée $F_n(\lambda)$ pour le cas de n échantillons. Le signal restauré est estimé par :

$$\mathbf{u}^*(\lambda) = (u_1^*(\lambda), \dots, u_n^*(\lambda)) = \arg \min F_n(\lambda) \quad (3.4)$$

La fonctionnelle considérée (3.3) est convexe mais non-dérivable. Comme nous travaillons sur un ensemble fini d'échantillons de signal, la solution $\mathbf{u}^*(\lambda)$ pourrait être considérée comme constante par morceaux. Nous utiliserons, par la suite, une notation similaire à [32] pour présenter les morceaux constants du signal : un ensemble d'index $\{j, j+1, \dots, k\}$ consécutifs dont les valeurs restaurées sont égales $u_j^*(\lambda) = \dots = u_k^*(\lambda)$ est appelé *palier* si cela ne peut pas être élargi. Autrement dit, $u_{j-1}^*(\lambda) \neq u_j^*(\lambda)$ (ou $j = 1$) et $u_k^*(\lambda) \neq u_{k+1}^*(\lambda)$ (ou $k = n$). Le nombre de paliers de $\mathbf{u}^*(\lambda)$ est noté $K(\lambda)$. Les notations suivantes sont introduites pour le j^e palier de $\mathbf{u}^*(\lambda)$:

- *Ensemble des index* : $\mathcal{N}_j(\lambda) = \{i_1^j, \dots, i_{n_j}^j\}$ avec $n_j(\lambda) = \text{Card}(\mathcal{N}_j(\lambda))$ contenant les index des points dans le j^e palier.
- *Niveau de palier* : $v_j^*(\lambda) = u_i^*(\lambda), \forall i \in \mathcal{N}_j(\lambda)$

Une représentation équivalente de $\mathbf{u}^*(\lambda)$ est fournie par $\{\mathbf{v}^*(\lambda), \mathcal{N}(\lambda)\}$ avec l'*ensemble des niveaux de palier* $\mathbf{v}^*(\lambda) = (v_1^*(\lambda), \dots, v_K^*(\lambda))$ où $v_1^*(\lambda) \neq v_2^*(\lambda)$, $v_2^*(\lambda) \neq v_3^*(\lambda)$, $\dots$, $v_{K-1}^*(\lambda) \neq v_K^*(\lambda)$ et le *découpage* de signal $\mathcal{N}(\lambda) = \{\mathcal{N}_1(\lambda), \dots, \mathcal{N}_K(\lambda)\}$. La variation totale du signal restauré sous la nouvelle présentation, donnée par (3.5), est dérivable.

$$VT(\mathbf{v}(\lambda)) = \sum_{j=1}^{K-1} |v_{j+1}(\lambda) - v_j(\lambda)| \quad (3.5)$$

Si le découpage $\mathcal{N}(\lambda)$ est connu, nous pouvons estimer le signal restauré $\mathbf{v}^*(\lambda)$ en minimisant la fonctionnelle suivante :

$$\mathbf{v}^*(\lambda) = (v_1^*(\lambda), \dots, v_K^*(\lambda))$$

$$= \arg \min \left\{ \sum_{j=1}^K \sum_{i \in \mathcal{N}_j} \tau_i (z_i - v_j)^2 + \lambda \sum_{j=1}^{K-1} |v_{j+1} - v_j| \right\} \quad (3.6)$$

Soient $s(\mathbf{v}) = \{s_0(\mathbf{v}), s_1(\mathbf{v}), \dots, s_K(\mathbf{v})\}$ avec :

$$s_i(\mathbf{v}) = \begin{cases} 0 & i = 0 \text{ ou } K \\ \text{signe}(v_{i+1} - v_i) & i = 1, \dots, K-1 \end{cases} \quad (3.7)$$

et $\mathbf{s}^* = s(\mathbf{v}^*(\lambda))$, les conditions d'optimalité du premier ordre de (3.6) impliquent que :

$$v_j^*(\lambda) = \bar{z}_j^* + \frac{\lambda}{2\mathcal{T}_j} (s_j^* - s_{j-1}^*) \quad (3.8)$$

avec $j = 1, \dots, K(\lambda)$, la longueur de palier $\mathcal{T}_j = \sum_{i \in \mathcal{N}_j(\lambda)} \tau_i$ et la valeur moyenne de j^e palier $\bar{z}_j^* = \frac{\sum_{i \in \mathcal{N}_j(\lambda)} \tau_i z_i}{\mathcal{T}_j}$.

Pour faciliter la présentation, nous introduisons les termes suivants, illustrés sur la Figure 3.1 :

- On va appeler un palier maximal local *palier max*, un palier minimal local *palier min* et le reste *palier neutre*.
- On va appeler le dernier point d'un palier (à part le dernier palier) *jonction* de paliers.

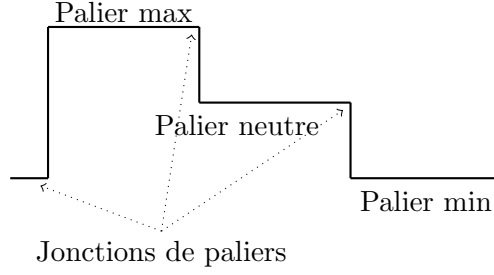


FIGURE 3.1 – Illustration des paliers et des jonctions. Le dernier point du dernier palier n'est pas une jonction de paliers.

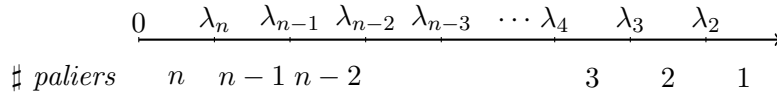
Influence de l'hyper-paramètre λ sur le découpage $\mathcal{N}(\lambda)$

Si on connaît le découpage $\mathcal{N}(\lambda)$, le calcul de $\mathbf{v}^*(\lambda)$ est direct via (3.8). Dans cette section, nous présentons comment le découpage $\mathcal{N}(\lambda)$ varie en fonction de la valeur de λ .

Les auteurs de [40] et [32] décrivent le comportement de la solution $\mathbf{u}^*(\lambda)$ lorsque λ diminue. Inspiré d'un théorème élégant dans [23] et [42] (Proposition 2.5.1, p52) concernant la relation par morceaux entre la restauration par les modèles Potts et le paramètre λ , nous reformulons les résultats de [40] et [32] par le théorème suivant :

Théorème 3.1. *Il existe une séquence $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_n, \lambda_{n+1})$ avec $0 = \lambda_{n+1} \leq \lambda_n \leq \lambda_{n-1} \leq \dots \leq \lambda_2 < \lambda_1 = \infty$ telle que $\forall \lambda_a, \lambda_b \in [\lambda_{l+1}, \lambda_l]$ avec $l = 1, \dots, n$, nous avons $\mathcal{N}^*(\lambda_a) = \mathcal{N}^*(\lambda_b)$. $\square$*

Remarque 1. *Presque sûrement sous hypothèse de bruit continu, seuls deux paliers sont fusionnés pour $\lambda \in \Lambda$. Nous avons donc $0 = \lambda_{n+1} < \lambda_n < \lambda_{n-1} < \dots < \lambda_2 < \lambda_1 = \infty$ tels que $\forall \lambda \in [\lambda_{l+1}, \lambda_l]$ avec $l = 1, \dots, n$, le signal restauré $\mathbf{u}^*(\lambda)$ a l paliers : $\text{card}(\mathcal{N}^*(\lambda)) = l$.*



Dynamique des niveaux de palier en fonction de λ

Le Théorème 3.1 nous permet de découper l'estimation de Λ dans des sous-problèmes plus simples : λ_l peut être estimé à partir de λ_{l+1} pour $1 < l \leq n$. Dans cette section, nous allons décrire la dynamique des signaux restaurés $u^*(\lambda)$ avec l'augmentation λ .

Pour $\lambda = 0$, (3.4) est une estimation des moindres carrés, donnant la solution $u^*(0) = z$. Si $z_i \neq z_{i+1}$ pour tout i , $u^*(\lambda)$ a n paliers pour $0 \leq \lambda < \lambda_n$, chaque palier ayant un point.

Pour simplifier la présentation, nous supposons que seuls deux paliers fusionnent pour $\lambda \in \Lambda$. Suivant la Remarque 1, Λ a $n + 1$ éléments distincts avec $\lambda_{n+1} = 0$ et $\lambda_1 = +\infty$.

Comme λ augmente, (3.8) implique que les paliers min et max s'approchent de leurs voisins, mais que les paliers neutres restent identique. Pour $\lambda \in \Lambda$, deux paliers voisins ont le même niveau et forment ensemble un nouveau palier. Prenons un exemple : supposons que λ_{l+1} est connu, ce paramètre nous donne une solution avec l paliers. Soient la solution $v^l(\lambda) = \{v_1^l(\lambda), \dots, v_l^l(\lambda)\}$ et $\mathcal{N}^l(\lambda) = \{\mathcal{N}_1^l(\lambda), \dots, \mathcal{N}_l^l(\lambda)\}$, nous introduisons les notations suivantes :

$$\beta_j^l = \frac{1}{2\mathcal{T}_j^l}(s_j^l - s_{j-1}^l) \quad (3.9)$$

$$\Gamma^l = (\gamma_1^l, \dots, \gamma_{k-1}^l) = (\beta_1^l - \beta_2^l, \dots, \beta_{l-1}^l - \beta_l^l) \quad (3.10)$$

$$\Delta v^l(\lambda) = (v_1^l(\lambda) - v_2^l(\lambda), \dots, v_{l-1}^l(\lambda) - v_l^l(\lambda)) \quad (3.11)$$

avec $s^l = s(v^l)$ et $\mathcal{T}_j^l = \sum_{i \in \mathcal{N}_j^l} \tau_i$. Soient $\lambda_{l+1} \leq \lambda_a < \lambda_b < \lambda_l$, comme le découpage de signal $\mathcal{N}^l(\lambda)$ reste identique, nous avons :

$$\Delta v_i^l(\lambda_a) = \Delta v_i^l(\lambda_b) + \Gamma_i^l(\lambda_a - \lambda_b)$$

et $|\Delta v_i^l(\lambda_a)| \geq |\Delta v_i^l(\lambda_b)|$ pour tout i . Le découpage change lorsque deux paliers fusionnent, ce qui implique $\Delta v_i^l(\lambda) = 0$. L'index des paliers fusionnés ($\mathcal{N}_k^l$ et $\mathcal{N}_{k+1}^l$) est donné par :

$$k = \arg \min_k (|\Delta v_k^l(\lambda_{l+1}) / \gamma_k^l|)$$

Dans ce cas, nous pouvons calculer λ_l par :

$$\lambda_l = \lambda_{l+1} + |\Delta v_k^l(\lambda_{l+1}) / \gamma_k^l| \quad (3.12)$$

La solution avec $l - 1$ paliers, $v^*(\lambda_l)$ et $\mathcal{N}(\lambda_l)$, est donnée par les équations suivantes :

$$v^*(\lambda_l) = \begin{cases} v_j^l + \beta_j^l(\lambda_l - \lambda_{l+1}), & j = 1, \dots, k-1 \\ v_{j+1}^l + \beta_{j+1}^l(\lambda_l - \lambda_{l+1}), & j = k, \dots, l-1 \end{cases} \quad (3.13)$$

$$\mathcal{N}(\lambda_l) = \begin{cases} \mathcal{N}_j^l, & j = 1, \dots, k-1 \\ \{\mathcal{N}_k^l, \mathcal{N}_{k+1}^l\}, & j = k \\ \mathcal{N}_{j+1}^l, & j = k+1, \dots, l-1 \end{cases} \quad (3.14)$$

Pour $\lambda \geq \lambda_2$, la régularisation de la variation totale devient trop importante, et tous les points forment un seul palier.

En résumé, lorsque λ augmente, la différence entre deux paliers voisins (i.e. $|v_j^* - v_{j+1}^*|$) se réduit, et la variation totale du signal restauré diminue.

Influence de l'hyper-paramètre λ sur le nombre d'extremums du signal restauré

Un bon choix de λ donne une solution qui préserve la variation intrinsèque du signal et élimine l'oscillation locale introduite par le bruit. Cependant, le nombre de paliers $\text{Card}(\mathcal{N}^*(\lambda))$ ne serait pas un bon indicateur de l'oscillation locale du signal restauré, car les paliers neutres n'ont pas participé à la variation totale du signal restauré.

Selon (3.5), seuls les paliers max et min sont considérés dans le calcul de la variation totale. On note $g(\lambda)$ le nombre d'extremums du signal restauré $\mathbf{u}^*(\lambda)$. Un signal bruité a une grande valeur de variation totale et aussi un grand nombre d'extremums locaux en raison de l'oscillation locale du signal bruité. Cependant, un signal bien restauré conservant uniquement la variation intrinsèque du signal original aurait un petit $g(\lambda)$. C'est pourquoi $g(\lambda)$ nous semble un bon indicateur de la qualité de restauration.

Dans cette section, nous allons analyser la relation entre $g(\lambda)$ et λ . Le Théorème 3.2 nous montre la monotonie de $g(\lambda)$ en fonction de λ .

Théorème 3.2. *Il existe une séquence $\mathbf{\Lambda}^g = (\lambda_0^g, \lambda_1^g, \lambda_2^g, \dots) \subset \mathbf{\Lambda}$ avec $\infty = \lambda_0^g > \lambda_1^g > \lambda_2^g > \dots$ telle que :*

- $g(\lambda_a) = g(\lambda_b), \forall \lambda_a, \lambda_b \in [\lambda_{l+1}^g, \lambda_l^g)$.
- $g(\lambda') > g(\lambda''), \forall \lambda' \in [\lambda_{l+1}^g, \lambda_l^g)$ et $\forall \lambda'' \in [\lambda_l^g, \lambda_{l-1}^g)$.

□

Les preuves sont rassemblées page 39.

En outre, nous pouvons directement estimer $g(\lambda)$ à partir de la liste $\mathbf{\Lambda}$. Une méthode simple est proposée dans la Proposition 3.1.

Proposition 3.1. *Nous supposons que deux paliers sont fusionnés ensemble pour $\lambda_l \in \mathbf{\Lambda}$. Dans ce cas, au moins l'un des deux paliers fusionnés est un extremum local. Soit $\Delta g(\lambda_l) = g(\lambda_{l+1}) - g(\lambda_l)$:*

- Si seul un palier est un extremum local, alors le nouveau palier sera aussi un extremum local. $\Delta g(\lambda_l) = 0$.
- Si les deux paliers sont des extremums locaux et ne contiennent pas ni le premier ni le dernier palier, alors le nouveau palier ne sera pas un extremum local. $\Delta g(\lambda_l) = -2$.
- Si les deux paliers sont des extremums locaux et contiennent le premier ou le dernier palier, alors le nouveau palier sera un extremum local. $\Delta g(\lambda_l) = -1$.

Nous proposons une méthode simple et parfaitement intégrable dans nos algorithmes pour calculer $\Delta g(\lambda_l)$. Soient $\{\mathbf{v}^l, \mathcal{N}^l\}$ la solution pour $\lambda = \lambda_{l+1}$, $\mathcal{N}_j^l$ et $\mathcal{N}_{j+1}^l$ les deux paliers à fusionner, et $s^l = s(\mathbf{v}^l)$ suivant (3.7), $\Delta g(\lambda_l)$ est obtenu suivant le Tableau 3.1. $\square$

Conditions		Résultats
$s_{j-1}^l \times s_{j+1}^l$	$ s_{j-1}^l + s_j^l + s_{j+1}^l $	$\Delta g(\lambda_l)$
$\neq 0$		$= - s_{j-1}^l + s_{j+1}^l $
$= 0$	< 2	$= -1$
	$= 2$	$= 0$

TABLEAU 3.1 – Calcul de $\Delta g(\lambda_l)$ en fonction de $s_{j-1}^l \times s_{j+1}^l$ et $s_{j-1}^l + s_j^l + s_{j+1}^l$ avec $\mathcal{N}_j^l$ et $\mathcal{N}_{j+1}^l$ les deux paliers à fusionner.

Diffusion locale du nouvel échantillon

Le signal 1D est une séquence chronologique, et les nouveaux échantillons arrivent au cours du temps. Supposons que nous disposons de n échantillons, le nouvel échantillon z_{n+1} est mesuré à l'instant t_{n+1} avec $t_n < t_{n+1}$. Dans cette section, nous comparons la restauration avec et sans la nouvelle observation.

Soient $\mathbf{u}^* = (u_1^*, \dots, u_n^*) = \arg \min F_n$, la restauration des n premières observations à λ fixé, et $\hat{\mathbf{u}} = (\hat{u}_1, \dots, \hat{u}_n, \hat{u}_{n+1}) = \arg \min F_{n+1}$ avec $F_{n+1} = F_n + (t_{n+1} - t_n)(z_{n+1} - u_{n+1})^2 + \lambda |z_{n+1} - z_n|$ la restauration des $n+1$ premiers points, nous introduisons les notations suivantes :

- Pour $\hat{\mathbf{u}} = \arg \min F_{n+1}$:
 - L'ensemble des niveaux de palier : $\hat{\mathbf{v}} = \{\hat{v}_1, \dots, \hat{v}_{\hat{K}}\}$.
 - L'ensemble des index dans le j^e palier : $\hat{\mathcal{N}}_j = \{\hat{i}_1^j, \dots, \hat{i}_{\hat{n}_j}^j\}$ avec $\hat{n}_j = \text{Card}(\hat{\mathcal{N}}_j)$.
 - L'ensemble des longueurs de palier : $\hat{\mathcal{T}} = \{\hat{\mathcal{T}}_1, \dots, \hat{\mathcal{T}}_{\hat{K}}\}$ avec $\hat{\mathcal{T}}_j = \sum_{i \in \hat{\mathcal{N}}_j} \tau_i$.
- Pour $\mathbf{u}^* = \arg \min F_n$:
 - L'ensemble des niveaux de palier : $\mathbf{v}^* = \{v_1^*, \dots, v_{K^*}^*\}$.
 - L'ensemble des index dans le j^e palier : $\mathcal{N}_j^* = \{i_1^{j,*}, \dots, i_{n_j^*}^{j,*}\}$ avec $n_j^* = \text{Card}(\mathcal{N}_j^*)$.
 - L'ensemble des longueurs de palier : $\mathcal{T}^* = \{\mathcal{T}_1^*, \dots, \mathcal{T}_{K^*}^*\}$ avec $\mathcal{T}_j^* = \sum_{i \in \mathcal{N}_j^*} \tau_i$.

Nous pouvons établir le théorème suivant concernant l'influence de z_{n+1} sur la restauration des n premiers échantillons :

Théorème 3.3. *S'il existe un index j tel que $\text{signe}(v_{j-1}^*(\lambda) - v_j^*(\lambda)) = \text{signe}(v_{K^*}^*(\lambda) - z_{n+1})$ pour $j = 2, \dots, K^*$, alors la nouvelle restauration $\hat{\mathbf{u}}$ satisfait les relations $\hat{u}_i(\lambda) = u_i^*(\lambda)$ pour tout $i < i_1^{j,*}$.* $\square$

L'introduction de l'échantillon z_{n+1} change la dernière partie de u^* , et la diffusion de l'influence de z_{n+1} est bloquée par la jonction de deux paliers dont le signe de la variation $v_{j-1}^*(\lambda) - v_j^*(\lambda)$ correspond à celui de $v_{K^*}^*(\lambda) - z_{n+1}$. La détection de cette jonction bloquante est immédiate.

Pour faire la mise à jour de $\mathbf{u}^*(\lambda)$ avec z_{n+1} , il n'est nécessaire de changer que la dernière séquence, et nous allons introduire la définition suivante :

Définition 3.1. *Une séquence $(u_m^*, \dots, u_n^*)$ est appelée non (droite-)isolée avec $m = i_1^j$ où j est le dernier palier qui satisfait la condition $\text{signe}(v_{j-1}^*(\lambda) - v_j^*(\lambda)) = \text{signe}(v_{K^*}^*(\lambda) - z_{n+1})$.* $\square$

Influence du premier échantillon

Lorsque qu'on applique la méthode sur une fenêtre glissante, le premier point d'une fenêtre est enlevé en passant une fenêtre à l'autre. Nous pouvons établir un théorème similaire concernant l'influence de la suppression du premier échantillon z_1 : la suppression de z_1 change uniquement les premiers points de la restauration.

Théorème 3.4. *S'il existe un index $j \in \{2, \dots, K^*(\lambda)\}$ tel que $\text{signe}(v_{j-1}^*(\lambda) - v_j^*(\lambda)) = \text{signe}(v_1^*(\lambda) - z_1)$, alors la nouvelle construction u^+ sans le premier point satisfait $\hat{u}_i(\lambda) = u_{i-1}^+(\lambda)$ pour tout $i \geq i_1^{j,*}$.* $\square$

En complément de la Définition 3.1, nous introduisons les définitions suivantes :

Définition 3.2. — Une séquence $(u_2^*, \dots, u_p^*)$ est appelée non gauche-isolée avec $p = i_1^j - 1$ où j est le premier palier qui satisfait $\text{signe}(v_{j-1}^*(\lambda) - v_j^*(\lambda)) = \text{signe}(v_1^*(\lambda) - z_1)$.
— La séquence qui n'est pas influencée par l'introduction d'un nouveau point et la suppression du premier point est appelée isolée. $\square$

Indépendance entre les paliers du signal restauré

D'abord, nous proposons de sauvegarder la liste Λ dans un nouveau vecteur de taille $n - 1$: $\Lambda^\circ = (\lambda_1^\circ, \lambda_2^\circ, \dots, \lambda_{n-1}^\circ)$ avec λ_i° la valeur de λ pour laquelle les points i et $i + 1$ fusionnent dans un même palier.

Fixons une valeur de λ , noté $\hat{\lambda}$. Pour chaque palier du signal restauré $u^*(\hat{\lambda})$, nous ajoutons artificiellement un point dans les jonctions de paliers suivant la Définition 3.3.

Définition 3.3. Soient $\hat{\lambda}$ et $\epsilon_\lambda > 0$, nous avons la restauration $v^*(\hat{\lambda}) = \{v_1^*(\hat{\lambda}), \dots, v_l^*(\hat{\lambda})\}$, $\mathcal{N}^*(\hat{\lambda}) = \{\mathcal{N}_1, \dots, \mathcal{N}_l\}$ pour $l = K(\hat{\lambda})$. Soit $s_i = \text{signe}(v_{i+1}^*(\hat{\lambda}) - v_i^*(\hat{\lambda}))$, pour chaque palier $\{z_{\mathcal{N}_j}, \tau_{\mathcal{N}_j}\}$ avec $j = 1, \dots, l$, nous introduisons le palier virtuel $\{z_{\mathcal{N}_j}^+, \tau_{\mathcal{N}_j}^+\}$ avec :

$$z_{\mathcal{N}_i}^+ = \begin{cases} \{z_{\mathcal{N}_1}, v_1^*(\hat{\lambda}) + (\hat{\lambda} + \epsilon_\lambda)/(2s_1)\}, & i = 1. \\ \{v_i^*(\hat{\lambda}) - (\hat{\lambda} + \epsilon_\lambda)/(2s_{i-1}), z_{\mathcal{N}_i}, v_i^*(\hat{\lambda}) + (\hat{\lambda} + \epsilon_\lambda)/(2s_i)\}, & i = 2, \dots, l-1. \\ \{v_l^*(\hat{\lambda}) - (\hat{\lambda} + \epsilon_\lambda)/(2s_{l-1}), z_{\mathcal{N}_l}\}, & i = l. \end{cases}$$

$$\tau_{\mathcal{N}_i}^+ = \begin{cases} \{\tau_{\mathcal{N}_1}, 1\}, & i = 1. \\ \{1, \tau_{\mathcal{N}_i}, 1\}, & i = 2, \dots, l-1. \\ \{1, \tau_{\mathcal{N}_l}\}, & i = l. \end{cases}$$

$\square$

Suivant la dynamique du signal restauré $u^*(\lambda)$ en fonction de λ décrite dans la Section 3.1.3, la variation de la valeur de palier dépend du signe de Δv mais pas sa valeur. Les fusions des petits sous-paliers dans un palier de $u^*(\hat{\lambda})$ (i.e. j^e palier $\mathcal{N}_j^*(\hat{\lambda}) = \{i_1^{j,*}, \dots, i_1^{(j+1)*} - 1\}$) sont indépendantes des points à l'extérieur du palier à condition que les signes des jonctions de paliers (i.e. le premier point du palier $\text{signe}(u_{i_1^{j,*}}^*(\hat{\lambda}) - u_{i_1^{j,*}-1}^*(\hat{\lambda}))$ et le dernier point du palier $\text{signe}(u_{i_1^{(j+1)*}}^*(\hat{\lambda}) - u_{i_1^{(j+1)*}-1}^*(\hat{\lambda}))$) restent identique pour $\lambda \leq \hat{\lambda}$. Pour chaque palier virtuel, ces conditions de bord sont garanties par l'introduction d'un point aux jonctions de paliers. On peut alors découper l'estimation de Λ° dans des sous-problèmes indépendants pour chaque palier virtuel, ce qui est énoncé dans la Proposition suivante :

Proposition 3.2. Soient Λ° l'estimation avec l'ensemble des échantillons $\{z_i, t_i\}_{1, \dots, n}$, $\Lambda_i^\circ = \{\lambda_{i,1}^\circ, \dots\}$ l'estimation avec le i^e palier virtuel $(z_{\mathcal{N}_i}^+, \tau_{\mathcal{N}_i}^+)$ et :

$$\Lambda_i^* = \begin{cases} \{\lambda_{i,1}^\circ, \dots, \lambda_{i,n_i-1}^\circ\}, & i = 1. \\ \{\lambda_{i,2}^\circ, \dots, \lambda_{i,n_i}^\circ\}, & i = 2, \dots, l. \end{cases}$$

on a $\cup_{i=1,\dots,l} \Lambda_i^* = \{\lambda | \lambda \leq \hat{\lambda} \cap \lambda \in \Lambda^\circ\}$. $\square$

La Proposition 3.2 nous permet d'estimer séparément les éléments de Λ° dans chaque palier de $u^*(\hat{\lambda})$ sauf les jonctions de paliers car ces points ont $\lambda^\circ > \hat{\lambda}$. En combinant avec l'influence locale de l'introduction du nouvel échantillon et la suppression du premier échantillon à la restauration (i.e. $u^*(\hat{\lambda})$), l'indépendance entre les paliers implique que ces deux actions apportent une modification locale au vecteur Λ° , respectivement les séquences non-isolées (gauches et droites) de $u^*(\hat{\lambda})$ et les jonctions de paliers de $u^*(\hat{\lambda})$. Cette influence locale nous permet de proposer une estimation en ligne de Λ° .

3.1.4 Proposition des Algorithmes

Dans cette section, nous allons proposer des algorithmes en temps réel pour estimer un bon choix du paramètre λ et la restauration $u^*(\lambda)$ avec un paramètre λ donné.

Estimation de la liste Λ

Les auteurs de [40] ont établi un algorithme de *path* pour estimer Λ en variant le paramètre λ de ∞ à 0. Nous allons revisiter cet algorithme afin d'obtenir simultanément Λ et $g(\lambda)$. L'algorithme d'estimation est présenté dans l'Algorithme 1 : à chaque itération l , λ_l , v^{l-1} et $\mathcal{N}^{l-1}$ sont calculés par (3.12), (3.13) et (3.14) basés sur λ_{l+1} , v^l et $\mathcal{N}^l$, les résultats obtenus de l'itération précédente, et la variation de fonction $g(\lambda)$ est estimée via la Proposition 3.1.

Algorithme 1 : PD-VT : l'algorithme pour estimer tous les éléments de Λ , les restaurations correspondantes et la variation de $g(\lambda)$ pour tout $\lambda \in \Lambda$.

Entrées : $\mathbf{z} = (z_1, \dots, z_n)$, $\tau = (\tau_1, \dots, \tau_n)$

- 1 $\lambda_{n+1} = 0$;
- 2 $v^n = \mathbf{z}, \mathcal{N}^n = \{\{1\}, \dots, \{n\}\}$;
- 3 $g(0) = \text{le nombre d'extremums de } z$;
- 4 **pour** $l = n, \dots, 1$ **faire**
- 5 Calculer β^l , Γ^l et Δv^l par (3.9), (3.10) et (3.11) ;
- 6 $k = \arg \min_k \left| \frac{\Delta v_k^l}{\gamma_k^l} \right|$;
- 7 $\lambda_l = \lambda_{l+1} + \left| \frac{\Delta v_k^l}{\gamma_k^l} \right|$;
- 8 $v_{\{1, \dots, k-1\}}^{l-1} = v_{\{1, \dots, k-1\}}^l + (\lambda_l - \lambda_{l+1}) \beta_{\{1, \dots, k-1\}}^l$;
- 9 $v_{\{k, \dots, l-1\}}^{l-1} = v_{\{k+1, \dots, l\}}^l + (\lambda_l - \lambda_{l+1}) \beta_{\{k+1, \dots, l\}}^l$;
- 10 $\mathcal{N}^{l-1} = \{\mathcal{N}_1^l, \dots, \mathcal{N}_{k-1}^l, \{\mathcal{N}_k^l, \mathcal{N}_{k+1}^l\}, \mathcal{N}_{k+2}^l, \dots, \mathcal{N}_l^l\}$;
- 11 obtenir $\Delta g(\lambda_l)$ par la Proposition 3.1 ;

Sorties : $\Lambda = (\lambda_1, \dots, \lambda_n)$, $v_{all} = (v^1, \dots, v^n)$, $\mathcal{N}_{all} = (\mathcal{N}^1, \dots, \mathcal{N}^n)$ et $g(\lambda)$

Remarque 2. 1. À chaque itération, nous créons des nouveaux vecteurs de taille n . La complexité globale pour estimer Λ , $g(\lambda)$ et les restaurations correspondantes $u^*(\lambda)$ pour $\lambda \in \Lambda$ est de $O(n^2)$.
 2. Si le signal bruité $\mathbf{z}$ a des morceaux constants (i.e. $\exists i, z_i = z_{i+1}$), nous devons replacer ces morceaux constants $\{z, \tau\}_{j, \dots, k}$ par $\{z, \sum_{i=j}^k \tau_i\}$ avant d'appliquer notre algorithme. Sous l'hypothèse de bruit continue, la probabilité d'avoir un morceau constant est proche de 0, mais cela reste un problème pratique dans l'implémentation sur les données réelles.

Estimation de $u^*(\lambda)$ avec un λ donné

Après avoir estimé la liste Λ , nous pouvons facilement obtenir le signal restauré $u^*(\lambda)$ avec λ donné. Nous devons d'abord récupérer le découpage de signal par la plus grande valeur de $\lambda_l \in \Lambda$ telle que $\lambda_l \leq \lambda$, et en suite ajuster les niveaux de palier par :

$$v^*(\lambda) = v^{l-1} + \beta^l(\lambda - \lambda_l)$$

Estimation de λ à partir de $g(\lambda)$

Notre objective est de débruiter les observations z . En connaissant le signal original u_{net} , l'erreur quadratique moyenne de la restauration $u^*(\lambda)$ est définie par :

$$\mathcal{R}(\lambda) = \frac{1}{n} \|u_{net} - u^*(\lambda)\|_2^2 \quad (3.15)$$

Un bon choix de λ devrait avoir une petite valeur de $\mathcal{R}(\lambda)$. La valeur optimale de λ est donnée par :

$$\lambda_{op} = \arg \min \mathcal{R}(\lambda) \quad (3.16)$$

Dans les applications réelles, nous avons peu (ou pas) d'information a priori sur u_{net} . Nous proposons une approche déterministe pour estimer une valeur proche de λ_{op} en temps réel sans aucune information a priori sur le signal original u_{net} ni sur le bruit ϵ .

Avec un découpage $\mathcal{N}$ connu, le débruitage est réalisé par le moyennage des points dans chaque palier. Un bon découpage pour le débruitage doit regrouper suffisamment de points dans chaque palier. La valeur de λ devrait être choisie avec beaucoup de précautions :

- Une petite valeur de λ fournit un découpage fin : le moyennage local sur une petite portion des points limite l'effet de débruitage.
- Par contre, une grande valeur de λ fournit un découpage grossier : la structure intrinsèque de u_{net} est détruite par le moyennage global des points, donc une mauvaise performance de débruitage.

La question essentielle est d'avoir un bon compromis entre ces comportements locaux et globaux. La clé est le nombre de paliers extremums $g(\lambda)$ de $u^*(\lambda)$.

Selon (3.8), le niveau de palier v est linéaire de λ pour $\lambda \in [\lambda_{l+1}, \lambda_l]$, et la vitesse de variation $\beta = \frac{\partial v_j^*}{\partial \lambda}$ dépend de la longueur de palier $\mathcal{T}_j$: un palier court est plus sensible à la variation de λ . Supposons qu'un bruit modéré est ajouté dans le signal original, nous pouvons avoir les constatations suivantes :

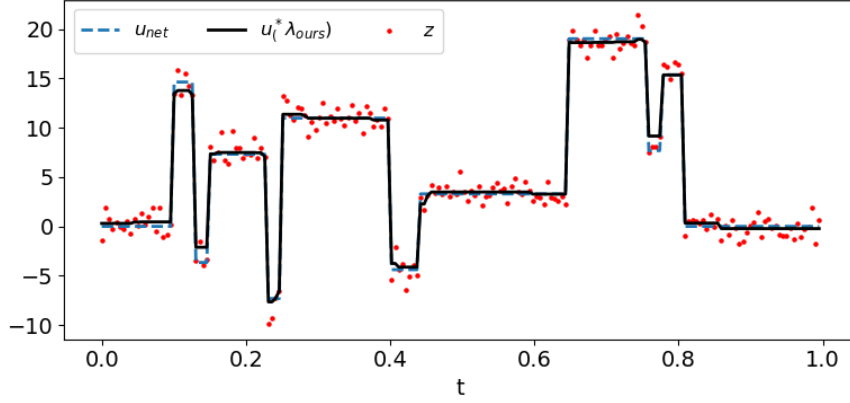
- Pour une petite valeur de λ , le signal restauré a un grand nombre d'extremums à cause de la présence de bruit. Une petite augmentation de λ permet de faire fusionner deux paliers extremums.
- Pour une grande valeur de λ , nous avons des paliers longs liés à la structure intrinsèque du signal. On a besoin d'une grande augmentation de λ pour faire fusionner deux paliers extremums.

En conséquence, les petites valeurs de λ (appelées *régime de débruitage*) font disparaître les extremums introduits par le bruit en moyennant sur un découpage fin, et les grandes valeurs de λ (appelées *régime de destruction*) regroupent les extremums intrinsèques du signal. Ces analyses nous indiquent une structure particulière de $g(\lambda)$: $g(\lambda)$ descend rapidement dans le régime de débruitage avec l'augmentation de λ , mais lentement dans le régime de destruction. Dans la transition entre les deux régimes, le bruit est bien atténué, mais la forme du signal original est préservée.

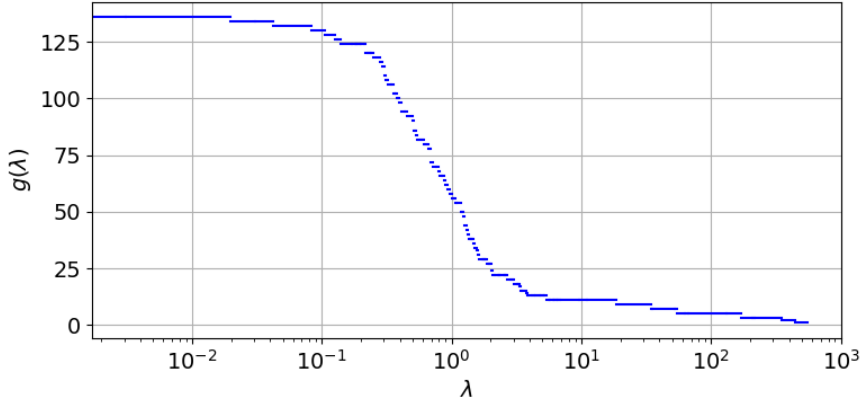
En résumé, un bon choix de λ pourrait correspondre à la discontinuité de la *tendance* (ou du gradient, si tant est que cette notion ait un sens pour une fonction constante par morceaux) de $g(\lambda)$.

Nous présentons ici une simulation avec le signal "block" [13] illustrée sur la Figure 3.2a : z est un signal constant par morceaux (noté u_{net}) dans lequel un bruit gaussien $\epsilon \sim \mathcal{N}(0, 1)$ est ajouté. La fonction $g(\lambda)$ de ce signal est affichée sur la Figure 3.2b. La forme de $g(\lambda)$ correspond bien à nos analyses :

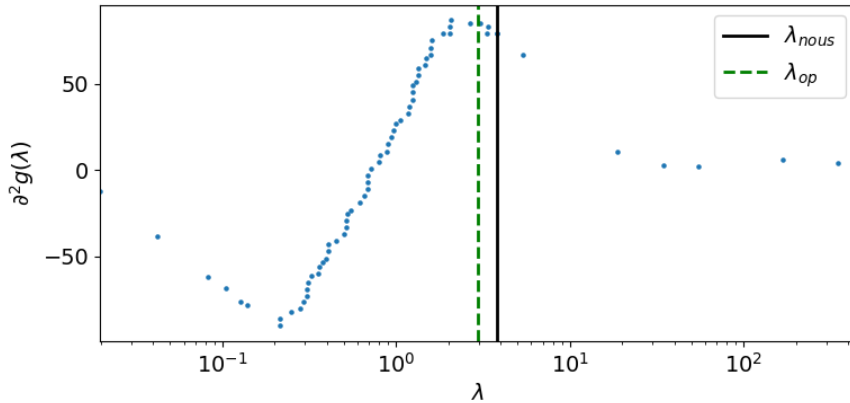
- Régime de débruitage : pour $\lambda < 2$, $g(\lambda)$ diminue rapidement car des petits paliers extremums fusionnent.
- Régime de destruction : pour $\lambda > 7$, $g(\lambda)$ diminue lentement. Pour $\lambda > 500$, la régularisation par la variation totale devient trop importante, et $g(\lambda)$ descend à 1 : $u^*(\lambda)$ a un seul palier $u^*(\lambda) = \text{moyenne}(z)$.



(a) Signal block simulé $z = u_{net} + \epsilon$ avec $\epsilon \sim \mathcal{N}(0, 1)$. SNR=16,91dB. Couleurs : **Echantillons simulé z** , **signal original u_{net}** et **restauration avec notre méthode proposée $u^*(\lambda_{nous})$** .



(b) Le nombre d'extremums locaux ($g(\lambda)$) en fonction de λ .



(c) $\partial^2 g(\lambda)$ avec $\log_{10}(g) = 1$ pour tout $\lambda \in \Lambda^g$. Couleurs : $\lambda_{op} = \arg \min \mathcal{R}(\lambda)$ et **notre approche automatique, $\partial^2 g(\lambda)$ pour tous les éléments de Λ^g** .

FIGURE 3.2 – Exemple simulé : comment choisir un bon λ à partir du nombre d'extremums locaux($g(\lambda)$).

Notre intuition est la suivante : un bon choix de λ devrait se situer dans la dernière partie de la transition entre les régimes de débruitage et de destruction, et la transition correspond à une discontinuité de tendance de $g(\lambda)$ en fonction de $\log(\lambda)$ après une descente raide.

Comme $g(\lambda)$ est monotonement décroissant et constant par morceaux, l'estimation de la tendance de $g(\lambda)$ en fonction de $\log(\lambda)$ n'est pas évidente. Nous appliquons les approximations suivantes : pour

$\lambda \in \Lambda^g$, nous introduisons les opérateurs différentiels numériques :

$$\partial g(\lambda)^+ = g(q\lambda) - g(\lambda)$$

$$\partial g(\lambda)^- = g(\lambda) - g(\lambda/q)$$

avec $q > 1$. $\partial g(\lambda)^-$ et $\partial g(\lambda)^+$ sont en fait l'approximation des dérivés à gauche et à droite de $g(\lambda)$ en fonction de $\log(\lambda)$.

L'approximation de la dérivé seconde de $g(\lambda)$ est donnée par :

$$\partial^2 g(\lambda) = \partial g(\lambda)^+ - \partial g(\lambda)^- \quad (3.17)$$

$\partial^2 g(\lambda)$ avec $\log_{10}(q) = 1$ pour tout $\lambda \in \Lambda^g$ est illustré sur la Figure 3.2c.

Nous proposons la méthode suivante pour estimer la valeur de λ :

- Calculer $\partial^2 g(\lambda)$ via (3.17) pour $\forall \lambda \in \Lambda^g$, et la transition entre les deux régimes est donnée par :

$$\lambda_{trans} = \arg \max \partial^2 g(\lambda)$$

- Ajustement : dans la transition, $\partial^2 g(\lambda)$ a des valeurs similaires, et $\partial^2 g(\lambda)$ décroît rapidement dans le régime de destruction. Un bon choix de λ correspond au dernier élément de $\lambda \in \Lambda$ dans la transition. Nous proposons deux options pour ajuster la valeur de λ :

1. Option semi-automatique : la fin de transition pourrait correspondre au dernier élément de $\lambda \in \{\lambda \in \Lambda^g\} \cap \{g(\lambda) \geq g(\lambda_{trans}) - np \log_{10} q\}$ avec p un paramètre à fixer. Cette méthode nous permet d'adapter la variation du signal restauré aux différentes applications en ajustant la valeur de p .
2. Option automatique : un bon choix de λ pourrait donner par la première descente raide de $\partial^2 g(\lambda)$ pour $\lambda \in \{\lambda \in \Lambda^g\} \cap \{\lambda \geq \lambda_{trans}\}$. Nous proposons d'ajuster λ_{trans} par :

$$\lambda_{nous} = \arg \min_{\lambda \in \{\lambda \in \Lambda^g\} \cap \{\lambda \geq \lambda_{trans}\}} \partial^4 g(\lambda) \quad (3.18)$$

$$\text{avec } \partial^4 g(\lambda_i^g) = \partial^2 g(\lambda_{i+2}^g) - 2\partial^2 g(\lambda_{i+1}^g) + \partial^2 g(\lambda_i^g).$$

Sans mention spécifique, nous utilisons toujours l'option automatique par la suite.

La valeur de q devrait être suffisamment grande afin d'éviter la variation locale du gradient de $g(\lambda)$. Typiquement, $0.5 \leq \log_{10}(q) \leq 1$ peut bien approximer la tendance de $g(\lambda)$. En outre, nous pouvons faire un choix automatique de q : nous proposons que q est égal à la longueur d'un long morceau constant de $g(\lambda)$, et notre proposition est $\log_{10}(q) = \max(\lambda_{i-1}^g/\lambda_i^g)$ sans les deux premiers morceaux constants de $g(\lambda)$.

La complexité pour estimer λ_{nous} à partir de $g(\lambda)$ est en $O(n)$.

Propositions d'améliorations de l'algorithme PD-VT

Nous souhaitons restaurer une grande quantité d'échantillons en temps réel. Cela nous demande une faible complexité temporelle et spatiale, et l'Algorithme 1, en $O(n^2)$, ne respecte pas le cahier de charge. Dans cette section, nous revisitons l'Algorithme 1, et proposons une version plus rapide avec moins d'espace demandé.

À chaque itération de l'Algorithme 1, nous cherchons deux paliers à fusionner et faisons une mise à jour sur tous les paliers en fonction de λ . Cependant, c'est inutile de faire les mises à jour sur les paliers qui n'ont pas fusionné avec ses voisins. Pour ces paliers non-fusionnés, comme β reste identique, la variation du niveau de palier v en fonction de λ reste linéaire avec la même pente (β).

Nous proposons un nouvel algorithme qui calcule et sauvegarde uniquement les informations nécessaires. À la l^e itération de PD-VT, la seule information nécessaire est le minimum de $|\frac{\Delta v}{\Gamma}|$ nous indiquant la valeur de λ_l et la position de fusion. La fusion de deux paliers modifie jusqu'à deux éléments de Γ , respectivement les voisins à gauche et à droite du nouvel palier fusionné.

Soient $\eta = (\eta_1, \dots, \eta_{n-1})$ avec

$$\eta_i = \frac{z_{i+1} - z_i}{\beta_{i+1} - \beta_i}$$

et $\beta = (\beta_1, \dots, \beta_n)$ où β est obtenu par (3.9) avec $\mathbf{z}$ et τ , nous introduisons un vecteur qui représente l'ordre de fusion $n^\circ = \arg \text{sort}(\eta)$. À chaque itération, nous traitons uniquement le premier élément de n° , et reproduisons n° pour la reste des paliers après la modification des valeurs de η des voisins du nouveau palier.

De plus, pour chaque fusion des paliers, nous créons deux nouvelles valeurs essentielles, respectivement λ_{nouveau} la valeur de λ qui provoque la fusion et $\Delta g(\lambda_{\text{nouveau}})$ la variation de $g(\lambda)$ après la fusion. Nous proposons une nouvelle méthode pour sauvegarder ces nouvelles valeurs dans les vecteurs de la taille $n - 1$:

- $\Lambda^\circ = (\lambda_1^\circ, \lambda_2^\circ, \dots, \lambda_{n-1}^\circ)$ avec λ_i° la valeur de λ pour laquelle les points i et $i + 1$ fusionnent dans un même palier.
- $\Delta g^\circ = (\Delta g(\lambda_1^\circ), \Delta g(\lambda_2^\circ), \dots, \Delta g(\lambda_{n-1}^\circ))$

L'algorithme amélioré (PD-VT-2) est présenté dans l'Algorithme 2. À chaque itération, après avoir trouvé la jonction à fusionner indiquée par le premier élément de n° , les nouvelles valeurs sont sauvegardées dans la jonction de ces deux paliers fusionnés.

Remarque 3. Comparé à l'algorithme PD-VT, les points forts de PD-VT-2 sont les suivants :

- Avec une implémentation soigneuse du re-tri de n° à la fin de chaque itération (c.f. Section 3.2), cet algorithme pourrait être implémenté avec une faible complexité temporelle et spatiale, respectivement $O(n \log n)$ et $O(n)$.
- Un grand problème dans l'implémentation sur les données réelles est la fusion de plusieurs paliers pour la même valeur de λ à cause de la précision numérique des micro-processeurs et des capteurs. Avec une légère modification, PD-VT-2 peut gérer cette situation, dans laquelle les premiers éléments de η_{n° sont égaux.

Estimation de $u^*(\lambda)$ et $g(\lambda)$ pour PD-VT-2 Estimer la solution $u^*(\lambda)$ avec un λ donné est légèrement plus compliqué pour PD-VT-2. Avec Λ° estimé par PD-VT-2, nous devons appliquer la méthode suivante :

- Retrouver le découpage de restauration : $j = \{j_1, \dots, j_{l+1}\}$ avec $j_1 = 0$, $j_{l+1} = n$ et $\lambda_{j_i}^\circ > \lambda$ pour $1 < i \leq l$.
- Initialisation : v_{sortie} et $\mathcal{T}_{\text{sortie}}$ deux vecteurs de la taille l . Pour chaque palier $[j_i + 1, j_{i+1}]$ avec $1 \leq i \leq l$: $\mathcal{T}_{\text{sortie},i} = \sum_{m=j_i+1}^{j_{i+1}} \tau_m$ et $v_{\text{sortie},i} = \frac{1}{\mathcal{T}_{\text{sortie},i}} \sum_{m=j_i+1}^{j_{i+1}} \tau_m z_m$.
- Ajuster le niveau des paliers avec λ donné : le niveau du i^e palier est donné par :

$$v_i^*(\lambda) = v_{\text{sortie},i} + \beta_i \lambda$$

avec $\beta_i = \frac{1}{2\mathcal{T}_{\text{sortie},i}}(s_i - s_{i-1})$ et $s = s(v_{\text{sortie}})$.

La complexité pour estimer $u^*(\lambda)$ avec λ donné est en $O(n)$.

L'estimation de $g(\lambda)$ est donnée par :

$$g(\lambda) = 1 - \sum_{i=1}^{n-1} \Delta g_i^\circ \mathbf{1}_{\{\lambda_i^\circ - \lambda\}}$$

avec :

$$\mathbf{1}_\alpha = \begin{cases} 1, & \alpha > 0 \\ 0 & \alpha \leq 0 \end{cases}$$

La complexité pour estimer $g(\lambda)$ est en $O(n \log n)$ car nous devons trier Λ° . Nous pouvons réduire la complexité en $O(n)$ en sauvegardant directement Λ et $\Delta g(\lambda)$ pour $\lambda \in \Lambda$ à chaque itération, au lieu de Δg° .

Algorithme 2 : PD-VT-2 : Une version améliorée de PD-VT.

Entrées : $\mathbf{z} = (z_1, \dots, z_n)$, $\tau = (\tau_1, \dots, \tau_n)$

```

1   $\tilde{v} = z$ ,  $\tilde{\tau} = \tau$ ; // Initialisation
2   $\tilde{\lambda} = (0, \dots, 0)$ ;
3   $s = s(\tilde{v})$  suivant (3.7);
4   $\tilde{\beta} = \beta$  suivant (3.9) avec  $\tilde{\tau}$  et  $s$ ;
5  Index du voisin à gauche  $nl = (0, \dots, n-1)$ ;
6  Index du voisin à droite  $nr = (1, \dots, n+1)$ ;
7  Obtenir  $\tilde{\Gamma}$  et  $\Delta\tilde{v}$  par (3.10) et (3.11) avec  $\beta$  et  $z$ ;
8   $\eta = |\Delta\tilde{v}/\tilde{\Gamma}|$ ;
9   $n^\circ = \text{arg sort}(\eta, \text{ascendant} = \text{vrai})$ ;
10 pour  $l = 1, \dots, n-1$  faire
11      $k = n_l^\circ$ ; // Trouver l'index de la jonction de fusion
12      $v_{new} = \tilde{v}_k + \tilde{\beta}_k(\eta_k - \tilde{\lambda}_k)$ ;
13      $\lambda_{new} = \eta_k$ ;
14      $\tau_{new} = \tilde{\tau}_k + \tilde{\tau}_{nr_k}$ ;
15      $\lambda_k^\circ = \lambda_{new}$ ; // Enregistrer pour les sorties
16     Obtenir  $\Delta g_k^\circ$  par Proposition 3.1;
17      $k_{left}, k_{right} = nl_k, nr_k$ ; // Mettre à jour  $\eta$  pour les prochaines fusions
18      $\tilde{\lambda}_{k_{right}}, \tilde{v}_{k_{right}}, \tilde{\tau}_{k_{right}} = \lambda_{new}, v_{new}, \tau_{new}$ ;
19      $\tilde{\beta}_{k_{right}} = \frac{1}{2\tau_{new}}(s_{k_{right}} - s_{k_{left}})$ ;
20      $nl_{k_{right}}, nr_{k_{left}} = k_{left}, k_{right}$ ;
21      $k_{rr} = nr_{k_{right}}$ ;
22     ; // Le nouveau palier n'est pas le premier palier de la solution
23     si  $k_{left} \geq 1$  alors
24          $v_{left} = \tilde{v}_{k_{left}} + (\lambda_{new} - \tilde{\lambda}_{k_{left}})\tilde{\beta}_{k_{left}}$ ;
25          $\eta_{k_{left}} = \left| \frac{v_{left} - v_{new}}{\tilde{\beta}_{k_{left}} - \tilde{\beta}_{k_{right}}} \right| + \lambda_{new}$ ;
26     ; // Le nouveau palier n'est pas le dernier palier de la solution
27     si  $k_{right} < n$  alors
28          $v_{rr} = \tilde{v}_{k_{rr}} + (\lambda_{new} - \tilde{\lambda}_{k_{rr}})\tilde{\beta}_{k_{rr}}$ ;
29          $\eta_{k_{right}} = \left| \frac{v_{new} - v_{rr}}{\tilde{\beta}_{k_{right}} - \tilde{\beta}_{k_{rr}}} \right| + \lambda_{new}$ ;
30     Re-trier  $n_{\{i+1, \dots, n-1\}}^\circ$  suivant  $\eta_{n_{\{i+1, \dots, n-1\}}^\circ}$ ;
Sorties :  $\Lambda^\circ = (\lambda_1^\circ, \dots, \lambda_{n-1}^\circ)$  et  $\Delta g^\circ = (\Delta g_1, \dots, \Delta g_{n-1})$ 

```

Implémentation en ligne de PD-VT-2

Dans cette section, basée sur les modifications locales introduites par le nouvel échantillon, nous allons proposer une implémentation en ligne de l'Algorithme 2.

Nous allons noter $(\Lambda^{\circ, n}, \Delta g^{\circ, n})$, deux vecteurs de taille $n-1$, les sorties de l'Algorithme 2 avec n échantillons. Avec le nouvel échantillon (z_{n+1}, t_{n+1}) , les nouveaux résultats $(\Lambda^{\circ, n+1}, \Delta g^{\circ, n+1})$ basés sur $n+1$ échantillons peuvent être obtenus par une modification partielle de $(\Lambda^{\circ, n}, \Delta g^{\circ, n})$. Par la suite, nous allons discuter uniquement de l'estimation en ligne de $\Lambda^{\circ, n+1}$ à partir de $\Lambda^{\circ, n}$ en détail. Pour la simplicité de présentation, nous notons $\Lambda^\circ = \text{PD-VT-2}(\{z\}, \{\tau\})$ une application de l'Algorithme 2 sur une séquence de signal $(\{z\}, \{\tau\})$.

Après avoir choisi une valeur de λ , notée $\hat{\lambda}$, nous pouvons avoir la restauration $u^*(\hat{\lambda})$ pour les n premiers points. Suivant le Théorème 3.3, l'introduction d'un nouvel échantillon change uniquement la séquence non-isolée de $u^*(\hat{\lambda})$. De plus, la Proposition 3.2 implique que $\lambda_i^{\circ, n} = \lambda_i^{\circ, n+1}$ si $\lambda_i^{\circ, n} \leq \hat{\lambda}$ pour la séquence isolée, et la fusion de paliers de la nouvelle $(n+1)$ -point restauration $\hat{u}(\hat{\lambda})$ (i.e. $\{\lambda_i^{\circ, n+1} > \hat{\lambda}\}$) dépend de la valeur de z_{n+1} . Pour faciliter la compréhension de la procédure de modification, les

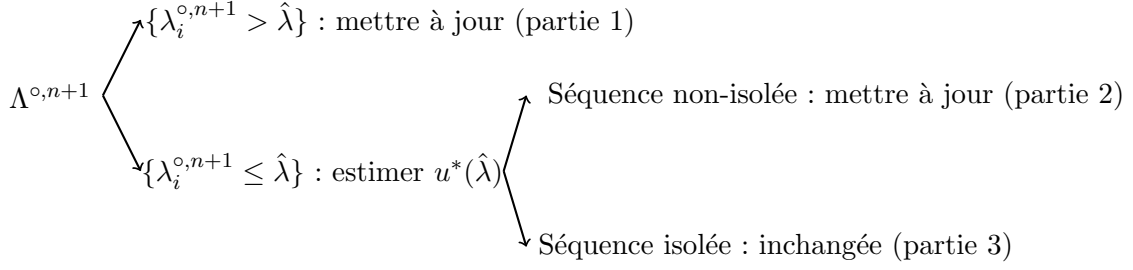


FIGURE 3.3 – Illustration de l’implémentation en ligne : les parties changées et inchangées de $\Lambda^{\circ, n+1}$ avec l’introduction d’un nouvel échantillon.

différentes parties de l’estimation en ligne sont présentées sur la Figure 3.3.

La valeur $\hat{\lambda}$ choisie est un *point de découpe* de $\Lambda^{\circ, n+1}$: $\{\lambda^{\circ, n+1} \leq \hat{\lambda}\}$ et $\{\lambda^{\circ, n+1} > \hat{\lambda}\}$ seront traités séparément. On note m et j^* respectivement le premier point et le premier palier de la séquence non-isolée de $u^*(\hat{\lambda})$ suivant la Définition 3.1.

On traite d’abord la partie inchangée de $\Lambda^{\circ, n+1}$ (partie 3 de la Figure 3.3) : tout $\lambda_i^{\circ, n+1} \leq \hat{\lambda}$ reste identique pour $i < m$. Formellement, nous avons l’équation suivante :

$$\lambda_{\{\lambda_1^{\circ, n+1} \leq \hat{\lambda}\}}^{\circ, n+1} = \lambda_{\{\lambda_1^{\circ, n} \leq \hat{\lambda}\}}^{\circ, n}$$

Pour la séquence non-isolée (partie 2 de la Figure 3.3), $\Lambda^a = \lambda_{\{\lambda_m^{\circ, n+1} \leq \hat{\lambda}\}}^{\circ, n+1}$ peut être estimé par PD-VT-2($z_{\{m, \dots, n+1\}}^+, \tau_{\{m, \dots, n+1\}}^+$) où :

$$\begin{aligned} \text{— } z_{\{m, \dots, n+1\}}^+ &= \{v_{j^*}^*(\hat{\lambda}) - \frac{\hat{\lambda} + \epsilon_\lambda}{2\text{signe}(v_{j^*} - v_{j^*-1})}, z_{\{m, \dots, n+1\}}\} \\ \text{— } \tau_{\{m, \dots, n+1\}}^+ &= \{1, \tau_{\{m, \dots, n+1\}}\} \end{aligned}$$

avec $\epsilon_\lambda > 0$.

Les séquences isolées et non-isolées sont assemblées dans $\Lambda^{temp} = \{\lambda_1^{temp}, \dots, \lambda_n^{temp}\}$ avec :

$$\lambda_i^{temp} = \begin{cases} \lambda_i^{\circ, n}, & i < m. \\ \lambda_{i-m+2}^a, & i \geq m. \end{cases}$$

Il nous reste la modification de $\{\lambda_i^{\circ, n+1} > \hat{\lambda}\} = \{\lambda_i^{temp} > \hat{\lambda}\}$ (partie 1 de la Figure 3.3). Soit $b = \{\lambda^{\circ, n+1} > \hat{\lambda}\}$ contenant en fait toutes les jonctions de $\hat{u}(\hat{\lambda})$, nous pouvons avoir $\Lambda^b = \lambda_b^{\circ, n+1} = \text{PD-VT-2}(\hat{v}(\hat{\lambda}), \hat{\mathcal{T}}(\hat{\lambda}))$.

Finalement, la solution $\Lambda^{\circ, n+1}$ est assemblée de la manière suivante :

$$\lambda_i^{\circ, n+1} = \begin{cases} \lambda_i^{temp}, & \text{if } \lambda_i^{temp} \leq \hat{\lambda}. \\ \lambda_{p(i)}^b, & \text{if } \lambda_i^{temp} > \hat{\lambda}. \end{cases} \quad (3.19)$$

avec $p : \mathbb{Z} \rightarrow \mathbb{Z}$ qui nous donne les index i dans le vecteur b .

Le vecteur $\Delta g^{\circ, n+1}$ est calculé à partir de $\Delta g^{\circ, n}$ de la même manière que $\Lambda^{\circ, n+1}$ (à partir de $\Lambda^{\circ, n}$).

La version en ligne de PD-VT-2 est résumée dans l’Algorithme 3 : PD-VT-2 est appliqué deux fois dans une petite partie de données (respectivement Λ^a et Λ^b), et les résultats sont assemblés en suivant (3.19).

Nous obtenons $u^*(\hat{\lambda})$ en $O(n)$, Λ^a en $O((n-m) \log(n-m))$ et Λ^b en $O(l_b \log l_b)$ avec $l_b = \text{Card}(\Lambda^b)$. La complexité globale dépend du choix du point de découpe $\hat{\lambda}$, un compromis entre la complexité de Λ_a et Λ_b :

- Une grande valeur de $\hat{\lambda}$ rend la restauration $u^*(\hat{\lambda})$ peu fluctuante, et la séquence non-isolée peut être longue, même aussi longue que z (c’est-à-dire $m = 1$). Dans ce cas, le calcul de Λ_a est en $O(n \log n)$.

- La reconstruction avec une petite valeur de $\hat{\lambda}$ a une courte séquence non-isolée. Cependant, tout $\lambda > \hat{\lambda}$ pour $\lambda \in \Lambda^{\circ,n}$ doit être recalculé. Dans le pire des cas, tous les éléments de $\Lambda^{\circ,n}$ doivent être recalculés en $O(n \log n)$.

Un choix correct de $\hat{\lambda}$ pourrait nous fournir une courte séquence non-isolée et une petite liste de Λ_b , c'est-à-dire $(n - m) \ll n$ et $l_b \ll n$. Dans ce cas, la complexité de l'implémentation en ligne est en $O(n)$. Nous proposons choisir la valeur de λ estimée par notre approche déterministe $\hat{\lambda} = \lambda_{\text{nous}}$ ou un peu plus grande (i.e. $\hat{\lambda} = 2\lambda_{\text{nous}}$).

Pour avoir $g(\lambda)$ en $O(n)$, nous proposons d'enregistrer $\Delta g(\lambda)$ suivant l'ordre de Λ . Pour l'implémentation en ligne, les calculs de $\Delta g(\Lambda^{n+1})$ et Λ^{n+1} (la liste ordonnée de $\Lambda^{\circ,n+1}$) pourrait être vus comme une insertion de deux petites listes ordonnées (Λ des parties 1 et 2 de la Figure 3.3) dans une grande liste ordonnée (Λ of part 3 in Figure 3.3).

Algorithme 3 : Implémentation en ligne de l'algorithme PD-VT-2

Entrées : $(z_1, \dots, z_n), (\tau_1, \dots, \tau_n), (z_{n+1}, \tau_{n+1})$
Entrées : $\Lambda^{\circ,n}, \Delta g^{\circ,n}, \hat{\lambda}$

- 1 Estimer la restauration $u^*(\hat{\lambda})$ de $\{(z_1, \dots, z_n), (\tau_1, \dots, \tau_n)\}$;
- 2 Trouver la séquence non-isolée $(m, \dots, n)$ de l'estimation $u^*(\hat{\lambda})$;
- 3 $(\Lambda^a, \Delta g^a) = \text{PD-VT-2}(z_{\{m, \dots, n+1\}}^+, \tau_{\{m, \dots, n+1\}}^+)$;
- 4 $\Lambda^{\circ,n+1} = (\lambda_1^{\circ,n+1}, \dots, \lambda_{n+1}^{\circ,n+1}) = \{\Lambda_{\{1, \dots, m-1\}}^{\circ,n}, \Lambda_{\{2, \dots\}}^a\}$;
- 5 $\Delta g^{\circ,n+1} = \{\Delta g_{\{1, \dots, m-1\}}^{\circ,n}, \Delta g_{\{2, \dots\}}^a\}$;
- 6 $b = \{\lambda^{\circ,n+1} > \hat{\lambda}\}$;
- 7 $(\lambda_b^{\circ,n+1}, \Delta g_b^{\circ,n+1}) = \text{PD-VT-2}(\hat{u}(\hat{\lambda}), \hat{\tau}(\hat{\lambda}))$;

Sorties : $\Lambda^{\circ,n+1}$ et $\Delta g^{\circ,n+1}$

Adaptation de l'implémentation en ligne aux fenêtres glissantes

Dans cette section, nous allons adapter l'Algorithme 3 aux fenêtres glissantes. Cette adaptation rend certaines applications plus efficaces (comme par exemple les applications aux signaux non-stationnaires).

Une fenêtre glissante de la taille m avec $m < n$ est notée $w_i^m = [i, i + m - 1]$. Pour deux fenêtres successives w_i^m et w_{i+1}^m , w_{i+1}^m est, en effet, w_i^m dans laquelle le premier point (z_i, τ_i) est enlevé et un nouveau point (z_{i+m}, τ_{i+m}) est ajouté à la fin de la séquence.

Nous notons $\Lambda^{\circ,i}$ et $\Delta g^{\circ,i}$, deux vecteurs de la taille $m - 1$, qui représentent la solution PD-VT-2 de la fenêtre w_i^m . Les résultats de w_{i+1}^m peuvent être obtenus par une mise à jour des résultats de w_i^m . Nous devons ajouter une étape pour mettre à jour la séquence non-gauche-isolée dans l'Algorithme 3. L'algorithme proposé est présenté dans l'Algorithme 4.

Algorithme 4 : Implémentation en ligne de PD-VT-2 adaptée aux fenêtres glissantes w_m^i

Entrées : $(z_i, \dots, z_{i+m-1}), (\tau_i, \dots, \tau_{i+m-1}), (z_{i+m}, \tau_{i+m})$

Entrées : $\Lambda^{\circ, i}, \Delta g^{\circ, i}, \hat{\lambda}$

```

1 Estimer la restauration  $u^*(\hat{\lambda})$  de  $\{(z_i, \dots, z_{i+m-1}), (\tau_i, \dots, \tau_{i+m-1})\}$  ;
2 Trouver la séquence non-isolée  $(l, \dots, m)$  de l'estimation  $u^*(\hat{\lambda})$  ;
3  $j_l^*$  est le numéro du premier palier de la séquence non-droite-isolée ;
4 Trouver la séquence non-gauche-isolée  $(2, \dots, p)$  de l'estimation  $u^*(\hat{\lambda})$ ;
5  $j_p^*$  est le numéro du dernier palier de la séquence non-gauche-isolée ;
6 si  $p < l - 2$  alors
7     ; // MAJ de la séquence non-droite-isolée
8      $z_a^+ = \{v_{j_l^*}^*(\hat{\lambda}) - \frac{\hat{\lambda} + \epsilon_\lambda}{2\text{signe}(v_{j_l^*}^* - v_{j_l^*-1}^*)}, z_{\{l+i-1, \dots, m+i\}}\}$ ;
9      $\tau_a^+ = \{1, \tau_{\{l+i-1, \dots, m+i\}}\}$  ;
10     $(\Lambda^a, \Delta g^a) = \text{PD-VT-2}(z_a^+, \tau_a^+)$  ;
11    ; // MAJ de la séquence non-gauche-isolée
12     $z_b^+ = \{z_{\{i+1, \dots, i+p-1\}}, v_{j_p^*}^*(\hat{\lambda}) + \frac{\hat{\lambda} + \epsilon_\lambda}{2\text{signe}(v_{j_p^*+1}^* - v_{j_p^*}^*)}\}$  ;
13     $\tau_b^+ = \{\tau_{\{i+1, \dots, i+p-1\}}, 1\}$  ;
14     $(\Lambda^b, \Delta g^b) = \text{PD-VT-2}(z_b^+, \tau_b^+)$  ;
15     $\Lambda^{\circ, i+1} = \{\Lambda_{\{1, \dots, p-1\}}^b, \Lambda_{\{p+1, \dots, l-1\}}^{\circ, i}, \Lambda_{\{2, \dots\}}^a\}$  ;
16     $\Delta g^{\circ, i+1} = \{\Delta g_{\{1, \dots, p-1\}}^b, \Delta g_{\{p+1, \dots, l-1\}}^{\circ, i}, \Delta g_{\{2, \dots\}}^a\}$  ;
17     $d = \{\lambda^{\circ, n+1} > \hat{\lambda}\}$  ;
18     $(\lambda_d^{\circ, i+1}, \Delta g_d^{\circ, i+1}) = \text{PD-VT-2}(\hat{u}(\hat{\lambda}), \hat{\mathcal{T}}(\hat{\lambda}))$  ;
19 sinon
20     $(\lambda_d^{\circ, i+1}, \Delta g_d^{\circ, i+1}) = \text{PD-VT-2}(\{z_i, \tau_i\}_{i+1, \dots, i+m})$  ; // Pas d'élément isolé dans cette
    fenêtre
21    ;
Sorties :  $\Lambda^{\circ, i+1}$  et  $\Delta g^{\circ, i+1}$ 

```

3.2 Implémentation

Notre objectif est d'avoir une méthode de restauration en temps réel, donc une bonne implémentation des algorithmes joue un rôle primordial dans la performance de notre méthode. Dans cette section, nous présentons d'abord une structure de données : l'arbre binaire de recherche, et ensuite l'implémentation de l'Algorithme 2 en $O(n \log n)$.

3.2.1 Arbre binaire de recherche

Définition

L'*arbre binaire de recherche* (ABR) est une structure de données particulièrement adaptée aux modifications itératives dans une suite ordonnée. Nous faisons d'abord quelques rappels sur un arbre binaire :

- Un arbre est un ensemble de nœuds.
- Un nœud est représenté par une étiquette.
- Le nœud initial est appelé *racine*.
- Chaque nœud (à part la racine) a un seul *parent* au niveau supérieur.
- Chaque nœud a au maximum deux fils au niveau inférieur, appelé respectivement fils *gauche* et *droit*.
- Un nœud n'ayant aucun fils est appelé *feuille*.

Un ABR (c.f. Figure 3.4) est un arbre binaire avec deux règles supplémentaires :

- L'étiquette du fils gauche et celles de ses descendants sont inférieures à celle du nœud parent.
- L'étiquette du fils droit et celles de ses descendants sont supérieures à celle du nœud parent.

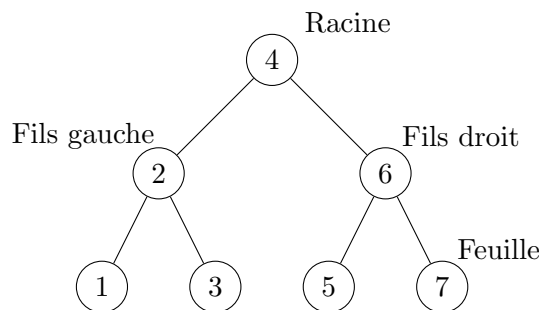


FIGURE 3.4 – Illustration d'un arbre binaire de recherche. Un nœud est représenté par un cercle, et l'étiquette d'un nœud correspond au chiffre affiché dans le cercle.

Opérations

Les opérations élémentaires sur un nœud sont notées de la manière suivante :

- L'étiquette d'un nœud x est donnée par $\text{étiquette}(x)$.
- Le fils gauche est donné par $\text{gauche}(x)$, et le fils droit est donné par $\text{droit}(x)$.
- Dans le cas où un nœud x n'est pas défini, $x = \text{null}$.

Nous allons présenter quatre opérations sur un ABR.

Recherche un nœud Pour chercher un nœud avec une étiquette particulière v dans un ABR, il faut appliquer la fonction $\text{recherche}(\text{racine}, v)$ (c.f. Algorithme 5). La complexité pour chercher un élément dans un ABR est en $O(\log n)$ en moyenne et en $O(n)$ dans le pire des cas.

Algorithme 5 : recherche(x, v)

```

1 si  $x = null$  ou  $v = \text{étiquette}(x)$  alors
2   | retourner  $x$ 
3 si  $v < \text{étiquette}(x)$  alors
4   | retourner recherche(gauche( $x$ ),  $v$ )
5 sinon
6   | retourner recherche(droit( $x$ ),  $v$ )

```

Insérer un nœud Pour insérer un nœud avec l'étiquette v dans un ARB, il faut appliquer Insert(v) (c.f. Algorithme 6). La construction d'un ARB à partir d'un vecteur des éléments consiste à insérer un par un les éléments dans un ARB initialement vide. La complexité pour insérer un élément dans un ARB est en $O(\log n)$ en moyenne et en $O(n)$ dans le pire des cas. La construction d'un arbre est en $O(n \log n)$ en moyenne.

Algorithme 6 : Insert(v)

```

1 si  $x = null$  alors
2   | Initialiser un arbre avec une racine avec l'étiquette  $v$  et retourner ;
3 tant que Vrai faire
4   | si  $v = \text{étiquette}(x)$  alors
5   |   | retourner  $x$ 
6   | si  $v < \text{étiquette}(x)$  alors
7   |   | si gauche( $x$ ) =  $null$  alors
8   |   |   | Créer un nœud  $n_G$  avec l'étiquette  $v$  ;
9   |   |   | gauche( $x$ ) =  $n_G$  ;
10  |   |   | Retourner  $n_G$ ;
11  |   | sinon
12  |   |   |  $x = \text{gauche}(x)$ 
13  | sinon
14  |   | si droit( $x$ ) =  $null$  alors
15  |   |   | Créer un nœud  $n_D$  avec l'étiquette  $v$  ;
16  |   |   | droit( $x$ ) =  $n_D$  ;
17  |   |   | Retourner  $n_D$ ;
18  |   | sinon
19  |   |   |  $x = \text{droit}(x)$ 

```

Trouver le minimum Le nœud ayant l'étiquette minimale se trouve à l'extrémité gauche de l'arbre, et il est donné par min(racine) (c.f. Algorithme 7). La complexité pour trouver le minimum est en $O(\log n)$ en moyenne et en $O(n)$ dans le pire des cas.

Algorithme 7 : min(x)

```

1 si gauche( $x$ )  $\neq null$  alors
2   | retourner min(gauche( $x$ ))
3 sinon
4   | retourner  $x$ 

```

Supprimer un nœud Pour supprimer un nœud ayant une étiquette v , il faut d'abord retrouver le nœud concerné par $x = \text{recherche}(\text{racine}, v)$ et ensuite le remplacer si nécessaire. Nous discutons des cas suivants :

- Le nœud x est une feuille : $\text{gauche}(\text{parent}(x)) = \text{null}$ si x est un fils gauche, ou $\text{droit}(\text{parent}(x)) = \text{null}$ sinon.
- Le nœud x a un seul fils : remplacer le nœud x par son fils.
- Le nœud x a deux fils : rechercher le minimum dans le sous-arbre de son fils droit $n_{D,\min} = \min(\text{droit}(x))$, ensuite remplacer x par $n_{D,\min}$ et supprimer $n_{D,\min}$.

Après avoir trouvé le nœud, la complexité pour supprimer une feuille ou un nœud avec un seul fils est en $O(1)$, tandis que celle pour un nœud avec deux fils est en $O(\log n)$ en moyenne et en $O(n)$ dans le pire des cas.

3.2.2 Détails de l'implémentation de PD-VT-2

Le point clé pour implémenter l'Algorithme 2 en $O(n \log n)$ est le vecteur n° qui représente l'ordre de η . Il faut re-trier efficacement n° une fois que les mises à jour de η sont faites. Nous proposons de sauvegarder n° dans un ARB suivant les valeurs de η et de modifier l'arbre à chaque itération. Ci-joint les détail d'implémentation :

- Après avoir généré le vecteur η (ligne 8 de l'Algorithme 2), nous construisons un ABR avec tous les éléments (i, η_i) . En effet, chaque nœud contient un index (i.e. i), et l'étiquette de ce nœud est donnée par η_i . Nous notons cet arbre $\tilde{n}$.
- L'index de la jonction de paliers à fusionner est donnée par $\arg \min \eta$ (ligne 11 de l'Algorithme 2). Cet index est obtenu par $x = \min(\text{racine}(\tilde{n}))$, et ce nœud x est ensuite supprimé de l'arbre $\tilde{n}$. La complexité de cette étape est en $O(\log n)$ à chaque itération car la suppression est réalisée en $O(1)$.
- Après la fusion de paliers, nous devons modifier au plus deux élément de η (i.e. les voisins du nouveau palier, lignes 22-29 de l'Algorithme 2). Soit i l'index à modifier, nous proposons de faire les mises à jour de la manière suivante :
 - Supprimer le nœud (i, η_i) avant la modification de η .
 - Modifier η_i
 - Réinsérer le nœud (i, η_i) . L'étiquette de ce nœud est à jour car η_i a été modifié.

La complexité de cette étape est en $O(3 \log n)$ pour un voisin à chaque itération.

Au total, la complexité en moyenne de PD-VT-2 est en $O(n \log n + n(\log n + 2 \times 3 \log n))$, donc en $O(8n \log n)$.

3.3 Performance des méthodes proposées

Dans cette section, nous évaluons la performance des différents algorithmes proposés. D'abord, nous comparons la restauration avec le paramètre λ_{nous} (3.18) avec les différentes méthodes existantes. Ensuite, nous comparons le temps d'exécution des différentes implémentations de PD-VT.

3.3.1 Métriques

Nous utilisons l'erreur quadratique moyenne $\mathcal{R}(\lambda)$ (3.15) entre la restauration et le signal original pour évaluer un candidat de λ . La valeur optimale λ_{op} est donnée par (3.16).

Pour comparer la performance de deux candidats (λ_1 et λ_2), nous appliquons le critère suivant :

$$d(\lambda_1, \lambda_2) = \mathcal{R}(\lambda_1) - \mathcal{R}(\lambda_2)$$

$d(\lambda, \lambda_{op}) \geq 0, \forall \lambda \in \mathbb{R}^+$ car λ_{op} fournit le minimum de $\mathcal{R}(\lambda)$. Plus petit $d(\lambda, \lambda_{op})$ est, meilleure est l'estimation λ .

Cependant, ce n'est pas évident de fixer un seuillage sur $d(\lambda, \lambda_{op})$ entre une estimation correcte et inacceptable de λ . Cela dépend de la variance du bruit et de la forme de signal original. Une méthode alternative est de comparer avec les méthodes existantes.

3.3.2 Méthodes existantes

Validation croisée

La validation croisée est une méthode statistique pour évaluer comment un modèle se comporte sur les données indépendantes. La sélection de modèle pour la VT-restauration revient à sélectionner le meilleur hyper-paramètre.

Nous présentons comment appliquer la validation croisée (en anglais cross validation (cv)) dans le cadre de la VT-restauration du signal. On répartit aléatoirement les échantillons en k parties, toutes les parties ayant pratiquement le même nombre de points. On note $(\mathbf{z}^i, \mathbf{t}^i)$ pour la i^e partie et $m_i = \text{Card}(\mathbf{z}^i)$. Une restauration est estimée sans une partie d'échantillons avec λ donné :

$$\mathbf{u}^{-i}(\lambda) = \arg \min_u \{F(\mathbf{z}^{-i}, u, \mathbf{t}^{-i}, \lambda)\}$$

où $(\mathbf{z}^{-i}, \mathbf{t}^{-i})$ les échantillons de signal sans la i^e partie.

La prédiction $\hat{u}^{-i}(t)$ à l'instant t est donnée par l'interpolation linéaire de $(\mathbf{u}^{-i}(\lambda), \mathbf{t}^{-i})$. L'erreur empirique de $\mathbf{u}^{-i}(\lambda)$ est estimée sur les données qui n'ont pas participé à la construction du modèle (i.e. la i^e partie) :

$$e^{-i}(\lambda) = \frac{1}{m_i} \sum_{(z,t) \in (\mathbf{z}^i, \mathbf{t}^i)} (z - \hat{u}^{-i}(t))^2$$

Au final, nous pouvons sélectionner le paramètre λ par :

$$\lambda_{cv} = \arg \min_{\lambda \in \mathbb{R}^+} \sum_{i=1}^K e^{-i}(\lambda) \quad (3.20)$$

Remarque 4. Une séquence des échantillons est en fait une série chronologique. La découpe aléatoire des données n'est pas légitime pour la série chronologique [3], car ces différentes parties ne sont plus indépendantes. Cependant, nous nous intéressons à la performance de l'interprétation du modèle mais pas à la capacité de prédiction sur des données indépendantes. Donc la découpe aléatoire des données est justifiée.

SURE et AUT

Certaines méthodes efficaces sont basées sur une variance du bruit σ^2 connue parmi lesquelles les méthodes Stein Unbiased Risk Estimation (SURE) [37] et Adaptive Universal Threshold (AUT) [35] sont particulièrement adaptées au signal 1D grâce à leurs bonnes performances et facilité d'implémentation.

- Stein Unbiased Risk Estimation (SURE) [37] est un estimateur non-biaisé de $|u^*(\lambda) - u_{net}|_2^2$. Les auteurs de [40] ont montré que le nombre de degrés de liberté de la VT-restauration d'un signal 1D est le nombre de paliers de la restauration, ce qui nous donne :

$$\text{SURE}(\lambda) = |z - u^*(\lambda)|_2^2 + 2\sigma^2 \text{Card}(v^*(\lambda)) - n\sigma^2$$

avec $\text{Card}(v^*(\lambda))$ le nombre de paliers.

La proposition de λ est donnée par :

$$\lambda_{\text{SURE}} = \arg \min \text{SURE}(\lambda)$$

- Adaptive Universal Threshold (AUT) [35] sélectionne le paramètre λ basé sur le comportement de la restauration avec un pré-choix élégant de λ . Le pré-choix de λ est donné par $\lambda_N = \sigma/2\sqrt{n \log \log n}$, et la restauration avec λ_N nous donne une estimation du nombre de paliers $\hat{K}$. La proposition finale de λ est donnée par :

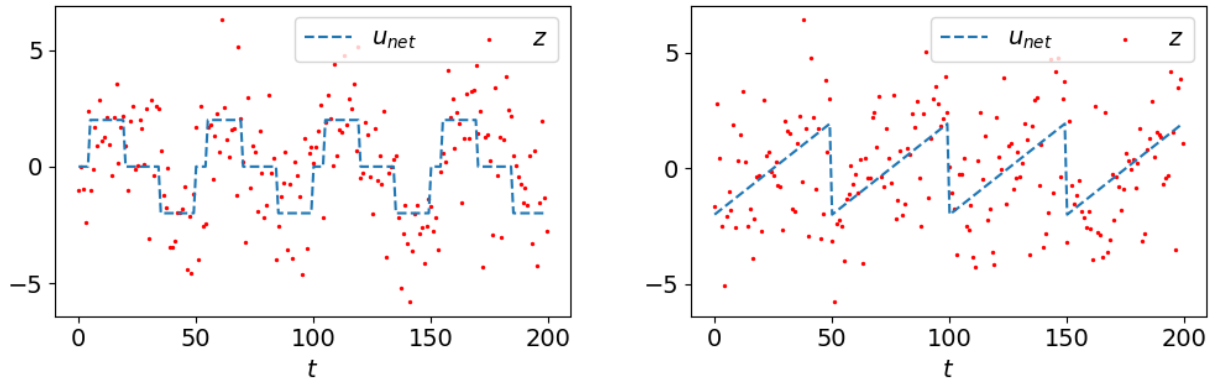
$$\lambda_{AUT} = \frac{\sigma}{2} \sqrt{\frac{n}{\hat{K}} \log \log \frac{n}{\hat{K}}}$$

3.3.3 Résultats

Restauration sous les différents niveaux de bruit

D'abord, nous nous intéressons à la performance des méthodes sous le bruit gaussien avec différentes variances σ^2 . Nous avons testé deux signaux périodiques illustrés sur la Figure 3.5 : constant par morceaux et linéaire par morceaux. Un bruit gaussien $\epsilon \sim \mathcal{N}(0, \sigma^2)$ est ajouté dans les signaux simulés. Deux exemples de signaux simulés sont illustrés sur la Figure 3.5.

Nous comparons notre proposition λ_{nous} , estimée par (3.18) avec $\log_{10} q \in \{0,5;1\}$ et le choix automatique de q , avec la proposition estimée par la validation croisée (3.20). Pour chaque type de signal et chaque $\sigma \in \{0,5;1;1,5;2;2,5;3\}$, nous avons effectué 500 simulations.



(a) Constant par morceaux

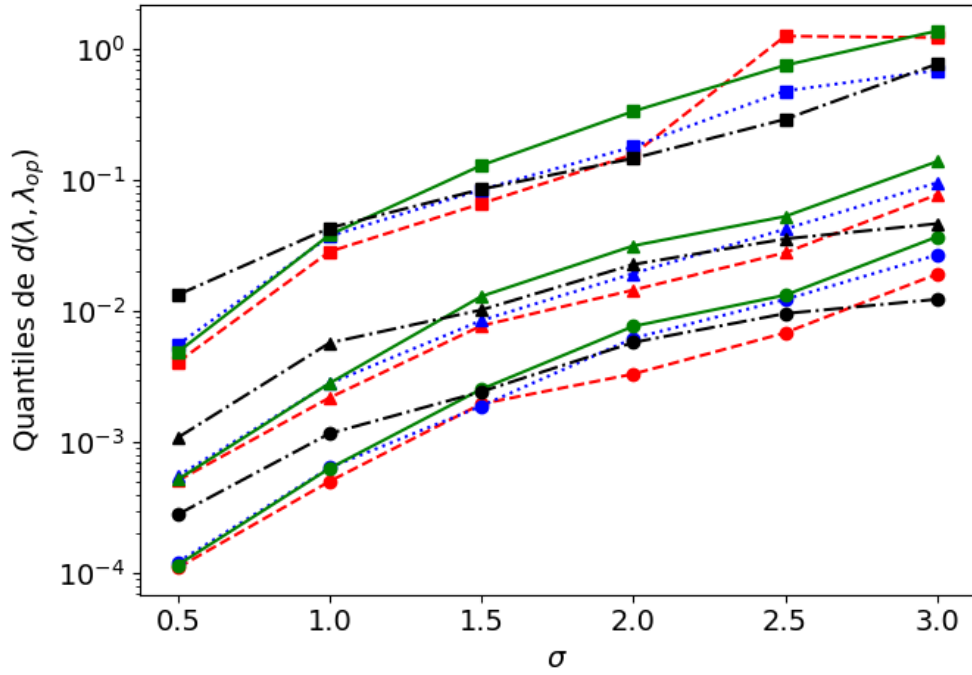
(b) Linéaire par morceaux

FIGURE 3.5 – Illustration des 2 types de signal périodique simulé (u_{net}). Exemples de $z = u_{net} + \epsilon$ avec $\epsilon \sim \mathcal{N}(0, 4)$ sont affichés en rouge.

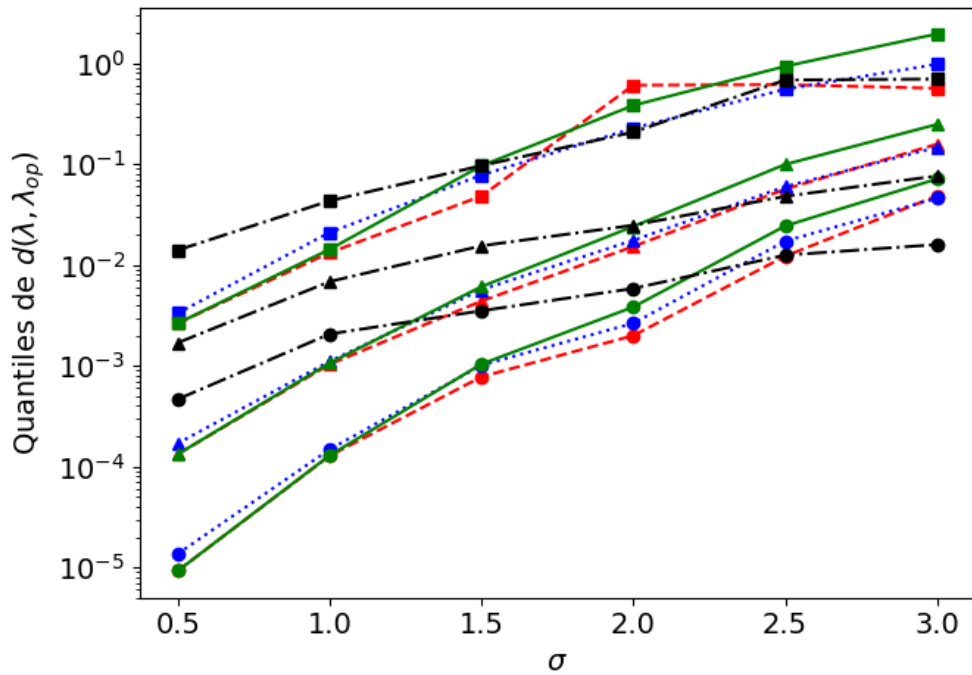
Les caractéristiques statistiques de $d(\lambda_{nous}, \lambda_{op})$ avec les différents paramètres q et celle de $d(\lambda_{cv}, \lambda_{op})$ sont affichées sur la Figure 3.6. Pour un faible niveau de bruit, toutes les méthodes testées fournissent une restauration $u^*(\lambda)$ pratiquement identique à la restauration avec λ optimal.

Avec l'augmentation de la variance du bruit, l'estimation de λ est de plus en plus difficile car la transition entre les régimes de débruitage et de destruction devient moins évidente. La performance de toutes ces méthodes décroît avec l'augmentation de σ^2 . Pour le cas d'un bruit fort (i.e. $\sigma = 3$), ces méthodes fournissent toujours une restauration proche de $u^*(\lambda_{op})$: 50% des cas ont $d(\lambda, \lambda_{op}) < 0,1$.

Ces simulations montrent que les différents paramètres q donnent une performance de restauration similaire, et la méthode proposée a une performance proche de la validation croisée.



(a) Signal constant par morceaux



(b) Signal linéaire par morceaux

FIGURE 3.6 – Les caractéristiques statistiques de $d(\lambda, \lambda_{op})$ sur 500 simulations de deux types de signaux périodiques avec des bruits gaussiens de différentes variances. Nous avons testé notre approche avec $\log_{10}(q) \in \{0,5;1\}$ et le choix automatique de q . **Ligne continue : choix automatique de q ; ligne de points : $\log_{10}(q) = 0,5$; ligne pointillée : $\log_{10}(q) = 1$; ligne tiret-point : 10-fold cross-validation.** Marqueur rond : 25% quantile ; Marqueur triangle : médiane ; Marqueur carré : 95% quantile.

Comparaison avec les méthodes existantes

Nous avons comparé notre proposition avec deux méthodes existantes : Stein Unbiased Risk Estimation (SURE) et Adaptive Universal Threshold (AUT). Les deux méthodes demandent la valeur de la variance du bruit (σ^2). Nous nous intéressons aux cas de la variance connue et de la variance inconnue où elle est estimée par la méthode *median absolute deviation* (MAD) [14]. Nous appliquons cet estimateur avec les ondelettes de Haar, formellement :

$$\hat{\sigma} = \text{MAD}(z) = \frac{1}{c} \text{médiane}(D(z) - \text{médiane}(D(z))) \quad (3.21)$$

avec $D(z) = \sqrt{2}(z_2 - z_1, \dots, z_n - z_{n-1})$ et $c = 0.675$ une constante de calibration proposée par [14].

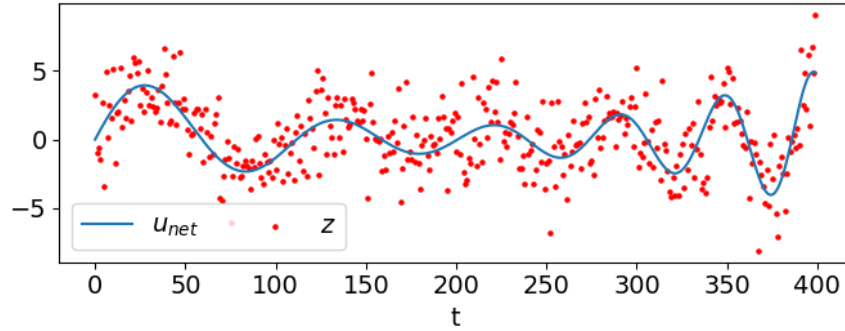


FIGURE 3.7 – Exemple d'un signal apériodique simulé. $z = u_{net} + \epsilon$ avec $\epsilon \sim \mathcal{N}(0, 4)$. Couleurs : [signal original](#) u_{net} et [échantillons simulés](#) z .

D'abord, nous avons testé ces méthodes sur un signal non-périodique (c.f. Figure 3.7) avec les différents types de bruit. Nous avons effectué 500 simulations pour chaque type de bruit, et les valeurs moyennes de $\mathcal{R}(\lambda)$ des différentes méthodes sont résumées dans le Tableau 3.2. Les résultats nous indiquent que notre approche avec les différents choix de q a une performance similaire à SURE pour les différents types de bruit testés, et surpasse AUT dans le cas des bruits élevés.

	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 2^2)$	$\mathcal{N}(0, 3^2)$	$\mathcal{N}(0, 2^2) + \mathcal{U}[-2, 2]$
$\min \mathcal{R}(\lambda)$	16,16	41,87	73,04	51,59
σ connu				
AUT	16,43	58,87	160,28	88,31
SURE	17,07	45,31	81,67	56,39
σ inconnu, $\hat{\sigma}$ estimé par [14]				
AUT	16,52	60,17	160,11	90,51
SURE	17,22	46,44	85,02	57,82
Nos méthodes				
$\log_{10}(q) = 0,5$	16,64	44,60	82,44	55,38
$\log_{10}(q) = 0,75$	16,58	44,04	80,26	54,41
$\log_{10}(q) = 1$	16,58	44,30	83,95	55,67
q automatique	16,58	45,02	87,64	56,77

TABEAU 3.2 – La valeur moyenne de $\mathcal{R}(\lambda) * 100$ sur 500 simulations d'un signal apériodique illustré sur la Figure 3.7. Ces simulations sont réalisées sous les différents types de bruit (gaussien et uniforme).

Ensuite, nous avons testé la restauration du signal block (c.f. Figure 3.2a) avec différents nombres d'échantillons (ce qui est équivalent aux différentes fréquences d'échantillonnage). Les valeurs moyennes de $\mathcal{R}(\lambda)$ avec $n = 199, 499$ et 999 sur 500 simulations sont affichées dans le Tableau 3.3. La performance

de toutes les méthodes testées s'améliore avec l'augmentation du nombre d'échantillons. AUT et SURE ont une erreur plus faible que les nôtres, mais la différence reste marginale (de l'ordre de 0,01). En effet, notre approche basée sur le nombre d'extremums est sensible à la fréquence d'échantillonnage. Moyenner sur un grand nombre de points permet d'avoir une meilleure restauration des extremums intrinsèques du signal original.

	199	499	999
$\min \mathcal{R}(\lambda)$	23,30	11,49	6,42
σ connu			
AUT	25,01	11,92	6,56
SURE	24,97	12,24	6,83
σ inconnu, $\hat{\sigma}$ estimé par [14]			
AUT	26,34	12,08	6,59
SURE	25,26	12,42	6,84
Nos méthodes			
$\log_{10}(q) = 0,5$	27,88	13,36	7,75
$\log_{10}(q) = 0,75$	28,55	13,57	7,39
$\log_{10}(q) = 1$	30,17	14,63	7,72
q automatique	30,54	13,52	7,51

TABEAU 3.3 – La valeur moyenne de $\mathcal{R}(\lambda) * 100$ sur 500 simulations du signal block (SNR=16,91dB) avec les différents nombres d'échantillons. Un exemple du signal original et bruité sont illustrés sur la Figure 3.2a.

Concernant la complexité pour estimer un choix de λ en appliquant un algorithme d'estimation en $O(n \log n)$ (comme par exemple Algorithme 2), la méthode SURE estime λ en $O(n \log n + Nn)$ avec N le nombre de candidats testés et $O(Nn)$ la complexité pour estimer $u^*(\lambda)$ à partir de Λ° pour tous les candidats, tandis que AUT et notre méthode sont en $O(n \log n)$. Avec un grand nombre de candidats pour garantir la précision du choix de λ , SURE est plus lent que AUT et nos méthodes.

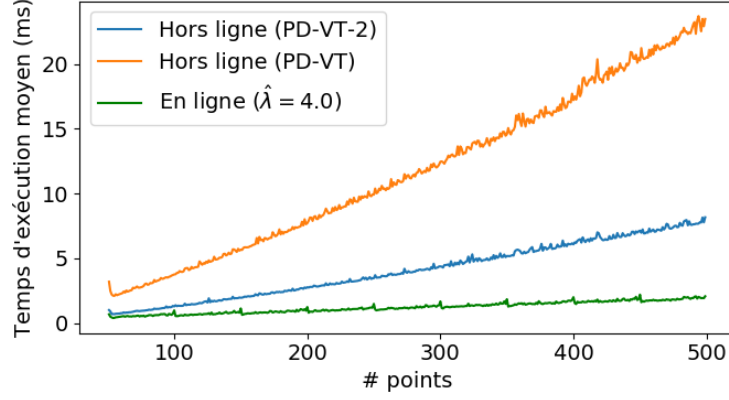
En résumé, notre approche a une performance similaire à la méthode SURE, et elle est adaptée pour la restauration sous les différents types de bruit avec une grande fréquence d'échantillonnage.

Temps d'exécution

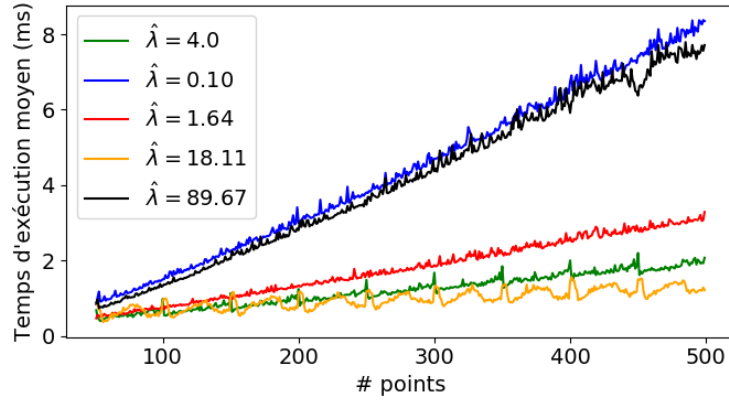
Dans cette section, nous comparons les temps d'exécution des différents algorithmes proposés. Nous mesurons le temps d'exécution pour estimer la liste Λ pour l'Algorithme 1 (hors ligne, $O(n^2)$), et la liste Λ° pour l'Algorithme 2 (hors ligne, $O(n \log n)$) et l'Algorithme 3 (en ligne, $O(n)$). Pour l'implémentation en ligne, l'estimation avec $n+1$ points est basée sur les résultats avec n points, tandis que les algorithmes hors ligne ré-estiment à nouveau les résultats. Les algorithmes sont implémentés en Python, et nous les avons exécutés sur un processeur de i7-7600U. Tous les résultats montrés sont la moyenne de 20 simulations : de 50 à 500 échantillons d'un signal périodique linéaire par morceaux dont une période correspond à la Figure 3.5b.

La performance de l'implémentation en ligne dépend du choix du point de découpe $\hat{\lambda}$. Nous allons, d'abord, fixer $\hat{\lambda} = 4$, et ensuite analyser le choix de $\hat{\lambda}$. Le temps d'exécution moyen est affiché sur la Figure 3.8a. L'implémentation en ligne est plus rapide que les implémentations hors ligne : l'Algorithme 3 prend moins de 2ms pour le 500^e échantillon, tandis que les Algorithmes 1 et 2 prennent respectivement plus de 20ms et 5ms pour une séquence de 500 points.

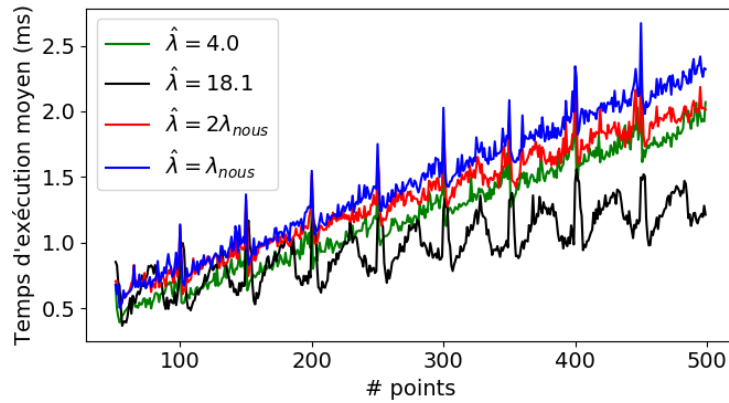
Nous comparons des choix du point de découpe $\hat{\lambda}$, et les résultats sont présentés sur la Figure 3.8b. Pour une valeur trop petite ou trop grande de $\hat{\lambda}$ (i.e. $\hat{\lambda} = 0,1$ et $89,67$), le temps d'exécution est identique à l'implémentation hors ligne car la plupart des éléments sont inclus dans la liste Λ^b pour le premier cas et dans la liste Λ^a pour le deuxième cas. Ces deux listes doivent être recalculées, ce qui rend l'implémentation en ligne inefficace. Les paramètres $\hat{\lambda} \in \{1,64; 4; 18,44\}$ qui nous fournissent un faible temps d'exécution nous semblent un choix pertinent du point de découpe : tous les trois prennent moins de 3ms pour le 500^e échantillon.



(a) Comparaison entre les méthodes en ligne et hors ligne (PD-VT en $O(n^2)$ et PD-VT-2 en $O(n \log n)$).



(b) Implémentation en ligne avec différents points de découpe $\hat{\lambda}$



(c) Propositions de point de découpe $\hat{\lambda}$. Pour le i^e point, $\hat{\lambda}_i = \lambda_{nous}^{i-1}$ avec λ_{nous}^{i-1} notre proposition d'hyper-paramètre pour les $i - 1$ premiers points avec q automatique.

FIGURE 3.8 – Temps d'exécution moyen des implémentations hors ligne et en ligne avec différents points de découpe $\hat{\lambda}$. Le temps d'exécution est moyenné sur 20 simulations.

Cependant, le choix de $\hat{\lambda}$ est loin d'être évident. Nous proposons de choisir $\hat{\lambda} = \lambda_{nous}$, la valeur proposée par notre méthode ou légèrement plus grande : pour le i^e nouveau point, nous prenons $\hat{\lambda}_i = \lambda_{nous}^{i-1}$, la valeur estimée à partir de $\Lambda^{c,i-1}$ (le résultat avec $i - 1$ points). Cela est un choix naturel car nous devons calculer λ_{nous}^{i-1} pour restaurer le signal avec $i - 1$ point. Les résultats sont affichés sur

la Figure 3.8c. Les résultats avec $\hat{\lambda} = \lambda_{\text{nous}}$ et $2\lambda_{\text{nous}}$, y compris le temps pour estimer λ_{nous} , ont un temps d'exécution similaire à celui avec $\hat{\lambda} = 4$ dans nos simulations. Nos propositions ne sont pas optimales, mais nous fournissent une bonne performance. Il faut remarquer que le temps d'exécution est plus important pour tous les points dans une position multiple de cinquante parce que la nouvelle période du signal original commence, et la séquence non-isolée est beaucoup plus longue. Le temps d'exécution de l'implémentation en ligne dépend de la forme du signal.

3.4 Conclusion

Dans ce chapitre, nous avons présenté une méthode de restauration fondée sur la régularisation par la variation totale. Nous avons analysé le comportement du signal restauré en fonction de l'hyper-paramètre λ , de l'introduction et de la suppression d'un échantillon aux deux extrémités de la séquence des échantillons. Nous avons proposé les différents algorithmes pour la VT-restauration d'un signal 1D en temps réel :

- Inspirés par [40] et [32], nous avons proposé des algorithmes efficaces pour estimer le signal restauré u_{VT}^* . Notre proposition a combiné les points forts des méthodes existantes : efficace pour à la fois la sélection de l'hyper-paramètre et la restauration en ligne des flux de données.
- Nous avons proposé une méthode rapide pour estimer la valeur de λ basée sur la variation des nombres d'extremums du signal restauré en fonction de λ . Nous avons comparé notre proposition avec des méthodes existantes (validation croisée, AUT et SURE) sous les hypothèses de bruit gaussien et/ou uniforme. Les simulations montrent que notre méthode a une performance similaire aux méthodes validation-croisée et SURE, aucune des deux méthodes n'étant adaptée au traitement en temps réel. L'estimation de l'hyper-paramètre reste une question ouverte pour les autres modèles de bruit.

La complexité pour estimer le signal restauré avec un choix automatique de λ est en $O(n \log n)$ avec l'implémentation hors ligne et en $O(n)$ avec une implémentation en ligne.

Preuves

Nous donnons, d'abord, quelques lemmes utilisés dans nos preuves. Ces lemmes sont basés sur une valeur fixe de λ , donc λ est omis dans la plupart des notations.

Lemme 3.1. *Pour le k^e palier ($\mathcal{N}_k^*$) avec $1 < k \leq K$, soient $\mathcal{T}_{m,k} = \sum_{j=i_1^k}^{i_m^k} \tau_i$ et $\bar{z}_{m,k} = \frac{1}{\mathcal{T}_{m,k}} \sum_{j=i_1^k}^{i_m^k} \tau_j z_j$, nous avons les inégalités suivantes :*

- $\bar{z}_{m,k} \geq v_k^*$ pour $m = 1, \dots, n_k^*$ si $v_{k-1}^* < v_k^*$.
- $\bar{z}_{m,k} \leq v_k^*$ pour $m = 1, \dots, n_K^*$ si $v_{k-1}^* > v_k^*$.

□

Lemme 3.2. *Pour le k^e palier ($\mathcal{N}_k^*$) avec $1 \leq k < K$, soient $\mathcal{T}_{-m,k} = \sum_{j=i_{n_k^*-m}^k}^{i_{n_k}^k} \tau_i$ et $\bar{z}_{-m,k} = \frac{1}{\mathcal{T}_{-m,k}} \sum_{j=i_{n_k^*-m}^k}^{i_{n_k}^k} \tau_j z_j$, nous avons :*

- $\bar{z}_{-m,k} \geq v_k^*$ pour $m = 1, \dots, n_k$ si $v_{k+1}^* < v_k^*$.
- $\bar{z}_{-m,k} \leq v_k^*$ pour $m = 1, \dots, n_K$ si $v_{k+1}^* > v_k^*$.

□

Lemme 3.3. — Si $u_m^* = \hat{u}_m$, $\exists m \leq n$, alors $u_i^* = \hat{u}_i$, $\forall i \leq m$.

- Si $\text{signe}(v_{j-1}^* - v_j^*) = \text{signe}(v_j^* - \hat{v}_j)$, alors $\hat{v}_i = v_i^*$, $\forall i \leq j - 1$.

□

Lemme 3.4. *Dans le cas où $v_{K^*-1}^* < v_{K^*}^*$:*

- Pour $z_{n+1} \geq v_{K^*}^*$, on a $v_j^* = \hat{v}_j$ et $\mathcal{N}_j^* = \hat{\mathcal{N}}_j$, $\forall j < K^*$.
- Pour $z_{n+1} < v_{K^*}^*$, s'il existe un index l tel que $v_l^* > v_{l+1}^*$, alors $v_j^* = \hat{v}_j$ et $\mathcal{N}_j^* = \hat{\mathcal{N}}_j$, $\forall j \leq l$

□

Démonstration de Théorème 3.2. Nous pouvons facilement montrer qu'au moins l'un des paliers fusionnés est un maximum ou minimum local, car les paliers neutres ne varient pas suivant λ .

Nous supposons que $k + 1$ paliers ont été fusionnés dans un nouveau palier pour un $\lambda_{l-k} \in \Lambda$ donné. Soit le nouveau palier après la fusion $\mathcal{N}_j(\lambda_{l-k}) = \{\mathcal{N}_j(\lambda_l), \dots, \mathcal{N}_{j+k}(\lambda_l)\}$, nous allons discuter des cas suivants :

- Pour $j > 1$ et $j + k < K^*(\lambda_l)$, trois cas sont possibles :
 - $v_{j-1}^*(\lambda_l) > v_j^*(\lambda_l)$ et $v_{j+k}^*(\lambda_l) < v_{j+k+1}^*(\lambda_l)$: le nouveau palier est minimal local. On peut facilement montrer qu'il existe au moins un palier min parmi les paliers à fusionner $\{\mathcal{N}_j(\lambda_l), \dots, \mathcal{N}_{j+k}(\lambda_l)\}$ et qu'il existe un palier min de plus que les paliers max. Donc, soit m le nombre de paliers min parmi $\{\mathcal{N}_j(\lambda_l), \dots, \mathcal{N}_{j+k}(\lambda_l)\}$, nous avons $g(\lambda_{l-k}) = g(\lambda_l) - 2(m - 1)$.
 - $v_{j-1}^*(\lambda_l) < v_j^*(\lambda_l)$ et $v_{j+k}^*(\lambda_l) > v_{j+k+1}^*(\lambda_l)$: le nouveau palier est maximal local. On peut facilement montrer qu'il existe au moins un palier max parmi les paliers à fusionner $\{\mathcal{N}_j(\lambda_l), \dots, \mathcal{N}_{j+k}(\lambda_l)\}$ et qu'il existe un palier max de plus que les paliers mins. Donc, soit m le nombre de paliers max parmi $\{\mathcal{N}_j(\lambda_l), \dots, \mathcal{N}_{j+k}(\lambda_l)\}$, nous avons $g(\lambda_{l-k}) = g(\lambda_l) - 2(m - 1)$.
 - $v_{j-1}^*(\lambda_l) > v_j^*(\lambda_l)$ et $v_{j+k}^*(\lambda_l) > v_{j+k+1}^*(\lambda_l)$ (ou $v_{j-1}^*(\lambda_l) < v_j^*(\lambda_l)$ et $v_{j+k}^*(\lambda_l) < v_{j+k+1}^*(\lambda_l)$) : le nouveau palier est un palier neutre. Soit m le nombre de paliers max parmi $\{\mathcal{N}_j(\lambda_l), \dots, \mathcal{N}_{j+k}(\lambda_l)\}$, nous avons $g(\lambda_{l-k}) = g(\lambda_l) - 2m$.
- Dans le cas où $j = 1$, deux cas sont possibles :
 - $v_{k+1}^*(\lambda_l) < v_{k+2}^*(\lambda_l)$: le nouveau palier est minimal local. Il doit exister au moins un palier min parmi $\{\mathcal{N}_1(\lambda_l), \dots, \mathcal{N}_{k+1}(\lambda_l)\}$. De plus, il existe un palier min de plus ou un nombre égal des paliers mins que les paliers max. Donc, soient m_1 le nombre de paliers mins et m_2 le nombre de paliers max parmi $\{\mathcal{N}_1(\lambda_l), \dots, \mathcal{N}_{k+1}(\lambda_l)\}$, nous avons $g(\lambda_{l-k}) = g(\lambda_l) - m_1 - m_2 + 1$.
 - $v_{k+1}^*(\lambda_l) > v_{k+2}^*(\lambda_l)$: le nouveau palier est maximal local. Il doit exister au moins un palier max parmi $\{\mathcal{N}_1(\lambda_l), \dots, \mathcal{N}_{k+1}(\lambda_l)\}$. De plus, il existe un palier max de plus ou un nombre égal des paliers max que les paliers mins. Donc, soient m_1 le nombre de paliers mins et m_2 le nombre de paliers max parmi $\{\mathcal{N}_1(\lambda_l), \dots, \mathcal{N}_{k+1}(\lambda_l)\}$, nous avons $g(\lambda_{l-k}) = g(\lambda_l) - m_1 - m_2 + 1$.
- Pour $j + k = K^*(\lambda_l)$: le nouveau palier devient le dernier. La variation de $g(\lambda)$ est identique au point précédent.

En résumé, la fonction $g(\lambda)$ est constante par morceaux et décroissante.

□

Démonstration de Théorème 3.3. C'est une conséquence directe du Lemme 3.4.

□

Démonstration de Proposition 3.2. Soit l tel que $\lambda_{l+1} \leq \hat{\lambda} < \lambda_l$. Pour $\lambda \leq \hat{\lambda}$, la fusion de éléments dans le j^e palier ($\mathcal{N}_j^*(\hat{\lambda})$) est indépendante des points en dehors de ce palier sous la même condition de bord : $\text{signe}(u_{i_1^j} - u_{i_{n_j-1}^j})$ et $\text{signe}(u_{i_1^{j+1}} - u_{i_{n_j}^j})$ restent identiques. Les conditions de bord sont garanties par l'introduction des points dans les jonctions de paliers, car ces points vont fusionner avec ce palier pour $\lambda = \hat{\lambda} + \epsilon_\lambda$. Donc, nous avons $\lambda_{i_1^j, \dots, i_{n_j-1}^j}^{\circ, n} = \lambda_{i_1^j, \dots, i_{n_j-1}^j}^{*, j}$.

Pour finir les preuves, $\cup \Lambda^{*, j} = \cup \lambda_{i_1^j, \dots, i_{n_j}^j}^{*, j} = \cup \lambda_{i_1^j, \dots, i_{n_j-1}^j}^{\circ, n} = \{\lambda \leq \hat{\lambda}, \forall \lambda \in \Lambda^\circ\}$.

□

Démonstration de Lemme 3.1. Nous pouvons montrer que $\sum_{j=i_1^k}^{i_m^k} \tau_j(z_j - \bar{z}_{m,k})^2 \leq \sum_{j=i_1^k}^{i_m^k} \tau_j(z_j - v_k^*)^2$.

Dans le cas où $\bar{z}_{m,k} \in [\min(v_{k-1}^*, v_k^*), \max(v_{k-1}^*, v_k^*)]$, $\sum_{j=i_1^k}^{i_m^k} \tau_j(z_j - \bar{z}_{m,k})^2 + \sum_{j=i_{m+1}^k}^{i_{n_k}^k} \tau_j(z_j - v_k^*)^2 + \lambda(v_k^* - v_{k-1}^*) < \sum_{j \in \mathcal{N}_k} \tau_j(z_j - v_k^*)^2 + \lambda(v_k^* - v_{k-1}^*)$.

Comme v_k^* est donné par la minimisation de F_n , $\sum_{j=i_1^k}^{i_m^k} \tau_j(z_j - \bar{z}_{m,k})^2 + \sum_{j=i_{m+1}^k}^{i_{n_k}^k} \tau_j(z_j - v_k^*)^2 + \lambda(v_k^* - v_{k-1}^*) \geq \sum_{j \in \mathcal{N}_k} \tau_j(z_j - v_k^*)^2 + \lambda(v_k^* - v_{k-1}^*)$. Donc $\bar{z}_{m,k} \notin [\min(v_{k-1}^*, v_k^*), \max(v_{k-1}^*, v_k^*)]$. $\square$

Démonstration de Lemme 3.2. Identique au Lemme 3.1. $\square$

Démonstration de Lemme 3.3. $(u_1, \dots, u_m) = \arg \min F_m$ pour $m < n$ sous la condition $u_m = u_m^*$. Si $u_m^* = \hat{u}_m$, les deux solutions $(u_1^*, \dots, u_m^*)$ et $(\hat{u}_1, \dots, \hat{u}_m)$ minimisent la même fonctionnelle F_m sous la même condition $u_m = u_m^*$. Donc $(u_1^*, \dots, u_m^*) = (\hat{u}_1, \dots, \hat{u}_m) = (u_1, \dots, u_m)$.

Supposons que les j premiers paliers contiennent m points, $(v_1^*, \dots, v_{j-1}^*) = \arg \min F_m$ sous la condition de bord $u_m^* < u_{m+1}^*$. Si $\text{signe}(v_{j-1}^* - v_j^*) = \text{signe}(v_j^* - \hat{v}_j)$, la condition de bord reste identique. Donc $\hat{v}_i = v_i^*, \forall i \leq j-1$. $\square$

Démonstration de Lemme 3.4. Le premier point est trivial suivant le Lemme 3.3.

Pour la simplicité de présentation, nous notons l'ensemble du dernier palier et le nouvel échantillon $\mathcal{N}_a = \{\mathcal{N}_{K^*}^*, n+1\}$. Pour $z_{n+1} < v^*(k)$, nous allons discuter des cas suivants :

- Influence au dernier palier $\mathcal{N}_{K^*}^*$: on a $\hat{u}_i < u_i^*, \forall i \in \mathcal{N}_{K^*}^*$ suivant le Lemme 3.1.
- Influence au avant-dernier palier $\mathcal{N}_{K^*-1}^*$ avec $v_{K^*-1}^* < v_{K^*}^*$: si $\hat{v}_{K^*} > v_{K^*}^*$, alors $\hat{v}_{K^*-1} = v_{K^*-1}^*$ suivant le Lemme 3.3. Dans le cas contraire, $\{\mathcal{N}_{K^*-1}^*, \mathcal{N}_a\}$ va soit fusionner en un seul palier, soit découper en plusieurs, et $\hat{v}_{K^*-1} < v_{K^*-1}^*$ suivant le Lemme 3.1. Autrement dit, si on a une séquence $v_j^* < \dots < v_{K^*}^*$ avec $v_{i+1}^* - v_i^*$ suffisamment petit pour tout $i \geq j$, alors l'introduction de z_{n+1} peut changer la valeur des élément du j^e palier $\mathcal{N}_j^*$.
- Influence au palier $\mathcal{N}_i^*$ avec $v_{i+1}^* < \dots < v_{K^*-1}^* < v_{K^*}^*$ et $v_i^* > v_{i+1}^*$: suivant les analyses précédentes, on a $\hat{v}_{i+1} \leq v_{i+1}^*$. Suivant le Lemme 3.3, $\hat{v}_j = v_j^*$ et $\hat{\mathcal{N}}_j = \mathcal{N}_j^*$ pour tout $j \leq i$. $\square$

Chapitre 4

Applications industrielles de la restauration automatique du signal

Sommaire

4.1 Applications pour un signal localement lisse	44
4.1.1 Présentation du signal collecté	44
4.1.2 Présentation des données industrielles	45
4.1.3 Restauration du signal	45
4.1.4 Détection des ruptures du signal	49
4.1.5 Introduction des informations reconstituées dans la modélisation de la qualité du produit	51
4.1.6 Analyse de la qualité du produit à l'aide des informations reconstituées	52
4.1.7 Classification de la performance d'un poste de production	56
4.1.8 Liens entre la performance d'un poste et l'écart-type des résidus	57
4.1.9 Conclusion	59
4.2 Applications autour d'une mesure de qualité en 2D	60
4.2.1 Présentation du signal collecté	60
4.2.2 Présentation des données	61
4.2.3 Critères des bons et mauvais postes	61
4.2.4 Analyse des inhomogénéités transversales	61
4.2.5 Conclusion	64
4.3 Applications autour d'une mesure de qualité en temps réel	65
4.3.1 Présentation des données	65
4.3.2 Méthode proposée : détection en ligne des ruptures	66
4.3.3 Sélection des paramètres	67
4.3.4 Méthode de détection robuste	68
4.3.5 Résultats de détection	68
4.3.6 Travaux en cours	70
4.3.7 Conclusion	70
4.4 Application : surveillance de la dérive de la variance du bruit	71
4.4.1 Méthode d'estimation de la variance du bruit	71
4.4.2 Analyse de la performance	72
4.4.3 Conclusion	75

Les méthodes automatiques de restauration proposées ont une faible complexité temporelle et une bonne performance. Elles sont particulièrement adaptées au traitement d'une grande quantité de données en temps réel. D'ailleurs, les signaux collectés sur les lignes de production ont souvent des changements brusques dus à la modification des réglages de machine, et la restauration localement lisse proposée par la VT-restauration nous permet de localiser précisément les instants de changements tout en proposant une bonne restauration pour les parties lisses. Nous avons appliqué la méthode proposée

dans cette thèse sur différents signaux issus des procédés de Saint-Gobain pour visualiser les variations intrinsèques du signal et détecter les changements brusques. La visualisation et la détection peuvent s'effectuer d'une manière hors ligne ou en ligne :

- Hors ligne : il s'agit d'analyses effectuées sur une séquence d'échantillons fixe. L'objectif est de comprendre les comportements du signal restauré (comme par exemple les changements brusques ou les variations périodiques), et de les lier aux réglages et au fonctionnement du procédé.
- En ligne : il s'agit d'applications sur les flux de données, dans lesquels les nouveaux échantillons sont collectés au cours du temps. Les signaux collectés sont souvent non-stationnaires : certaines caractéristiques statistiques (par exemple : l'espérance du signal, la variance du bruit) changent au cours du temps. L'objectif des applications en ligne est de construire des indicateurs liés aux états de fonctionnement du procédé basés sur le comportement des signaux, et de trouver rapidement les indices des changements d'état (comme par exemple le dysfonctionnement de capteurs).

Dans ce chapitre, nous allons présenter les analyses et les applications sur des signaux venant de différents procédés de Saint-Gobain ou des signaux simulés, basées sur la méthode de restauration proposée au Chapitre 3. Nous allons d'abord présenter des analyses autour d'un signal localement lisse dans la Section 4.1, puis des analyses réalisées sur une mesure de qualité en 2D sont présentées dans la Section 4.2. Deux applications en ligne sur des signaux non-stationnaires sont en cours de développement : la détection en ligne des changements brusques d'un signal réel est présentée dans la Section 4.3, et la détection en ligne de la dérive de la variance du bruit est présentée dans la Section 4.4.

4.1 Applications pour un signal localement lisse

Nous avons appliqué la méthode de restauration proposée pour analyser un signal enregistré dans le *Process 1*. Les travaux des analyses statistiques de *Process 1* sont présentés dans la Partie I. L'objectif est de détecter les changements brusques du signal puis d'essayer de relier le signal débruité du paramètre concerné à la qualité finale du produit.

4.1.1 Présentation du signal collecté

Le paramètre procédé, noté t_c , est mesuré par un capteur pour chaque produit réalisé. Pour un poste de fabrication, nous collectons un vecteur d'échantillons $\mathbf{t}_c = \{t_{c,1}, \dots, t_{c,n}\}$ avec $t_{c,i}$ la mesure pour le i^e produit.

Ce paramètre est l'un des paramètres suivis régulièrement par l'usine pour caractériser le fonctionnement d'un poste de fabrication. Étant identifié par les méthodes de sélection de variables pour la modélisation statistique de la qualité de produit dans la Partie I, t_c est un paramètre important selon notre interlocuteur expert métier pour les raisons suivantes :

- t_c caractérise la synchronisation entre deux étapes cruciales de fabrication.
- L'algorithme de contrôle du procédé propose plusieurs réglages basés sur la valeur de t_c , et ces réglages ont une influence directe sur la qualité des produits finaux.

La valeur du signal t_c est influencée par de multiples facteurs, et le couplage de différents phénomènes physiques implique d'importantes variations sur les signaux mesurés. Un des facteurs influençant directement le signal de t_c est l'état d'un des composants de la machine, que nous appellerons C : le versement des matières premières abîme l'état du composant C , ce qui entraîne une augmentation de la valeur de t_c au cours du temps. D'ailleurs, un opérateur fait régulièrement des maintenances afin de maintenir le bon état du composant C . Ces actions de maintenance provoquent une descente brusque du signal mesuré, et elles ont un impact important sur le fonctionnement de machine. Cependant, l'information concernant les actions de maintenance n'est pas correctement archivée dans la base des données.

Le rôle important du signal de t_c et la forte variabilité de ce signal en raison de causes extérieures non remontées dans les données nous motivent à appliquer les méthodes de restauration proposées dans le Chapitre 3 afin de permettre une analyse plus approfondie.

4.1.2 Présentation des données industrielles

Nous avons analysé plusieurs types de produits réalisés par deux usines différentes. Les ensembles de données analysées sont les suivants :

- Ensemble F1 : un premier type de produit réalisé par une ligne de production pendant une année.
- Ensemble P1 : un deuxième type de produit réalisé par plusieurs lignes de production pendant une année.
- Ensemble P2 : un troisième type de produit réalisé par plusieurs lignes de production pendant une année.

4.1.3 Restauration du signal

L'étape de débruitage est indispensable pour comprendre la relation entre le signal t_c et la qualité des produits finaux. L'objectif du débruitage est de garder les grandes variations intrinsèques du signal en éliminant les oscillations locales introduites par le bruit de mesure.

Pré-traitements

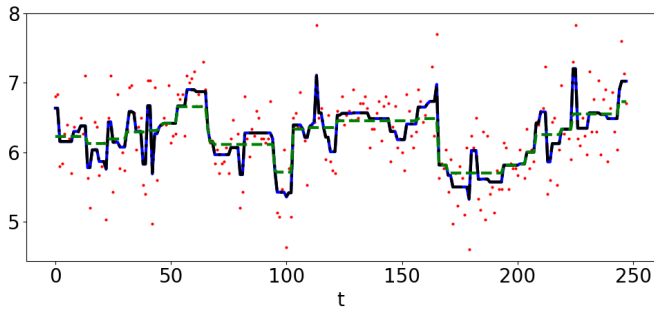
Le signal possède parfois des valeurs aberrantes à cause de problèmes de versement des matières premières. Nous considérons un point i comme une mesure aberrante si l'approximation du gradient du second ordre $|\nabla t_{c,i} - \nabla t_{c,i+1}| > \eta$ avec $\nabla t_{c,i} = t_{c,i} - t_{c,i-1}$ et η une valeur de seuillage à fixer. Autrement dit, une mesure aberrante correspond à un pic éloigné de ses voisins. Par la suite, nous avons fixé $\eta = 10/3$, ce qui correspond à une variation entre trois points consécutifs irréalisable en réalité. Les mesures aberrantes et les points ayant un indicateur d'anomalie actif dans les données de suivi du procédé ne sont pas considérés dans les analyses que nous allons maintenant dérouler.

Résultats de restauration

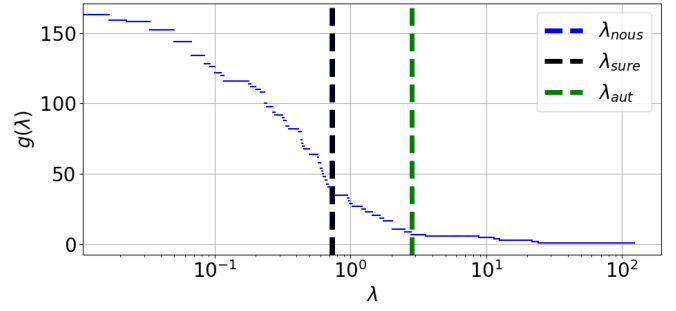
Nous avons d'abord appliqué notre méthode avec l'option automatique (3.18) sur les données de plusieurs postes de fabrication. Nous avons comparé les résultats ainsi obtenus avec ceux issus de restaurations par des méthodes existantes (SURE et AUT) dans lesquelles l'écart-type du bruit (σ) est estimé par l'estimateur MAD (3.21).

Les restaurations de trois postes choisis parmi l'ensemble F1 ainsi que les fonctions $g(\lambda)$ correspondantes sont affichées dans la Figure 4.1. Les trois méthodes proposent une valeur de λ autour du changement de tendance de $g(\lambda)$ en fonction de λ , ce qui valide notre intuition de départ. Nous pouvons constater, dans la plupart des postes, que la méthode SURE propose une petite valeur de λ , tandis que la méthode AUT propose une grande valeur de λ . Les restaurations avec λ_{sure} , contenant souvent des pics, ne sont pas suffisamment régulières pour les futures applications. Quant à la restauration avec λ_{aut} , elle fournit une solution bien régulière, mais certaines variations intrinsèques du signal sont masquées en raison de l'effet de régularisation trop important. Notre méthode propose souvent une valeur de λ entre les propositions de SURE et AUT, et les restaurations avec λ_{nous} nous semblent cohérentes dans la plupart des postes. Cependant, le bruit de mesure de t_c n'est pas identiquement distribué, et certaines parties du signal t_c ont une variance du bruit plus significative, comme par exemple pour les 50 premiers points de la Figure 4.1a. Cela donne une fausse région de transition autour de $\lambda = 0,8$ (c.f. Figure 4.1b), ce qui perturbe l'estimation de l'hyper-paramètre λ . La méthode entièrement automatique basée sur l'approximation de $\partial^4 g(\lambda)$ n'est pas robuste pour détecter le changement des régimes à partir d'une fonction $g(\lambda)$ irrégulière. Après plusieurs essais, notre méthode ajustée par l'option semi-automatique (c.f. Section 3.1.4) avec $\log_{10}(q) = 1$ et $p = 0,03$ (c.f. Figures 4.2) nous paraît mieux conserver la variation de t_c tout en atténuant l'effet du bruit.

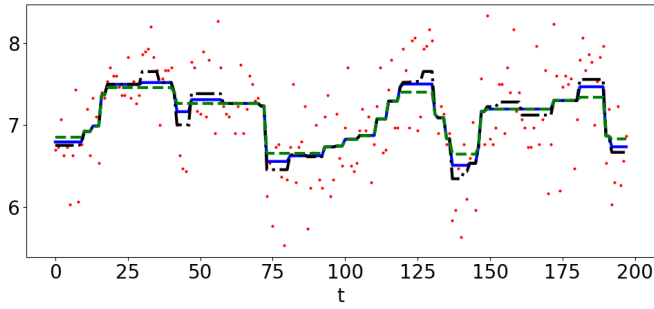
Par la suite, nous effectuons les analyses du signal de t_c en nous basant sur les restaurations de l'option semi-automatique avec $\log_{10}(q) = 1$ et $p = 0,03$.



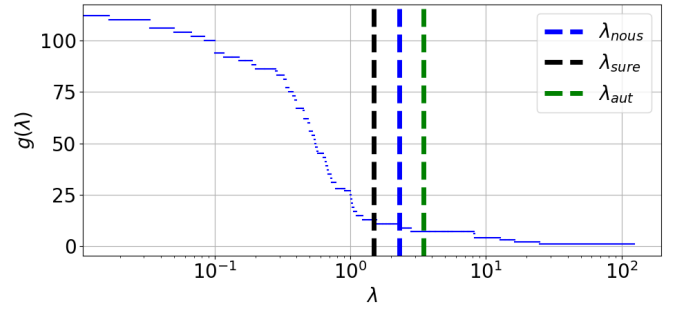
(a) Poste 1



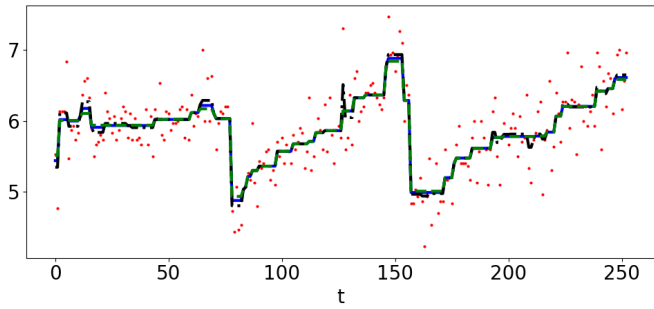
(b) $g(\lambda)$ du poste 1



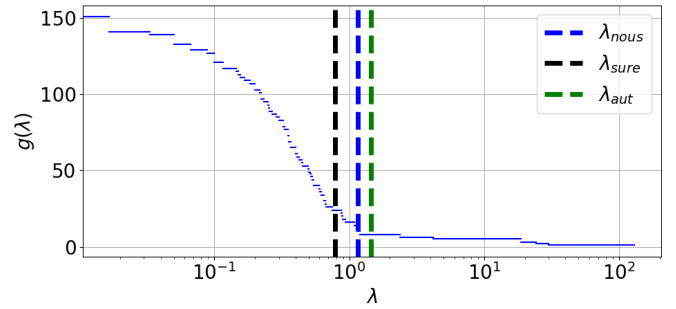
(c) Poste 2



(d) $g(\lambda)$ du poste 2

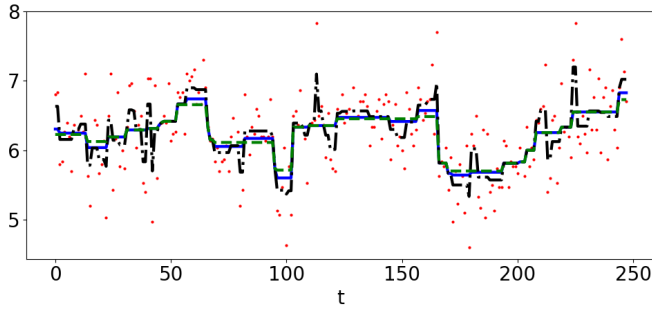


(e) Poste 3

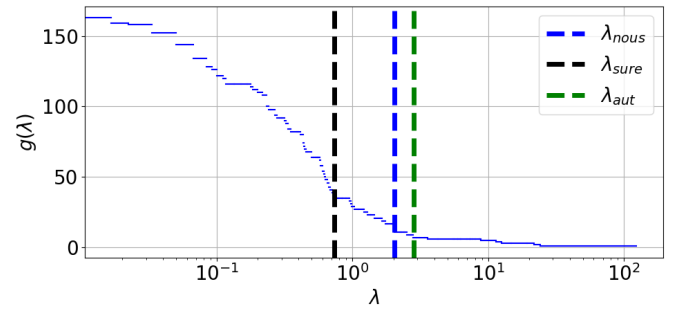


(f) $g(\lambda)$ du poste 3

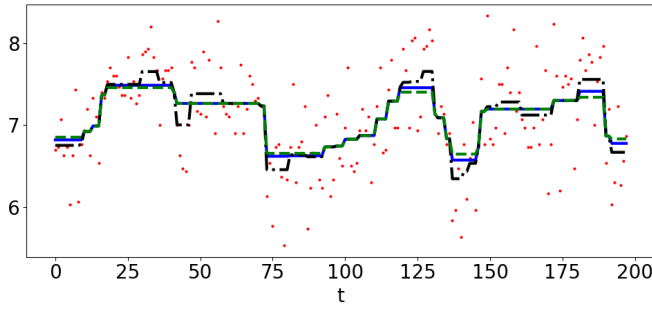
FIGURE 4.1 – Illustration des VT-restaurations avec les valeurs de λ estimées par différentes méthodes. Couleurs : signal brut t_c , SURE, AUT et notre méthode automatique (3.18) avec le choix automatique de q . Données : ensemble F1.



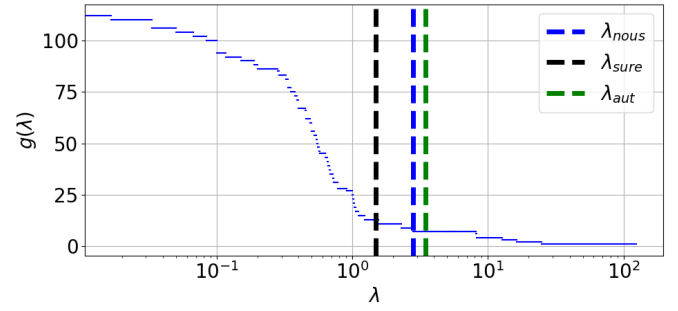
(a) Poste 1



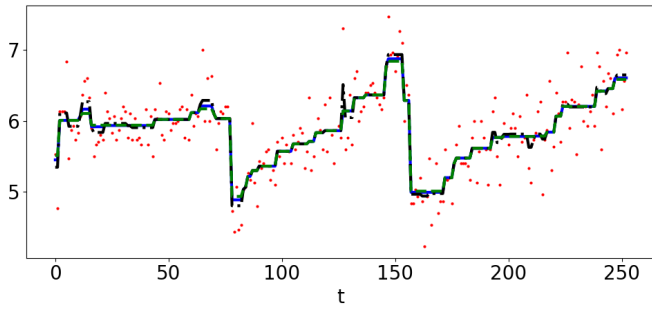
(b) $g(\lambda)$ du poste 1



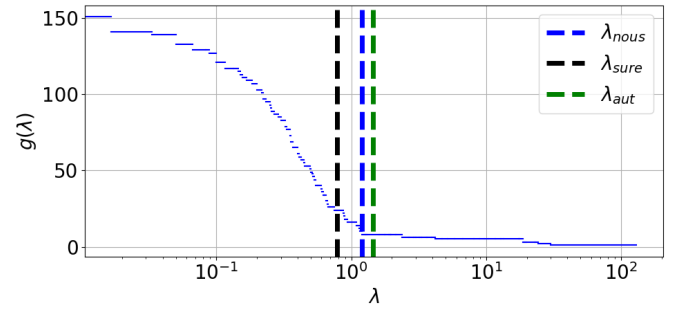
(c) Poste 2



(d) $g(\lambda)$ du poste 2



(e) Poste 3



(f) $g(\lambda)$ du poste 3

FIGURE 4.2 – Illustration des VT-restaurations avec les valeurs de λ estimées par différentes méthodes. Couleurs : **signal brut** t_c , **SURE**, **AUT** et **notre méthode semi-automatique** avec $\log_{10}(q) = 1$ et $p = 0,03$. Données : ensemble F1.

4.1.4 Détection des ruptures du signal

En nous appuyant sur le signal restauré, nous proposons une méthode pour détecter les ruptures (i.e. les descentes brusques) liées aux actions de maintenance.

Les causes possibles des descentes de valeur de t_c sont multiples, comme par exemple les actions de maintenance ou la variation d'un paramètre de versement. Les variations de paramètres de versement sont représentées par une faible variation de t_c , tandis que les actions de maintenance entraînent des descentes significatives du signal, comme illustré sur la Figure 4.3.

Soit $\nabla \mathbf{u} = (u_2^* - u_1^*, \dots, u_n^* - u_{n-1}^*)$ avec $\mathbf{u}^* = (u_1^*, \dots, u_n^*)$ la restauration proposée par notre méthode, la méthode naïve pour détecter les maintenances consiste à retrouver les index des descentes brusques par $\{i | \nabla u_{i-1} < \alpha\}$ avec $\alpha < 0$. Cependant, les différentes expérimentations montrent que les grandes descentes représentant les actions de maintenances sont parfois une suite décroissante avec plusieurs petits paliers. Un simple seuillage risque donc de détecter plusieurs descentes en proximité.

Nous proposons une méthode plus robuste pour détecter les descentes. Nous appelons “séquence décroissante” $(u_k^*, \dots, u_l^*)$ satisfaisant les conditions suivantes :

- $\nabla u_i \leq 0$ pour tout $i = k + 1, \dots, l$.
- Deux conditions de bord :
 1. $k = 1$, ou $\nabla u_k > 0$ si $k > 1$.
 2. $l = n$, ou $\nabla u_{l+1} > 0$ si $k < n$.

Autrement dit, la séquence décroissante ne peut pas être élargie.

Une séquence strictement décroissante est une séquence décroissante avec $\nabla u_i < 0$ pour tous les éléments i .

Nous détectons les descentes du signal restauré avec la méthode suivante : une descente est représentée par une séquence décroissante $k, \dots, l$ avec $u_l^* - u_k^* < \alpha_1$, et le point qui correspond au premier produit réalisé après la maintenance est le dernier point $i \in [k, l]$ satisfaisant la condition $u_j^* - u_i^* < \alpha_2$ avec $\{j, \dots, i\}$ une séquence strictement décroissante. Un exemple est illustré dans la Figure 4.3. Cette méthode propose une seule détection pour chaque descente brusque du signal restauré, ce qui facilite l'interprétation des détections.

Exemple de décroissance due au changement d'un paramètre procédé

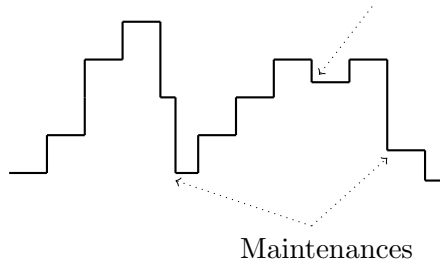


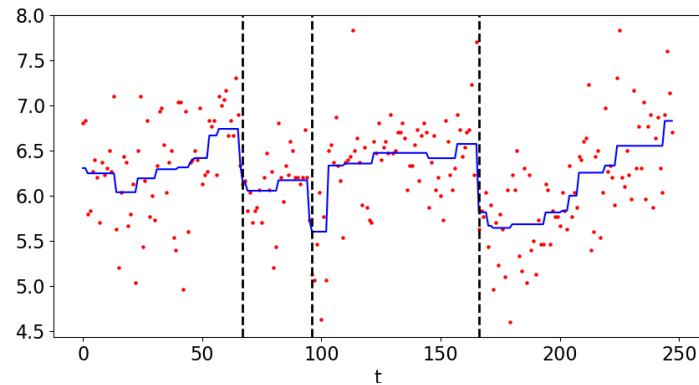
FIGURE 4.3 – Illustration des actions de maintenance détectées à partir d'un signal restauré. Les maintenances correspondent au dernier point des grandes descentes du signal restauré.

Au final, nous avons deux paramètres α_1 et α_2 à fixer. L'intervalle usuel des valeurs de t_c dépend du type de produits réalisés et de la forme de l'outil de versement utilisé. Nous devons avoir des valeurs spécifiques de α_1 et α_2 pour chaque type de produit. Après plusieurs expérimentations, nous avons fixé $\alpha_1 = \frac{5}{12}$ et $\alpha_2 = \alpha_1/1,5$ pour l'ensemble F1.

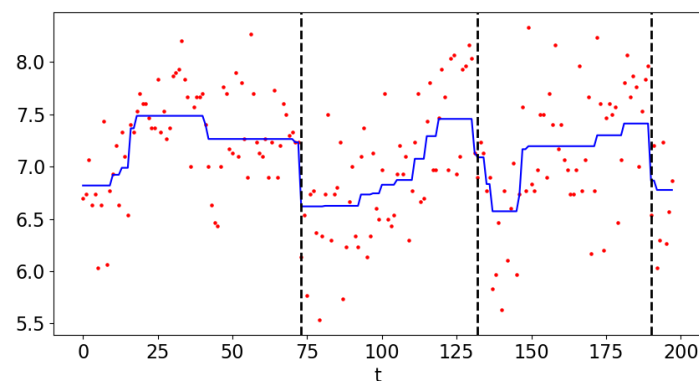
Trois exemples de détection sont illustrés dans la Figure 4.4. Nous pouvons constater que les maintenances détectées correspondent bien à la fin des grandes descentes du signal.

Validation des détections

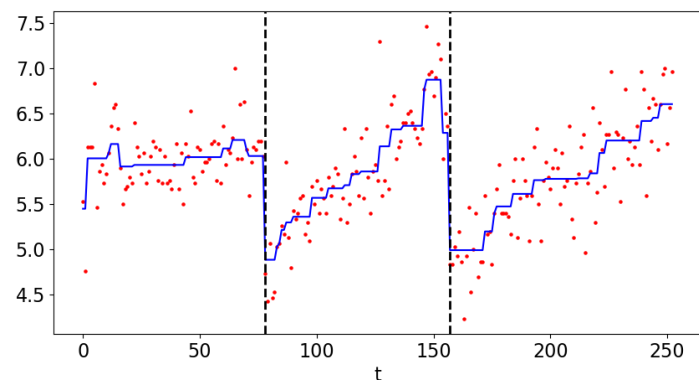
Le manque d'information précise concernant l'occurrence de maintenance ne nous permet pas de valider qualitativement si les détections proposées correspondent réellement aux actions de maintenance.



(a) Poste 1



(b) Poste 2



(c) Poste 3

FIGURE 4.4 – Illustration de maintenances détectées. Couleurs : **maintenances détectées avec $\alpha_1 = 5/12$ et $\alpha_2 = \alpha_1/1,5$** , **signal brut t_c** et **restauration t_c proposée**. Données : ensemble F1.

Les experts procédé nous proposent de valider la détection en suivant la variation d'un réglage machine : les opérateurs modifient souvent manuellement un paramètre après une maintenance effectuée. Cependant, ce réglage est souvent modifié au cours d'un poste pour maintenir le bon fonctionnement de la machine, donc la variation de ce réglage machine n'est pas non plus l'assurance qu'une maintenance ait eu lieu.

Nous avons appliqué notre méthode de détection de maintenance à partir du signal de t_c sur les productions de plusieurs lignes durant l'année 2020. Nous avons comparé les descentes détectées avec la variation du réglage machine : 36,4% des postes analysés ont plus de 75% des détections qui correspondent à une variation significative de ce réglage, tandis que 31,4% des postes ont entre 50% et

75% de détections correspondantes. Contrairement à l'intuition de l'expert métier, le réglage machine ne permet pas de valider les détections de maintenance par notre méthode.

4.1.5 Introduction des informations reconstituées dans la modélisation de la qualité du produit

Dans cette section, nous souhaitons déterminer si les actions de maintenance détectées par notre méthode combinée au signal restauré u^* pourraient améliorer les modèles prédictifs de la qualité de produit (c.f. Partie I).

La mesure de qualité $y \in \mathbb{R}$ est réalisée directement en sortie de machine. Nous nous sommes concentrés sur les produits “normaux” (l'ensemble de produits exclu de ceux réalisés immédiatement après le remplissage des matières premières). Nous avons effectué une sélection de n_v variables par méthodes statistiques, et le meilleur modèle de régression linéaire utilisant ces variables sélectionnées pour prédire y avant toute opération de restauration de signal nous donne un score R^2 de 0,193 pour les données de l'usine F. En introduisant les relations non-linéaires dans la modélisation, le meilleur modèle additif généralisé (GAM) obtenu a un score R^2 de 0,290 pour les données de l'usine F. Ce modèle est un GAM utilisant les produits des splines naturels (ns).

Nous avons appliqué la méthode de restauration au signal brut de t_c , et les variables reconstituées par la méthode de restauration pour chaque produit considéré sont le signal restauré u^* et la durée d_g par rapport à la dernière maintenance réalisée. La durée du produit i par rapport à la dernière maintenance, notée $d_{g,i}$, est donnée par le nombre de produits réalisés entre la dernière maintenance détectée avant le produit i et la réalisation de celui-ci. Par exemple, la durée du premier produit réalisé après une maintenance est de 0, et celle du deuxième après la maintenance est de 1.

Dans un premier temps, nous avons ajouté aux n_v variables initialement sélectionnées les deux nouvelles variables issues de la restauration du signal t_c : u^* (le signal tp restauré) et d_g (le nombre de produits réalisés depuis la dernière maintenance) dans la modélisation linéaire. Le nouveau modèle linéaire avec $n_v + 2$ variables impliquées a un score R^2 de 0,200 pour le test test, soit +0,007 par rapport au modèle avec initialement n_v variables sélectionnées. Les deux nouvelles variables introduites ont une p -valeur inférieure à 5%, ce qui indique que ces deux variables apportent de l'information dans la modélisation linéaire de la cible y .

Dans un second temps, nous avons introduit ces deux nouvelles variables dans le modèle GAM utilisant les splines multi-dimensionnelles des variables sélectionnées. Après plusieurs expérimentations, les variables $(u^* - t_c)$, $(u^* - t_c)^2$ et $ns(d_g, df = 5) * u^*$ apportent une légère amélioration au modèle : le meilleur modèle a un score R^2 de 0,308 pour les données de l'usine F, soit une amélioration de +0,018 par rapport au modèle de même type avec initialement n_v variables sélectionnées.

Les analyses sur les caractéristiques statistiques du signal t_c ont été effectuées en collaborant avec les ingénieurs procédé et les ingénieurs de recherche en charge de ce système. La variation des informations utiles (i.e. l'état du composant C) est masquée dans les mesures bruitées de t_c . Les relations linéaires entre t_c et les variables de l'étape de versement restent très faibles, allant à l'encontre de l'intuition métier. Influencée par de multiples facteurs, tout en étant noyée dans un fort bruit de mesure, la variable t_c se révèle être insuffisamment fiable pour représenter l'état actuel de la machine. Ces observations, suites aux analyses poussées des signaux à disposition dans les données procédé, ont amené les ingénieurs de recherche de Saint-Gobain à adopter une autre méthodologie et à mettre en place de nouveaux capteurs qui sont en cours d'implémentation afin d'obtenir d'autres mesures capables de mieux cerner l'état de la machine.

En résumé, l'introduction des informations reconstituées (i.e. de u^* et d_g pour chaque signal restauré), en l'état actuel des données, n'apporte pas une amélioration sur les modèles statistiques de prédiction de la qualité finale. Elles ont cependant motivé l'implémentation d'autres capteurs afin d'améliorer le suivi de production.

4.1.6 Analyse de la qualité du produit à l'aide des informations reconstituées

Une mesure de qualité est réalisée directement en sortie de machine de fabrication, et l'écart (noté y) entre la qualité réalisée et celle visée nous donne une idée sur la reproductibilité et la fiabilité de la ligne de production. Dans cette section, nous analysons la répartition de y à partir des informations reconstituées.

Qualité du produit

Les produits réalisés sont séparés en trois catégories selon l'écart de qualité :

- Un produit i est dit *insuffisant* si $y_i < -2$.
- Un produit i est dit *excessif* si $y_i > 2$.
- Un produit i est dit *bon* si $-2 \leq y_i \leq 2$.

Le choix du seuillage doit prendre en compte la précision des capteurs, les aléas dans la fabrication ainsi que la répartition des individus dans chaque catégorie. L'ingénieur procédé nous a conseillé de considérer un seuillage supérieur à 1, et a validé notre proposition de valeur de seuillage à 2.

Suite à une analyse statistique sur l'écart de qualité, nous avons pu constaté que les produits réalisés immédiatement après le remplissage du système en matières premières ont un comportement spécifique, et que cet ensemble de produits (noté A) présente plus de produits finaux problématiques (i.e insuffisants et excessifs) que la normale. Nous notons B l'ensemble des produits complémentaire de l'ensemble A ($A + B =$ totalité des produits considérés). La répartition des produits dits "insuffisants" et "excessifs" dans les ensembles A et B est affichée le Tableau 4.1 : l'ensemble B a moins de produits problématiques que l'ensemble A. Par la suite, nous analysons uniquement la relation entre la qualité des produits et les informations reconstituées par la méthode de restauration dans l'ensemble B.

	Insuffisant	Bon	Excessif	Nombre de produits
A	19,8%	65,6%	14,6%	5039
B	8,0%	86,4%	5,6%	23484

TABLEAU 4.1 – Répartition des produits excessifs et insuffisants dans l'ensemble A (produits réalisés immédiatement après le remplissage) et l'ensemble B (produits réalisés dans l'état normal de fonctionnement de la machine). Données : ensemble F1.

Influence des maintenances détectées

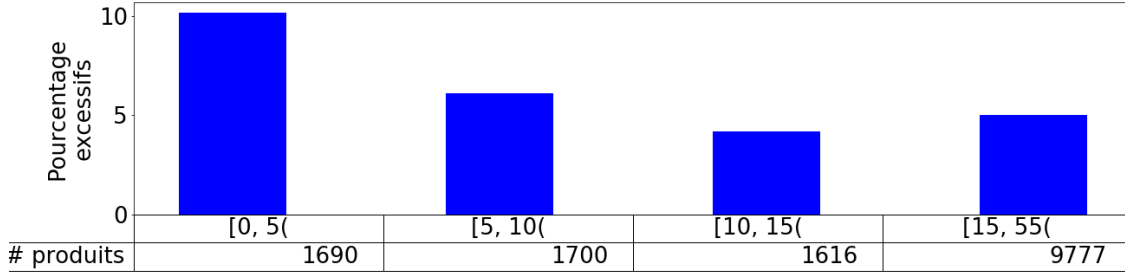
Dans cette section, nous cherchons à comprendre l'influence des actions de maintenance sur la qualité des produits.

Les pourcentages des produits excessifs et insuffisants en fonction de la durée d_g sont affichés dans la Figure 4.5. Les 5 premiers produits réalisés après une maintenance détectée ont un pourcentage plus important de produits anormaux par rapport aux autres produits de l'ensemble B. Nous avons constaté une tendance décroissante des pourcentages de produits anormaux en fonction de la durée d_g . En analysant la répartition des produits anormaux poste par poste, nous avons pu observer que les produits anormaux fabriqués après une maintenance sont souvent tous du même type (excessif ou insuffisant). La machine est, en effet, en régime transitoire après l'action de maintenance, et l'algorithme de contrôle ne retrouve pas toujours immédiatement les bons paramètres de contrôle après cet événement perturbateur.

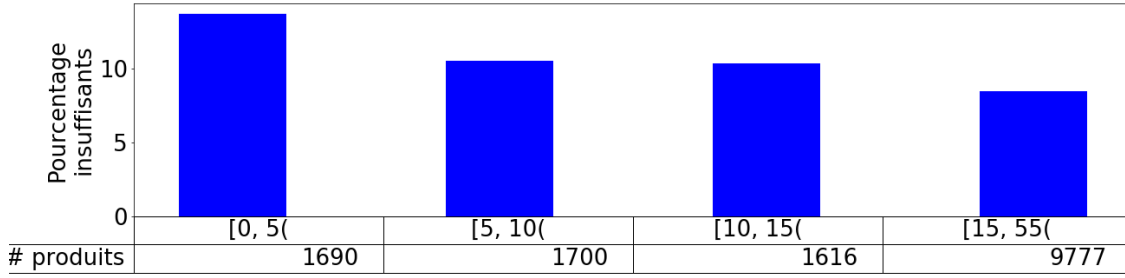
Influence des résidus de restauration

Une autre variable pourrait avoir une influence sur la qualité des produits : le résidu de restauration. Le résidu de restauration d'un produit (e.g. le produit i) est défini par :

$$r_i = t_{c,i} - u_i^* \quad (4.1)$$

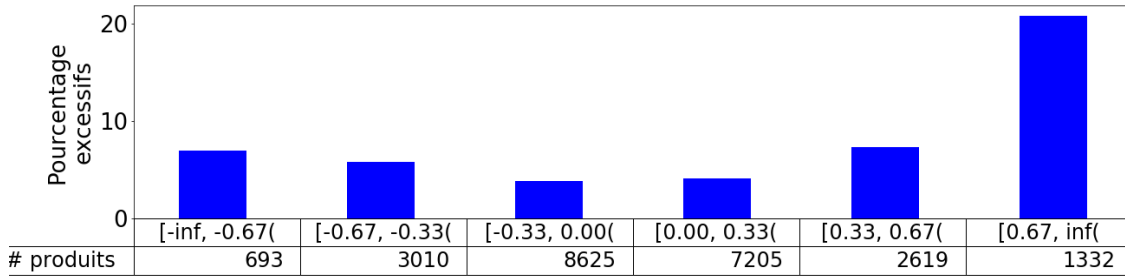


(a) Pourcentages des produits excessifs en fonction de la durée d_g

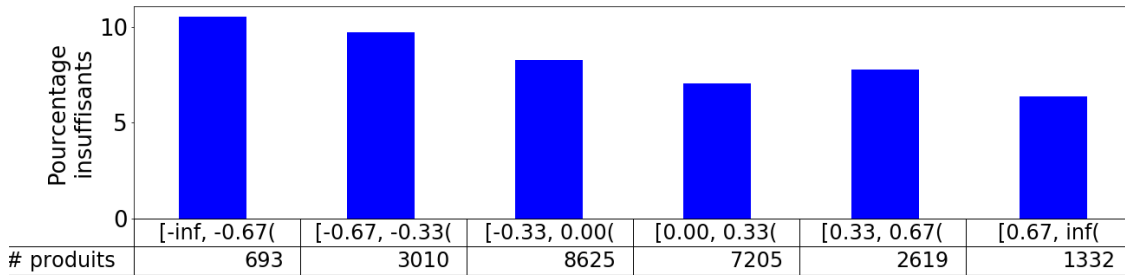


(b) Pourcentages des produits insuffisants en fonction de la durée d_g

FIGURE 4.5 – Pourcentages des produits anormaux en fonction de la durée d_g . La durée correspond au nombre de produits réalisés depuis la dernière maintenance pendant le poste, ou depuis le démarrage du poste au début du poste. Données : ensemble B, inclus dans F1.



(a) Pourcentages des produits excessifs en fonction des résidus de restauration



(b) Pourcentages des produits insuffisants en fonction des résidus de restauration

FIGURE 4.6 – Pourcentages des produits anormaux en fonction des résidus de restauration. Données : ensemble B, inclus dans F1.

avec $t_{c,i}$ la mesure du paramètre t_c pour le produit i et u_i^* la restauration de $t_{c,i}$ par notre méthode proposée. La méthode de restauration propose en effet une segmentation des sous-séquences d'échantillons consécutives sans recouvrement en fonction des comportements locaux des échantillons et bien entendu de la valeur de l'hyper-paramètre λ . Les échantillons voisins ayant une valeur similaire sont

regroupés dans un même palier, et la valeur de restauration est donnée par la valeur moyenne des échantillons dans un palier en ajoutant un biais qui dépend de la relation avec les paliers voisins et de la valeur de λ .

En conséquence, une grande valeur de résidu indique que le paramètre t_c du produit concerné a un comportement spécifique par rapport aux produits voisins (bien que réalisé dans des conditions de fabrication très similaires), comme par exemple une valeur aberrante de t_c . Ces comportements spécifiques sont souvent dus à une mauvaise introduction des matières premières.

Les pourcentages des produits dits “insuffisants” et “excessifs” en fonction des résidus de restauration sont affichés dans la Figure 4.6 : les produits ayant un grand résidu positif ont 20% de produits qualifiés d’excessifs, soit 4 fois plus que celui ayant un faible résidu de restauration. Un résidu positif et important indique en effet que pour le paramètre t_c considéré, une valeur très supérieure a été appliquée pour un produit particulier, ce qui perturbe visiblement la qualité finale du produit.

Analyses sur les données d’une autre usine

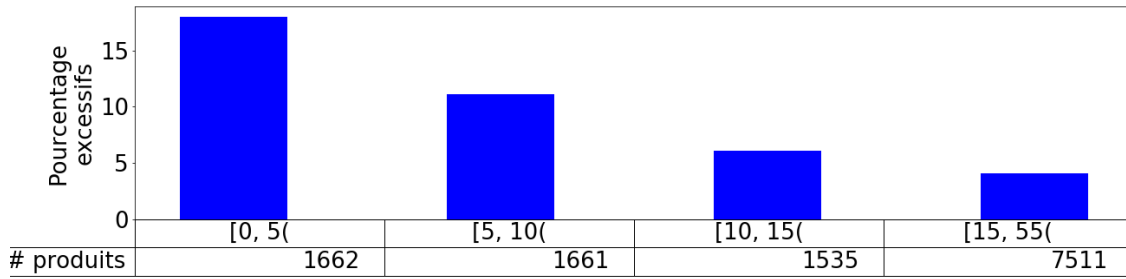
Afin de vérifier si les conclusions obtenues précédemment sur les produits excessifs et insuffisants sont reproductibles, nous avons réalisé le même type d’analyses sur un autre type de produit fabriqué par plusieurs lignes de production en parallèle dans une autre usine (l’ensemble P1). Pour ce type de produit analysé, nous considérons un produit i avec $y_i < -5$ comme insuffisant et $y_i > 5$ comme excessif. Les pourcentages des différentes catégories sont affichés dans le Tableau 4.2 : les pourcentages dans chaque catégorie sont équivalents à ceux impliqués dans les analyses présentées dans la section précédente.

	Insuffisant	Bon	Excessif	Nombre de produits
A	19,5%	67,4%	13,1%	2224
B	6,6%	86,7%	6,7%	14507

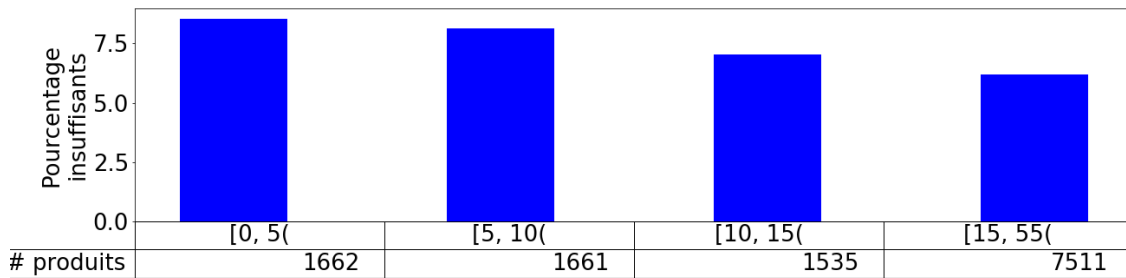
TABLEAU 4.2 – Répartition des produits excessifs et insuffisants dans l’ensemble A (produits réalisés immédiatement après le remplissage) et l’ensemble B (produits réalisés dans l’état normal de fonctionnement de la machine). Données : ensemble P1.

Les pourcentages des produits anormaux en fonction de la durée d_g et du résidu de restauration sont affichés respectivement dans les Figures 4.7 et 4.8. Nous pouvons tirer les mêmes conclusions à partir des analyses sur ce type de produit : les premiers produits réalisés après une maintenance détectée et ceux ayant un comportement spécifique de t_c par rapport aux produits voisins sont plus probablement des produits anormaux.

Les produits anormaux après la maintenance sont un phénomène bien connu dans les usines, tandis que l’analyse sur les résidus de restauration nous apporte une information intéressante : une mesure fiable de t_c permet aux opérateurs et aux algorithmes de contrôle de bien réguler la machine.

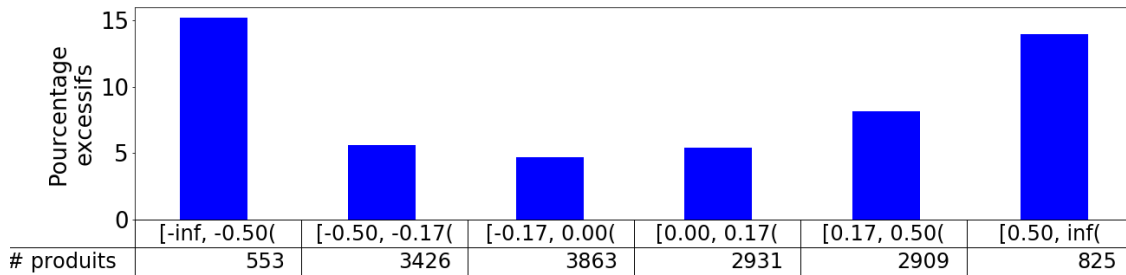


(a) Pourcentages des produits excessifs en fonction de la durée d_g

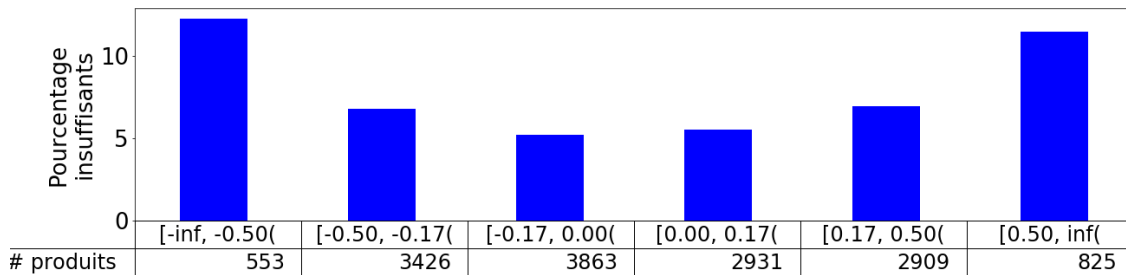


(b) Pourcentages des produits insuffisants en fonction de la durée d_g

FIGURE 4.7 – Pourcentages des produits anormaux en fonction de la durée d_g . La durée correspond au nombre de produits réalisés depuis la dernière maintenance pendant le poste, ou depuis le démarrage du poste au début du poste. Données : ensemble B, inclus dans P1.



(a) Pourcentages des produits excessifs en fonction des résidus de restauration



(b) Pourcentages des produits insuffisants en fonction des résidus de restauration

FIGURE 4.8 – Pourcentages des produits anormaux en fonction des résidus de restauration. Données : ensemble B, inclus dans P1.

4.1.7 Classification de la performance d'un poste de production

Dans cette section, nous cherchons à classer la performance d'un poste de production à partir des caractéristiques des résidus de restauration. Ces analyses sont effectuées sur l'ensemble B inclus dans l'ensemble F1.

L'histogramme des écarts-types de y d'un poste de production est affiché dans la Figure 4.9. Afin de distinguer les postes ayant une bonne ou mauvaise performance, nous fixons le seuillage $std(y) = 15,3$: les mauvais postes sont ceux ayant $std(y) > 15,3$. Ce choix de seuillage nous permet d'avoir une proportion équivalente de bons et mauvais postes : 60 bons postes et 52 mauvais postes pour un type de produit réalisé sur une ligne de production pendant une année.

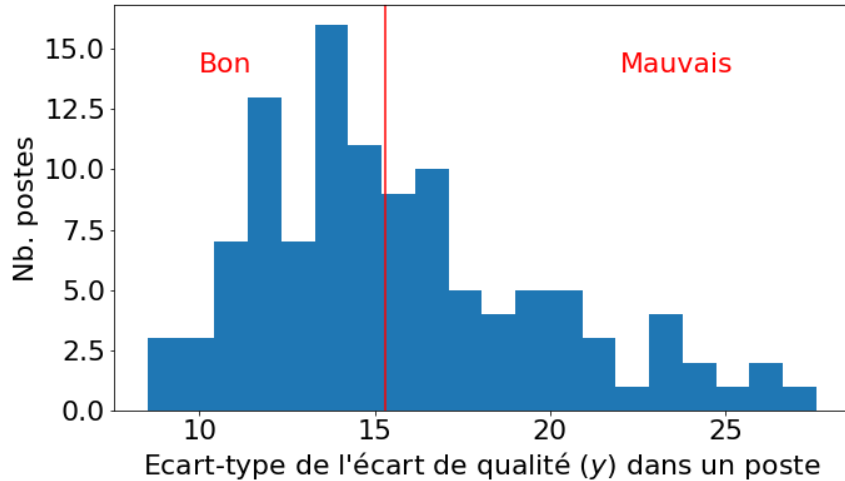


FIGURE 4.9 – Histogramme de l'écart-type de y dans un poste de production. Données : ensemble B, inclus dans F1.

Classification des résidus de restauration

Le résidu de restauration est défini par (4.1). Dans cette section, nous analyserons les fonctions de répartition des résidus poste par poste.

Afin de faciliter les analyses, nous avons d'abord classifié les fonctions de répartition par la méthode classification ascendante hiérarchique (CAH). La distance appliquée est celle de Kolmogorov-Smirnov (KS), définie par :

$$d(F_i, F_j) = \sup_x |F_i(x) - F_j(x)|$$

avec F_i et F_j deux fonctions de répartition. Le dendrogramme avec la distance KS et le saut moyen est affiché dans la Figure 4.10, et trois groupes de résidus sont identifiés : le groupe bleu ne contient que 2 postes.

Les fonctions de répartition des résidus dans un poste sont affichées dans la Figure 4.11. Les résidus dans un poste sont centrés sur 0 en raison du terme d'attachement aux données L^2 dans la fonctionnelle considérée (3.3). Nous pouvons constater que le groupe rouge représente les postes ayant une faible valeur de variance de résidu, tandis que le groupe vert regroupe les postes ayant une variance de résidu plus significative que le groupe rouge. Le groupe bleu regroupe les postes ayant une forte variance de résidu.

Les histogrammes normalisés de l'écart-type de y des groupes verts et rouges sont affichés dans la Figure 4.12. Nous pouvons constater que le groupe vert est plus susceptible d'avoir un poste ayant un grand écart-type de y , ce qui correspond à une mauvaise performance de poste. L'écart-type des résidus de restauration représente en fait la puissance moyenne du bruit de mesure de t_c dans un poste, donc un poste qui maîtrise mieux la machine dont la mesure de t_c aurait plus souvent une bonne performance de production.

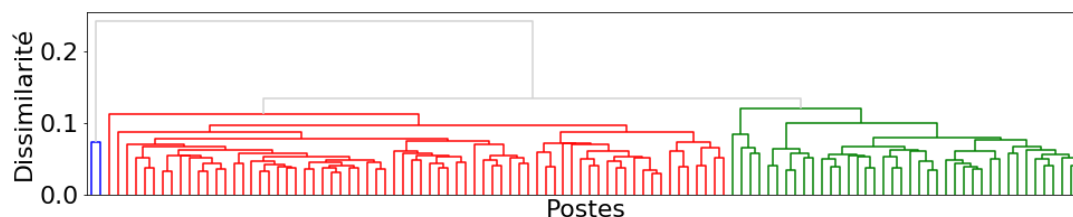


FIGURE 4.10 – Classification des résidus de restauration : ce dendrogramme utilise la distance Kolmogorov-Smirnov et le saut moyen. Les groupes identifiés sont représentés par les branches ayant une même couleur dans le dendrogramme. Chaque feuille de l'arbre binaire affiché correspond à un poste de production. Données : ensemble B, inclus dans F1.

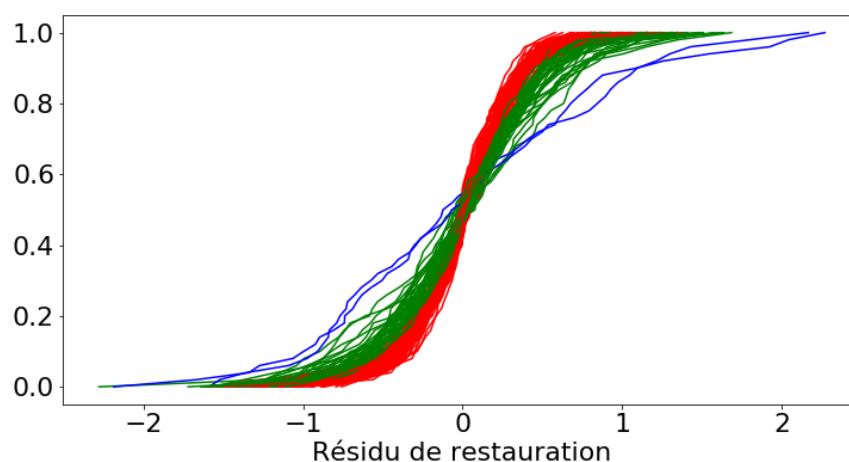


FIGURE 4.11 – Illustration des fonctions de répartition du résidu de restauration pour les différents postes de production considérés. Une courbe correspond à un poste de production. Chaque couleur représente un groupe identifié par la méthode CAH (c.f. Figure 4.10). Données : ensemble B, inclus dans F1.

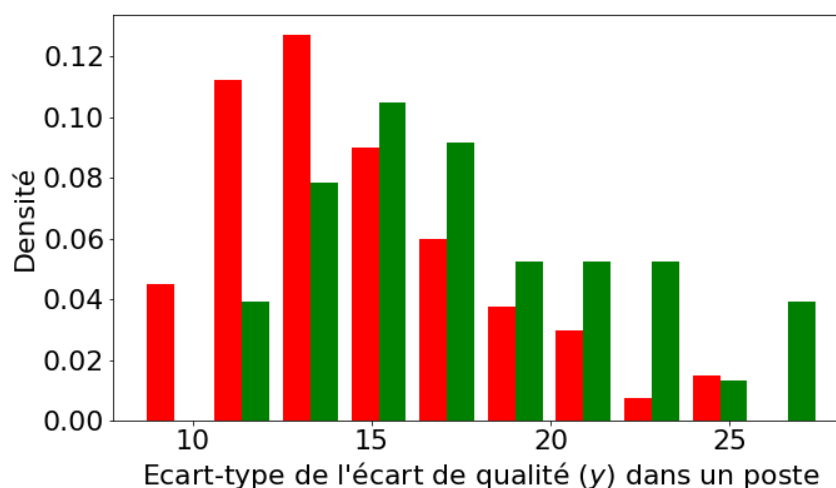


FIGURE 4.12 – Histogramme normalisé de l'écart-type de l'écart de qualité (y) d'un poste dans les groupes verts et rouges identifiés par la CAH (c.f. Figure 4.10). Données : ensemble B, inclus dans F1.

4.1.8 Liens entre la performance d'un poste et l'écart-type des résidus

Dans cette section, nous analysons la répartition des bons et mauvais postes avec l'écart-type des résidus sur les différents types de produit.

Dans la Figure 4.13, nous pouvons visualiser les histogrammes normalisés de l'écart-type des résidus des bons et mauvais postes d'un produit donné réalisé sur une ligne de production pendant une année (ensemble B, inclus dans F1), et les bons postes sont plus susceptibles d'avoir une faible variance des résidus.

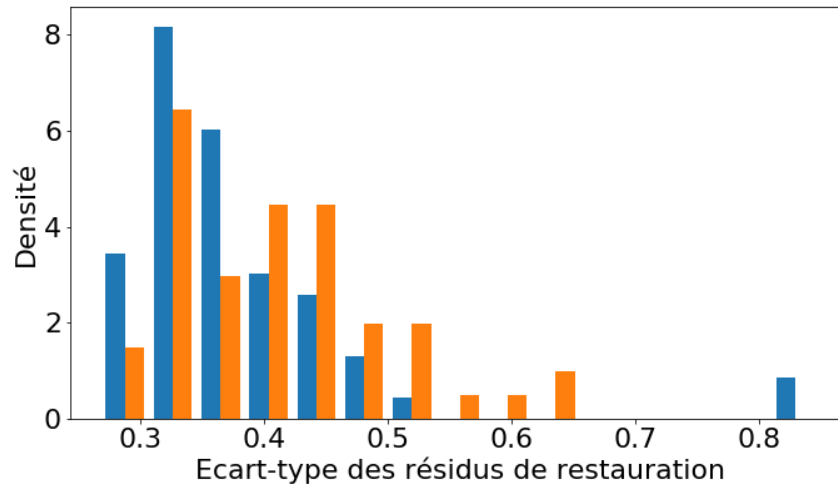


FIGURE 4.13 – Histogramme normalisé de l'écart-type des résidus de restauration. Couleurs : bon postes et mauvais postes. Données : ensemble B, inclus dans F1.

Une telle analyse est aussi effectuée sur les deux autres types de produits réalisés par plusieurs lignes de production pendant une année, et les résultats sont affichés sur la Figure 4.14. Nous pouvons constater à nouveau que les postes ayant une forte variance des résidus ont souvent une mauvaise performance.

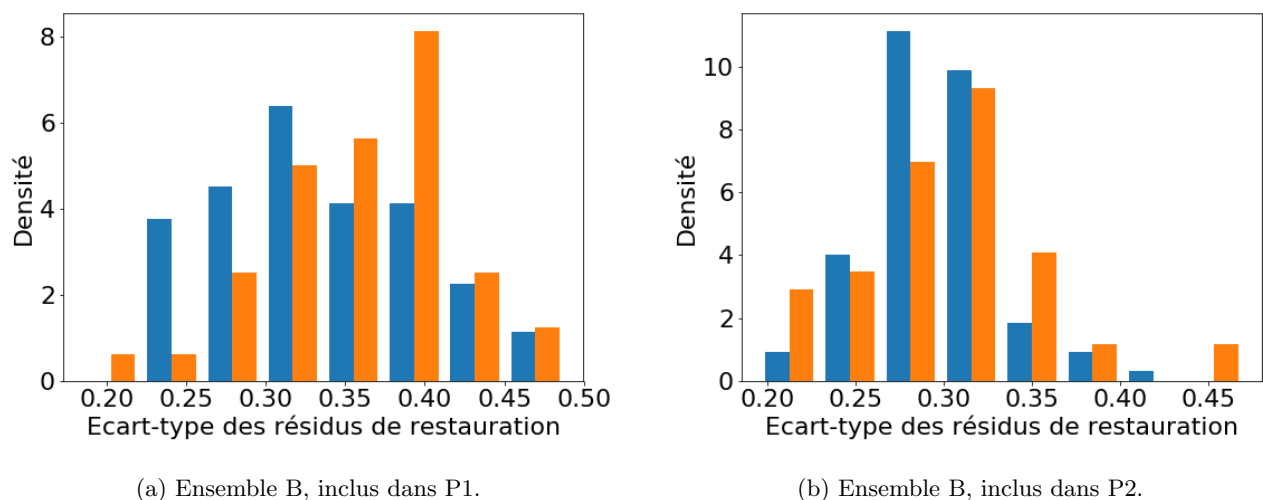


FIGURE 4.14 – Histogrammes normalisés de l'écart-type des résidus de restauration. Couleurs : bon postes et mauvais postes.

En résumé, suite aux analyses sur plusieurs types de produit, les postes ayant une faible variance de résidus de restauration montrent une tendance à avoir une faible dispersion de l'écart de qualité. Les postes qui maîtrisent le versement des matières premières pourraient avoir moins d'aléas dans la mesure de t_c , donc une bonne performance.

4.1.9 Conclusion

Dans cette section, nous avons appliqué les méthodes de restauration proposées pour analyser un signal t_c mesuré dans un procédé de Saint-Gobain. Ce signal est l'un des facteurs clés pour le bon fonctionnement du procédé d'après l'intuition des opérateurs et de l'ingénieur process impliqués, et les maintenances d'un composant de la machine introduisent des descentes brusques de ce signal. Nous avons proposé une méthode pour détecter les descentes brusques à partir du signal restauré, et les résultats sur les productions des différentes lignes nous montrent que les maintenances détectées ne correspondent pas toujours aux indices de "maintenances probables".

Des informations issues de la restauration du signal du paramètre t_c ont été introduites dans la modélisation de l'écart de qualité du produit, soit l'écart entre la qualité réalisée et celle visée. Le meilleur modèle a un score R^2 de 0,308, soit +0,018 que le même type de modèle sans ces nouvelles variables issues de la restauration. L'amélioration apportée par les informations reconstituées reste faible, et le modèle additif généralisé n'a pas un score satisfaisant, ce qui implique que :

- Certains facteurs importants dans la qualité du produit ne sont pas inclus dans les données à notre disposition.
- Le couplage des phénomènes est plus complexe : par exemple, la dépendance entre les produits voisins.

Nous avons analysé les produits anormaux et la performance des postes à partir des informations reconstituées sur les différents types de produits fabriqués par deux usines. Les premiers produits réalisés après une action de maintenance et ceux ayant une valeur de t_c spécifique par rapport aux produits voisins sont souvent problématiques. D'ailleurs, les analyses montrent que les postes ayant un faible écart-type des résidus de restauration sont plus susceptibles d'avoir une bonne performance, ce qui indique que le fait de bien gérer l'ensemble du procédé minimise l'occurrence des produits anormaux. En outre, ces analyses effectuées nous indiquent qu'il est probable que certains facteurs de premier ordre ne soient pas remontés dans la base de données, ce qui a donné lieu à des actions d'implémentation de nouveaux capteurs sur les lignes.

4.2 Applications autour d'une mesure de qualité en 2D

Dans cette section, nous présentons les analyses d'une mesure 2D de la qualité d'un produit fabriqué par Saint-Gobain, à l'aide des méthodes de restauration proposées dans le Chapitre 3.

L'objectif est de visualiser les différentes variations de mesures par ailleurs très bruitées.

4.2.1 Présentation du signal collecté

Le procédé considéré, appelé *Process 2* dans cette section fabrique en continu un produit ayant une certaine qualité visée. Le produit est déposé sur un convoyeur qui avance avec une vitesse constante. Une série de capteurs est installée transversalement à la sortie d'une étape de fabrication. Ces capteurs mesurent en continu la qualité du produit sur toute sa largeur, et permettent donc de suivre en temps réel la qualité réelle du produit, qui peut être considéré ici comme un ruban fini. Les 24 capteurs sont répartis de manière homogène sur la largeur du produit, et mesurent avec une fréquence d'échantillonnage fixée.

Nous noterons le vecteur des mesures du i^e capteur $\mathbf{z}_i = (z_{i,1}, \dots, z_{i,n})$ avec n le nombre d'échantillons collectés. Les mesures des 24 capteurs forment une structure de grille, assimilable à une image 2D, notée Z , ayant une largeur de 24 pixels et une longueur finie :

$$Z = \{z_{i,j}\}_{i=1,\dots,24, j=1,\dots,n}$$

Dans la suite, nous appellerons ligne de mesure la mesure simultanée des 24 capteurs dans la largeur du produit : $\mathbf{z}^j = (z_{1,j}, \dots, z_{24,j})$ pour un j donné.

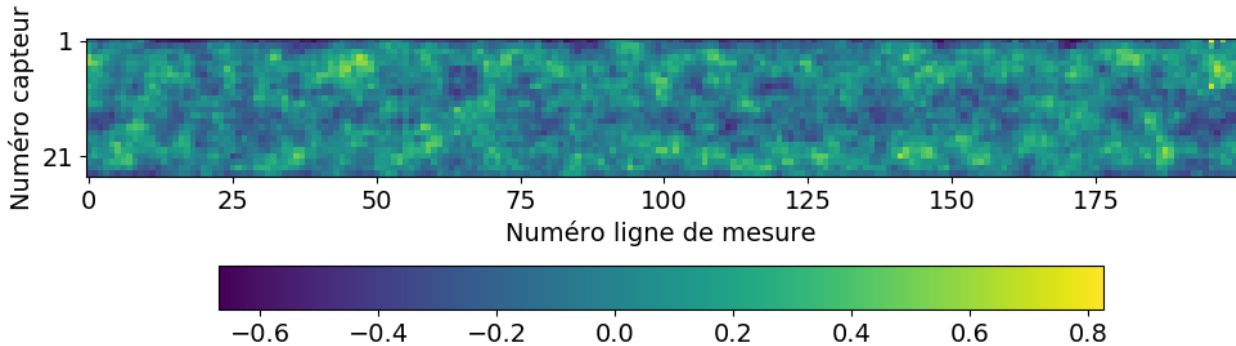


FIGURE 4.15 – Mesure brute de qualité : exemple de 200 lignes de mesure. Plus la couleur est claire, plus la valeur de la mesure est grande. L'objectif serait d'homogénéiser tout le produit à la valeur de mesure de zéro, qui est la qualité visée.

Un exemple de 200 lignes de mesure est affiché sur la Figure 4.15. Nous pouvons constater que les mesures ont des régions en sur-qualité (en couleurs claires) et en sous-qualité (en couleurs foncées) : par exemple, les mesures entre la 125^e et la 150^e ligne ont une région en sous-qualité au centre du produit. Cependant, les motifs des mesures (i.e. la variation de qualité) sont très bruités. Une étape de débruitage faciliterait donc l'identification et l'interprétation des régions ayant une qualité que l'on pourrait qualifier d'anormale (au sens de sortant de la norme).

Les origines du bruit sont différentes selon les deux directions : le bruit le long d'une ligne horizontale (donc correspondant à un capteur donné) peut provenir du bruit de mesure du capteur considéré et de certaines variations du procédé ; tandis que le bruit vertical (le long de la ligne de capteurs) peut être dû aux bruits de mesures des différents capteurs, à la différence entre les capteurs et à d'autres variations du procédé lui-même.

Nous avons imposé les hypothèses suivantes dans l'analyse de la qualité :

- Bruit additif : les mesures brutes Z correspondent à la qualité réelle de produit (notée $U_{net} \in \mathbb{R}^{24 \times n}$, inconnue en réalité) à laquelle les bruits ϵ sont ajoutés :

$$Z = U_{net} + \epsilon$$

avec $\epsilon = (\epsilon_1, \dots, \epsilon_{24})^T$ où $\epsilon_j = (\epsilon_{j,1}, \dots, \epsilon_{j,n})$ est le vecteur du bruit de mesure du j^e capteur.

- Caractéristique du bruit : pour chaque capteur (e.g i^e capteur), nous supposons que $\mathbb{E}(\epsilon_i) = 0$ et $\mathbb{V}(\epsilon_i) = \sigma_i^2$.
- Différence entre les 24 capteurs : nous supposons que les capteurs sont correctement calibrés, donc $\sigma_i = \sigma$ pour tout i . Autrement dit, en négligeant les différences entre les capteurs, nous supposons que les bruits sont isotropes sur les deux directions.

4.2.2 Présentation des données

Nous avons analysé six postes de production fabriquant le même type de produit. Ces postes sont réalisés durant différentes périodes de l'année. D'ailleurs, les postes analysés durent tous plus de deux heures, ce qui nous permet de comprendre le fonctionnement établi de la machine. Dans cette section, les Figures affichées sont tracées avec un ensemble de mesures d'un poste de fabrication (appelé ensemble E2).

4.2.3 Critères des bons et mauvais postes

La qualité des produits finaux nécessite différentes mesures, dont certaines sont réalisées hors-ligne, pour être parfaitement caractérisée. La mesure continue de qualité présentée ici, bien que ne caractérisant que partiellement le produit, est donc liée à la qualité finale du produit et au rendement du procédé. Une région mesurée ici comme étant en sous-qualité correspondra à des propriétés insuffisantes du produit final, tandis que la réalisation d'une zone de sur-qualité consommera plus de matières premières et d'énergie pour une même quantité de produit final, ce qui diminuera donc le rendement du procédé. En résumé, un bon poste correspond à une production ayant des mesures continues de qualité homogènes et centrées sur zéro.

Afin de faciliter le suivi de la qualité durant la production, les 24 capteurs sont répartis en 8 groupes, notés g_i avec $i = 1, \dots, 8$. Chaque groupe contient 3 capteurs successifs, illustré sur la Figure 4.16 :

$$g_i = \{3i - 2, 3i - 1, 3i\}$$

g_1			g_2			g_3			g_4			g_5			g_6			g_7			g_8		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Partie gauche									Partie centrale						Partie droite								

FIGURE 4.16 – Illustration des groupes de capteur. Le numéro de capteur est affiché dans la cellule de tableau. Chaque groupe contient 3 capteurs successifs.

À des fins de simplification, nous appellerons $\{g_2, g_3\}$ la partie gauche, $\{g_4, g_5\}$ la partie centrale et $\{g_6, g_7\}$ la partie droite. Les valeurs moyennes de différentes parties sont illustrées sur la Figure 4.17.

Certains critères caractérisant l'inhomogénéité verticale (dans une même ligne de mesure) sont suivis pendant la production afin de paramétrer correctement la machine. Parmi lesquels, le critère *Core* représente l'inhomogénéité entre la partie centre (g_4 et g_5) et les bords du produit (g_2, g_3, g_6 et g_7) :

$$\text{core}_j = \frac{2 \sum_{i \in \{g_4, g_5\}} z_{i,j}}{\sum_{i \in \{g_2, g_3, g_6, g_7\}} z_{i,j}} \quad (4.2)$$

4.2.4 Analyse des inhomogénéités transversales

Nous avons d'abord analysé les caractéristiques statistiques des lignes de mesures. L'objectif de cette analyse était d'identifier les variations de qualité au cours de la fabrication.

La variation des médianes de 6000 lignes de mesure est affichée sur la Figure 4.18 : la restauration par la variation totale et celle par la moyenne mobile sur 600 points nous indiquent une variation de l'ordre 0,02, ce qui correspond à environ 2% de la variation des mesures brutes. En outre, nous pouvons observer un profil périodique sur les restaurations proposées.

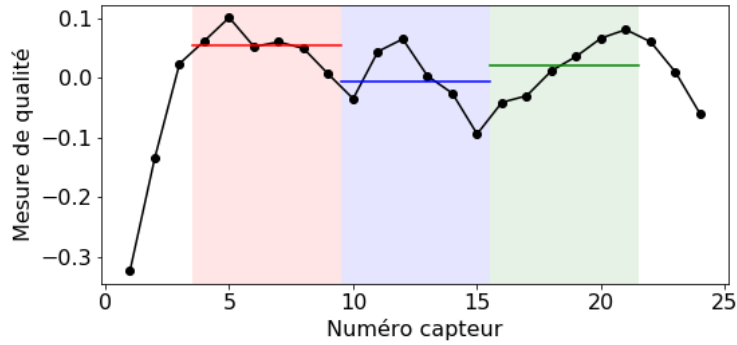
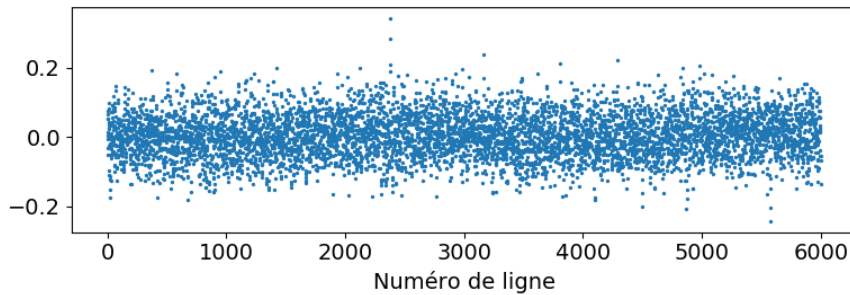
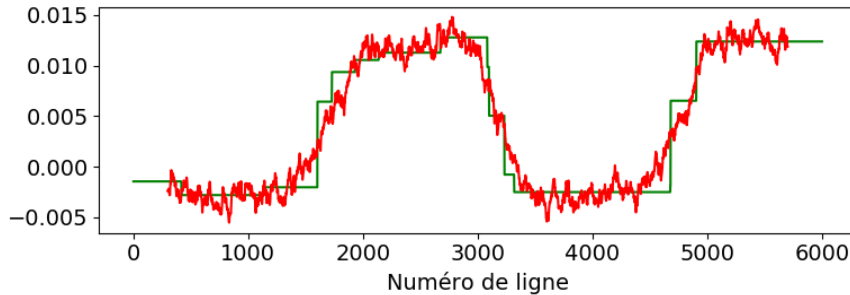


FIGURE 4.17 – Illustration de la découpe des parties dans une ligne de mesure. La ligne noire correspond au profil moyen d'une production d'une heure. Les valeurs moyennes dans chaque partie sont représentées par une ligne avec la couleur correspondante. Abscisse : le numéro de capteur, ordonnée : la mesure de qualité. Couleurs : **partie gauche**, **partie centrale** et **partie droite**.



(a) La médiane brute des lignes de mesures ($\text{médiane}(z^j)$ pour $j = 1, \dots, 6000$) au cours d'une production



(b) La médiane restaurée du même poste de production

FIGURE 4.18 – Illustration des caractéristiques statistiques de 6000 lignes de mesures lors d'un poste de production. (a) : la médiane des lignes de mesure. (b) : la restauration des médianes proposée par **notre méthode automatique** et **la moyenne mobile sur 600 points**. Données : ensemble E2.

Afin de comprendre la périodicité transversale dans la production, nous avons analysé plusieurs longues périodes de production. La restauration des quantiles par ligne de mesure durant une longue production est affichée sur la Figure 4.19, et nous pouvons constater que les quantiles analysés varient d'une manière synchronisée, ayant tous une période d'environ 3000 lignes. La variation synchronisée de différents quantiles indique que la qualité de produit varie d'une manière périodique au cours de la fabrication : les régions de sous- et sur-qualité alternent, ce qui a un impact non-négligeable à la qualité de produit et au rendement du procédé.

Afin d'identifier la période de variation des quantiles, nous avons effectué un FFT sur les valeurs moyennes des lignes de mesure pour une longue production. Les résultats sont affichés sur la Figure

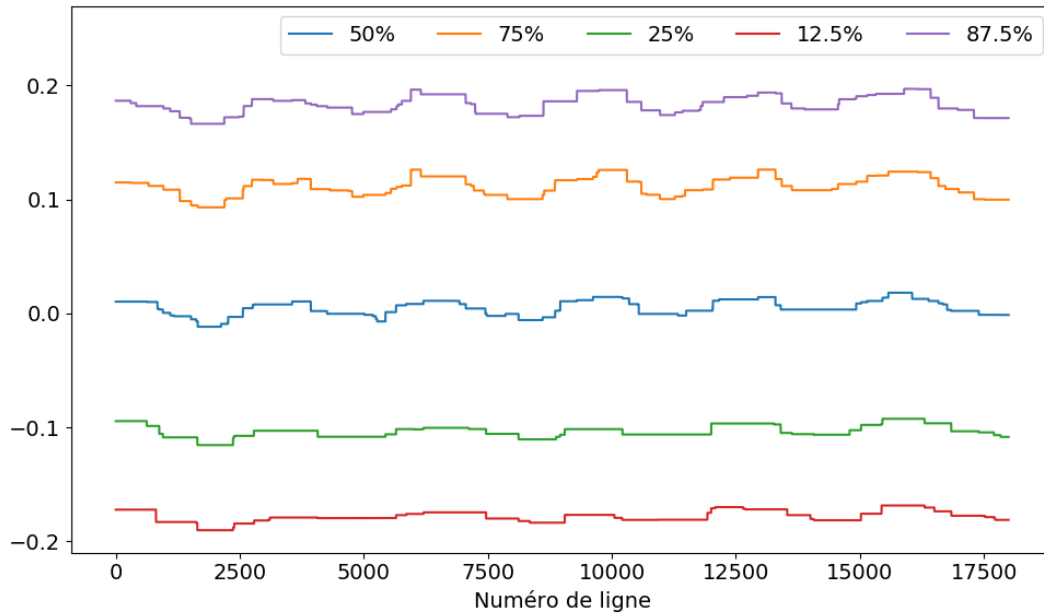


FIGURE 4.19 – Illustration des restaurations des quantiles de mesure de qualité pour une longue production. Couleurs : quantile 87,5%, quantile 75%, médiane, quantile 25% et quantile 12,5%. Données : ensemble E2.

4.20. Nous pouvons identifier 3 composantes de variation qui ressortent particulièrement. Les oscillations observées sur la Figure 4.19 correspondent à la variation de plus grande période (composante 1).

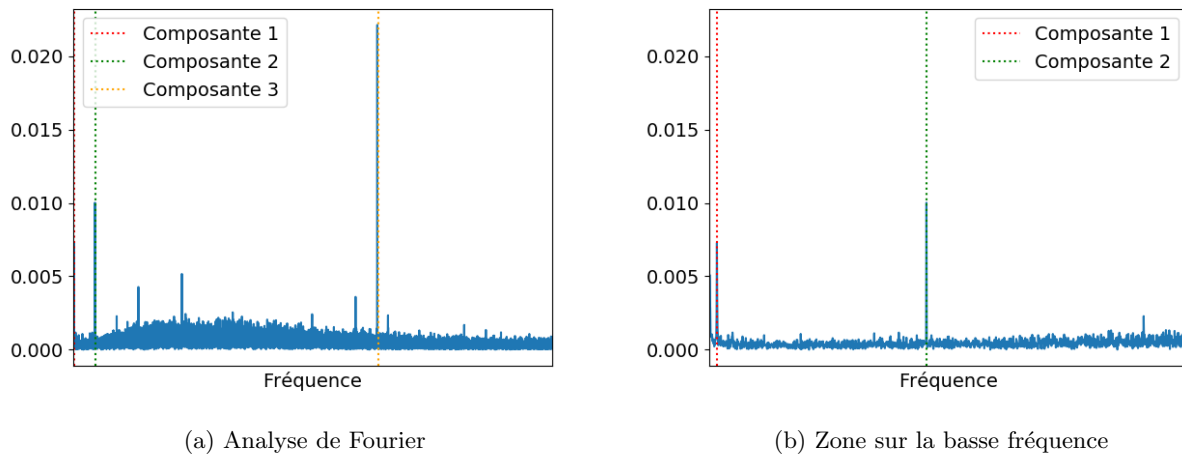


FIGURE 4.20 – Analyse de Fourier des valeurs moyennes des lignes de mesure ($\text{moyenne}(z^j)$ pour $j = 1, \dots, n$) sur une longue production. Trois composantes périodiques se détachent. Données : ensemble E2.

Nous avons analysé plusieurs postes de production réalisés dans les différentes périodes de l'année, et la variation des quantiles avec cette même période caractéristique est aussi observée. Cette variation n'est donc pas occasionnelle. Elle est indépendante des conditions de fabrication souvent changées entre deux postes (par exemple, la température ambiante ou les opérateurs), et elle pourrait être liée à des conditions de fabrication stables (et notamment à la régulation de certains actionneurs).

La variation périodique présentée sur la Figure 4.19 n'est pas connue dans l'usine. Les quantiles bruts de mesure de qualité sont traités par la moyenne mobile sur une grande fenêtre afin de bien atténuer le bruit de mesure. Par conséquence, cette variation périodique est lissée par l'effet de moyennage sur une grande fenêtre. Cela souligne l'importance du choix de l'hyper-paramètre : avec une taille

correcte de la fenêtre (c.f. Figure 4.18b), la mobile moyenne pourrait reconstituer cette variation périodique. Cependant, il est difficile de choisir une taille correcte de fenêtre sans connaissances a priori sur la forme du signal original. Notre méthode de restauration avec la détermination de l'hyper-paramètre permet de reconstituer les motifs récurrents d'un signal très bruité sans connaissances a priori, et elle pourrait être adaptée aux différents cas d'applications.

Cette variation transversale impacte l'homogénéité du produit réalisé. Il est donc important d'identifier les causes de ces variations. L'oscillation avec la période la plus courte semble correspondre à certains mouvements répétitifs se déroulant à l'approvisionnement de matières premières. Les causes des autres oscillations restent à déterminer.

4.2.5 Conclusion

Dans cette section, nous avons appliqué notre méthode de restauration pour visualiser la variation dans une mesure de qualité en deux dimensions. En nous intéressant aux restaurations des quantiles des mesures, nous avons identifié une variation périodique caractéristique au cours de la production, qui pourrait avoir un impact non-négligeable sur la qualité des produits finaux. L'identification des causes de cette variation est importante pour l'amélioration de la qualité du produit et aussi du rendement du procédé, et les investigations sont en cours.

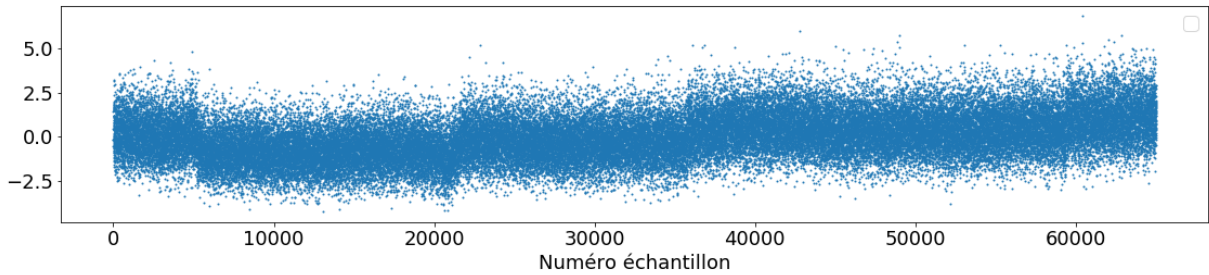
4.3 Applications autour d'une mesure de qualité en temps réel

Un nouvel instrument de mesure est en cours de déploiement sur un procédé de Saint-Gobain, et cet appareil de mesure sert à contrôler le procédé. Dans cette section, nous proposons une méthode en ligne pour détecter les changements brusques et visualiser les petites variations du signal entre deux changements dans un flux de données afin de faciliter le travail quotidien des opérateurs.

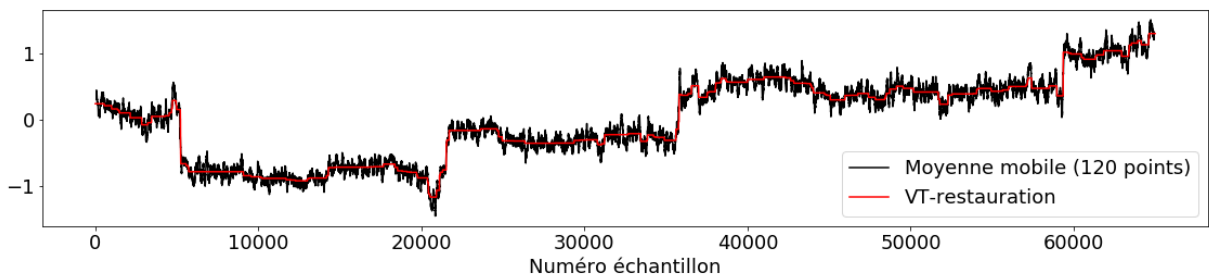
4.3.1 Présentation des données

Le procédé considéré dans cette section fabrique en continu un produit ayant une certaine qualité visée. Un capteur situé en fin de ligne de production mesure en continu la qualité du produit, et permet donc de suivre en temps réel l'état de fonctionnement du procédé en cours. Le vecteur des échantillons est noté $\mathbf{z} = (z_1, \dots, z_n)$ avec n le nombre d'échantillons et $z_i \in \mathbb{R}$ pour tout i .

Le capteur concerné est installé dans des conditions de travail extrêmes, ce qui rend le signal collecté très bruité. Les opérateurs modifient parfois les paramètres de machine, ce qui provoque un changement d'état de fonctionnement du procédé et donc un changement brusque des valeurs mesurées des échantillons $\mathbf{z}$. Un exemple des échantillons collectés pendant plusieurs jours de production (appelé *ensemble E3*) est affiché sur la Figure 4.21a, et nous pouvons constater que le signal fortement bruité a quelques changements de paliers. Les restaurations par la moyenne mobile sur une fenêtre de 120 points et celle par la VT-restauration sont affichées sur la Figure 4.21b : nous pouvons observer 4 changements brusques des valeurs moyennes, et des petites variations entre deux changements brusques successifs sont mises en évidence par les restaurations proposées.



(a) Signal brut. Chaque échantillon est représenté par un point bleu.



(b) Restaurations du signal. Couleurs : VT-restauration avec notre proposition de λ et moyenne mobile sur une fenêtre de 120 points.

FIGURE 4.21 – Un exemple de signal collecté durant plusieurs jours de production. Données : ensemble E3.

Afin de visualiser la variation du signal en ligne, la méthode de la moyenne mobile sur une fenêtre de 120 points est actuellement appliquée aux mesures brutes dans l'usine. Le choix de la taille de fenêtre est en effet un compromis entre l'effet de débruitage et la ponctualité des changements brusques : une grande fenêtre permet de bien atténuer le bruit de mesure, mais les changements brusques sont lissés par la moyenne et détectés avec d'autant plus de retard que la fenêtre est grande. Ce dernier point a un fort impact pour les opérateurs : après avoir modifié un paramètre opératoire entraînant une variation du signal mesuré, il faut attendre l'établissement (i.e. 120 échantillons) de la moyenne mobile après la

rupture pour déterminer l'effet introduit à la machine. La méthode de la moyenne mobile n'est donc pas adaptée pour suivre rapidement l'impact dû à des changements de paramètres opératoires.

4.3.2 Méthode proposée : détection en ligne des ruptures

Nous sommes en train de développer une méthode en ligne fondée sur la VT-restauration pour restaurer le signal mesuré tout en gardant les changements brusques présentés. La moyenne mobile, étant simple à implémenter, lisse les changements brusques. Pour améliorer ce point faible, une première proposition est une méthode hybride de la VT-restauration et la moyenne mobile. L'idée est de détecter les ruptures sur un flux de données et de proposer une restauration par la moyenne mobile sur une longue fenêtre sans considérer les échantillons avant les ruptures.

Pour cela, nous avons proposé une méthode de détection en ligne utilisant la dynamique des VT-restaurations. La détection se sert d'un indicateur estimé dans une fenêtre glissante. La performance de la méthode proposée dépend de la rapidité et de la précision des détections des ruptures. Pour une fenêtre $w_i^m = [i, i + m - 1]$ ayant une rupture au point $j \in w_i^m$, la rapidité de la détection $\hat{j} \in w_i^m$ est donnée par $R(\hat{j}, w_i^m) = i + m - 1 - \hat{j}$, le nombre d'échantillons supplémentaires après la rupture nécessaire à sa détection, tandis que la précision est donnée par $Pr(j, \hat{j}) = |j - \hat{j}|$. Une bonne détection $\hat{j}$ devrait être à la fois rapide et précise, donc avoir une faible valeur de $R(\hat{j}, w_i^m)$ et $Pr(j, \hat{j})$.

Nous avons introduit les hypothèses suivantes pour la méthode proposée :

- La variance du bruit de mesure est constante pendant la production. Autrement dit, nous supposons que le capteur fonctionne correctement, et qu'aucune dérive de capteur n'a eu lieu.
- Il y a au plus un changement brusque dans une fenêtre considérée.

Les deux paramètres suivants sont adaptés pour indiquer la présence d'une rupture dans la fenêtre :

- λ_2 , étant la valeur minimale de λ donnant la restauration avec un seul palier.
- λ_2^g , étant la valeur minimale de λ donnant une restauration monotone, donc $g(\lambda_2^g) = 2$.

Suivant les théorèmes 3.1 et 3.2, nous avons $\lambda_2 > \lambda_2^g$. Basée sur la dynamique des restaurations, nous avons les intuitions suivantes :

- Pour une fenêtre ayant une rupture, nous pouvons avoir une restauration monotone lorsque les points des deux côtés de la rupture sont suffisamment regroupés. Par contre, pour avoir la restauration avec un seul palier, il faut une augmentation significative de la valeur de λ pour regrouper les points de chaque côté de la rupture.
- Pour une fenêtre sans rupture, une petite augmentation de λ permettrait d'avoir la restauration avec un seul palier à partir de la restauration monotone.

Notons $\psi = \lambda_2 - \lambda_2^g$ et $\psi = (\psi_1, \dots, \psi_{n-m})$ avec ψ_i l'estimation pour la fenêtre w_i^m , nous proposons de suivre la quantité ψ_i pour chercher les fenêtres contenant un changement brusque : les fenêtres ayant une rupture sont données par une grande valeur de ψ_i , par exemple $\{i | \psi_i > \alpha\}$ avec α un seuillage à préciser. Un exemple d'évolution de ψ_i est visible sur la Figure 4.22.

Pour une fenêtre w_i^m , la rupture potentielle r_i est donnée par le changement de niveaux de la restauration $u^*(\lambda_3)$ avec λ_3 la valeur minimale de λ donnant la restauration avec deux paliers :

$$r_i = \{j | u_j^*(\lambda_3) \neq u_{j+1}^*(\lambda_3)\}$$

avec $j \in w_i^m$ et $u^*(\lambda_3)$ la restauration de w_i^m avec deux paliers.

Au final, l'ensemble de ruptures détectées est donné par :

$$R = \{r_i | \psi_i > \alpha\} \quad (4.3)$$

Cette méthode, appelée *détection locale*, est rapide : les paramètres $(\lambda_2, \lambda_2^g, \lambda_3)$, étant les éléments de la liste Λ° , sont directement calculés à partir de nos différents algorithmes proposés. Avec l'implémentation adaptée aux fenêtres glissantes (Algorithme 4), nous pouvons estimer l'ensemble de ruptures R en $O(m(n - m))$ avec m la taille de fenêtre et n la taille du signal.

4.3.3 Sélection des paramètres

Deux paramètres restent à déterminer dans la méthode de détection locale, respectivement la taille de fenêtre m et le seuillage α .

Taille de la fenêtre

La taille de fenêtre joue un rôle important dans la rapidité et la précision de la méthode proposée.

La fenêtre a une grande valeur de ψ lorsque la rupture se situe au milieu de la fenêtre : la fusion de deux longues paliers (respectivement de part et d'autre de la rupture) demande une grande augmentation de λ suite aux analyses de la dynamique de restauration présentées dans la Section 3.1.3.

Pour une grande fenêtre, comme on a suffisamment de points des deux côtés de la rupture, nous pouvons facilement détecter un grand saut noyé dans le bruit en éliminant les influences des petits sauts. Cependant, la rapidité de détection avec une longue fenêtre serait limitée car il faut avoir un nombre de points similaire avant et après la rupture pour en détecter. D'autre part, une petite taille de fenêtre permet d'avoir une estimation rapide de ψ_i et une bonne rapidité de détection, mais la distinction entre les petits et les grands sauts est moins évidente.

Les paramètres ψ d'un exemple des données (c.f. Figure 4.21a) estimés à partir des fenêtres glissantes avec $m \in \{100; 200; 500; 1000\}$ sont affichés sur la Figure 4.22. Nous pouvons constater que les plages de valeurs de i consécutives ayant de grandes valeurs de ψ (on en voit bien 4 par exemple sur la Figure 4.22d) correspondent bien aux ruptures affichées sur la Figure 4.21b, ce qui a validé notre intuition. Ces plages sont de moins en moins évidentes avec la diminution de la taille de fenêtre : par exemple, la deuxième rupture autour de $i = 20\,000$ n'a pas une valeur de ψ_i bien distincte de celle des fenêtres voisines pour $m = 100$.

Après plusieurs expérimentations, la fenêtre avec $m = 200$ nous paraît adaptée pour la détection en ligne des changements brusques.

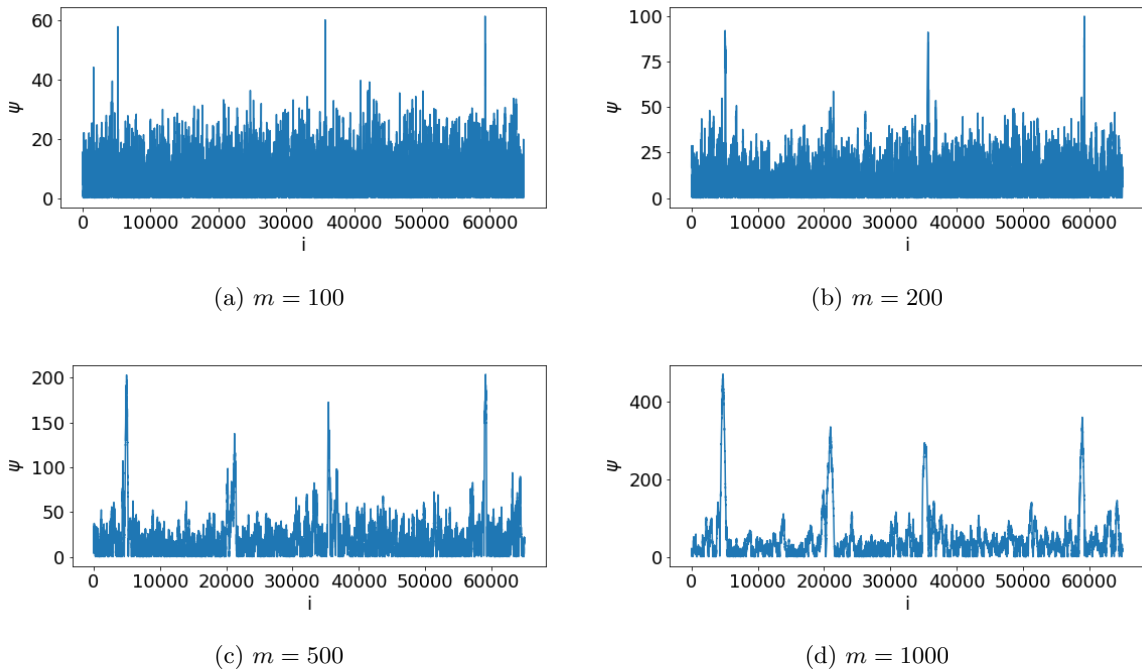


FIGURE 4.22 – Estimation de ψ fondée sur les fenêtres glissantes w_i^m avec différentes tailles de fenêtre (m). Abscisse : index i de la fenêtre w_i^m . Données : ensemble E3.

Seuillage de la détection

Le seuillage α dans (4.3) permet de détecter les plages ayant une grande valeur de ψ , qui correspondent aux fenêtres contenant un changement brusque. La valeur de α devrait être grande pour capturer uniquement les fenêtres ayant un changement brusque. Cependant, les différentes expérimentations montrent qu'une grande valeur de α réduit la rapidité de méthode de détection. Donc, nous devons faire un compromis entre la précision et la rapidité de détection.

L'histogramme de ψ avec $m = 200$ est affiché sur la Figure 4.23, et nous pouvons constater que le nombre de points diminue rapidement en fonction de ψ , et qu'un changement de tendance a eu lieu pour $30 < \psi < 40$. Le choix $\alpha = 40$, nous donnant peu de fenêtres candidates, nous semble adapté à cette application.

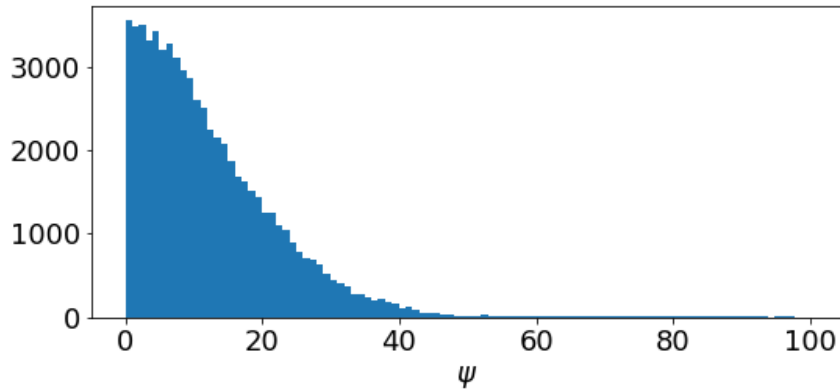


FIGURE 4.23 – Histogramme de ψ avec les fenêtres glissantes de la taille $m = 200$. Le signal brut est affiché sur la Figure 4.21a. Données : ensemble E3.

4.3.4 Méthode de détection robuste

Afin d'avoir une méthode robuste pour détecter uniquement les changements brusques avec un seuillage α relativement faible, nous proposons de détecter les ruptures par les premières séquences $(\psi_{i-p+1}, \dots, \psi_i)$ telles que $\psi_j > \alpha$ pour tout $j \in [i - p + 1, i]$ avec p un paramètre fixé. Formellement, l'ensemble de ruptures est donné par :

$$R_{robuste} = \{r_i | \psi_j > \alpha \text{ pour } j = i - p + 1, \dots, i \text{ et } \psi_{i-p} \leq \alpha\} \quad (4.4)$$

Après plusieurs expérimentations, $p \geq 20$ donne des détections précises.

4.3.5 Résultats de détection

Nous avons appliqué la méthode de détection locale avec $m = 200$, $\alpha = 40$ et $p = 20$ sur plusieurs productions à différentes périodes de l'année. La détection sur une production de plusieurs journées (c.f. Figure 4.21a) est affichée sur la Figure 4.24 : les changements brusques sont bien détectés, et il y a peu de détections "faux positives" dans les postes analysés.

Les instants des changements brusques peuvent être précisément détectés sur la VT-restauration globale de l'ensemble d'échantillons, illustrée sur la Figure 4.21b. Formellement, la détection globale est donnée par $\{j | |u_j^* - u_{j-1}^*| > \beta\}$ avec u^* la restauration globale du signal et $\beta > 0$ un seuillage à fixer. Nous utilisons ces détections sur la restauration globale pour évaluer les détections locales proposées par (4.4).

Nous avons affiché sur la Figure 4.25 les fenêtres dans lesquelles les quatre changements brusques du signal sont détectés. Les détections par la méthode locale (4.4) correspondent bien à celles par la restauration globale. La rapidité de détection est entre 60 à 100 points dans cet exemple analysé, ce qui nous permet d'isoler les influences des échantillons avant la rupture et d'avoir une estimation de

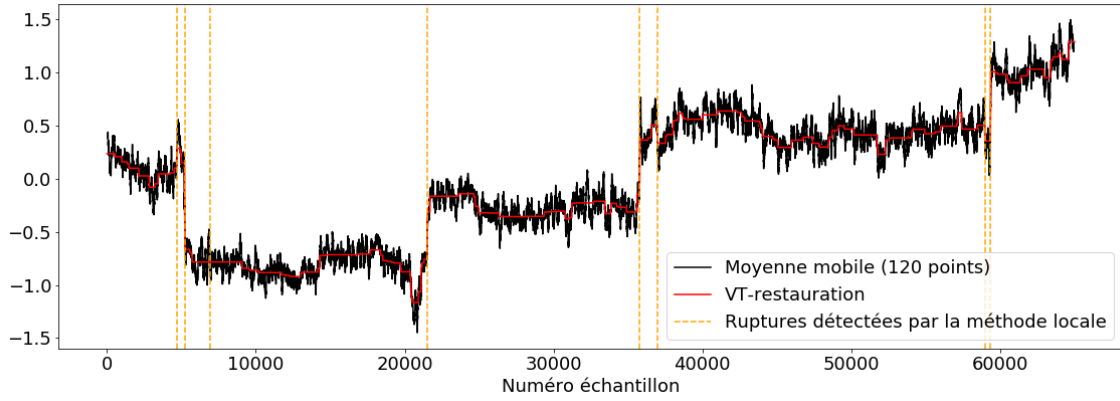


FIGURE 4.24 – Résultats de détection avec la méthode locale robuste. Le signal brut est affiché sur la Figure 4.21a. Couleurs : VT-restauration globale avec notre proposition de λ , moyenne mobile sur une fenêtre de 120 points et ruptures détectées par la méthode locale robuste (4.4) avec $m = 200$, $\alpha = 40$ et $p = 20$. Données : ensemble E3.

moyenne plus rapidement que la moyenne mobile actuellement utilisée dans l'usine. La quantification de la rapidité sur plusieurs productions est en cours.

Les essais sur plusieurs ensembles de données montrent que la méthode proposée est adaptée à la détection en ligne des changements brusques. Cela est d'ailleurs confirmé par l'ingénieur de recherche en charge de l'implémentation du système de mesure.

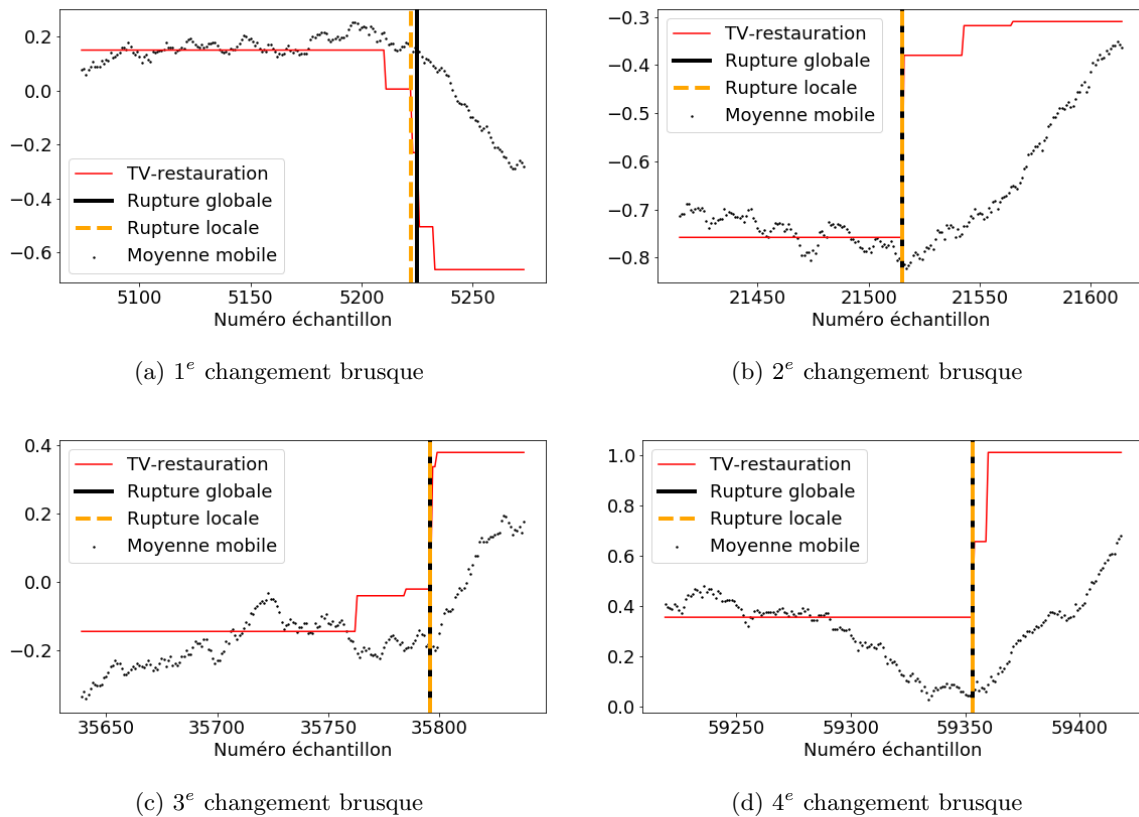


FIGURE 4.25 – Comparaison entre les détections par la méthode locale et celles par la méthode globale. Les quatre fenêtres détectant les changements brusques sont affichées. L'ensemble de détections par la méthode locale est affiché sur la Figure 4.24. Couleurs : ruptures détectées par la méthode locale (4.4), ruptures détectées par la méthode globale, VT-restauration sur l'ensemble d'échantillons avec notre proposition de λ et restauration par la moyenne mobile de 120 points (points noirs). Données : ensemble E3.

4.3.6 Travaux en cours

La quantification de la précision et de la rapidité de la méthode locale proposée, ainsi que leur comparaison avec les méthodes existantes de la détection en ligne des ruptures [34] [1] sont en cours.

Une perspective en cours de développement consiste à proposer une méthode en ligne pour visualiser la variation de signal. Notre première idée a été de combiner la détection des ruptures en ligne et la moyenne mobile sur une grande fenêtre : après avoir détecté une rupture, la moyenne mobile est redémarrée sans considérer les points avant la rupture. La restauration proposée est localement lisse, permettant de visualiser en ligne à la fois les ruptures brusques et les petites variations entre deux ruptures sur les données en live. Notre interlocuteur sur cette application a validé la restauration proposée par la méthode combinée : cette méthode sera implémentée dans l'outil de visualisation de l'usine.

4.3.7 Conclusion

Dans cette section, nous avons développé une méthode en ligne pour détecter les changements brusques dans un signal réel présentant l'état de fonctionnement d'un procédé de Saint-Gobain. Cette méthode est fondée sur un indicateur estimé par la VT-restauration à partir d'une petite fenêtre glissante en considérant la dynamique des signaux restaurés. Les premiers résultats montrent que la méthode proposée a de bonnes précision et rapidité sur les différents ensembles de données testés et surpasse les méthodes utilisées actuellement dans les usines. Notre interlocuteur sur cette application a validé les résultats proposés par notre méthode.

Nous sommes en train de comparer la méthode proposée avec les méthodes existantes de détection en ligne de changements de valeurs moyennes et d'implémenter la méthode proposée dans l'outil de visualisation utilisé dans les usines.

4.4 Application : surveillance de la dérive de la variance du bruit

La dérive de la variance du bruit (σ^2) est un des indices de la défaillance des capteurs. Une détection au stade précoce de la dérive permet de remplacer les capteurs probablement défaillants avant de futures dégradations sur la production.

Dans cette section, nous proposons une méthode en ligne pour détecter la dérive de la variance du bruit utilisant la méthode de restauration proposée. La méthode VT-restauration se fonde sur une hypothèse de σ^2 connu. Afin de l'adapter aux signaux non-stationnaires, une des idées clés est d'appliquer la méthode de restauration dans une fenêtre glissante en supposant la stationnarité locale dans chaque fenêtre. Cette hypothèse est forte mais souvent utilisée pour le traitement des signaux non-stationnaires, comme par exemple la transformation des ondelettes. Afin d'adapter notre méthode aux fenêtres glissantes, deux problèmes doivent être résolus :

- Estimation de la restauration : en appliquant l'Algorithme 4, nous pouvons estimer la restauration $u^*(\lambda_{nours})$ dans une fenêtre en $O(n)$.
- Choix de la taille de fenêtre : nous devons faire un compromis entre l'hypothèse de stationnarité locale et la performance de la restauration. Par exemple, l'influence du nouvel échantillon ne pénètre pas la fenêtre courante dans le cas des grandes fenêtres, mais la stationnarité locale n'est plus correctement fondée. Par la suite, nous supposons que l'influence du nouvel échantillon reste dans la fenêtre courante, mais le choix optimal de la taille de fenêtre reste à déterminer.

4.4.1 Méthode d'estimation de la variance du bruit

Pour une séquence d'échantillons $z = (z_1, \dots, z_n)$, nous prenons une fenêtre glissante de taille m avec $m < n$. Pour chaque fenêtre $w_i^m = [i, i + m - 1]$, nous appliquons notre méthode de restauration sur les séquences $(z_i, \dots, z_{i+m-1})$ et $(\tau_i, \dots, \tau_{i+m-1})$ pour estimer le signal restauré $u^*(\lambda_{nours}) = (u_1^*, \dots, u_m^*)$. Les résidus de restauration pour $j = 1, \dots, m$ sont donnés par :

$$r_j = z_{j+i-1} - u_j^*(\lambda_{nours})$$

La variance des résidus de restauration dans la fenêtre w_i^m peut être estimée par :

$$(\sigma_i^{m*})^2 = \frac{1}{m-1} \sum_{j=1}^m (r_j - \bar{r}_i)^2 \quad (4.5)$$

avec la valeur moyenne des résidus $\bar{r}_i = \frac{1}{m} \sum_{j=1}^m r_j$.

Dans les simulations, les réalisations de bruit sont données par :

$$\hat{\epsilon}_j = z_j - u_{net,j}$$

avec u_{net} le signal original pour $j = 1, \dots, n$.

L'estimation de la variance du bruit dans la fenêtre w_i^m est donnée par :

$$(\hat{\sigma}_i^m)^2 = \frac{1}{m-1} \sum_{j=i}^{i+m-1} (\hat{\epsilon}_j - \bar{\hat{\epsilon}}_i)^2$$

avec $\bar{\hat{\epsilon}}_i = \frac{1}{m} \sum_{j=i}^{i+m-1} \hat{\epsilon}_j$.

Comme $u^*(\lambda_{ours})$ est une bonne restauration de z , $u^*(\lambda_{ours})$ est similaire à u_{net} , ce qui implique que σ_i^{m*} est proche de $\hat{\sigma}_i^m$ dans chaque fenêtre w_i^m . Nous proposons de détecter la dérive de la variance du bruit en utilisant $\sigma^{m*} = (\sigma_1^{m*}, \dots, \sigma_{n-m+1}^{m*})$ avec σ_i^{m*} obtenu par (4.5) dans la fenêtre w_i^m .

Pour estimer la variance du bruit dans une fenêtre de taille m , la complexité est en $O(m)$. La complexité globale pour monitorer la variance d'un signal de taille n est en $O((n-m)m)$.

4.4.2 Analyse de la performance

Pour surveiller la variance du bruit, la méthode proposée devrait précisément estimer la variance du bruit dans le cas idéal, ou capturer la variation des variances du bruit dans le cas plus réaliste. Nous appliquons les critères suivants pour évaluer la performance du monitoring de la variance :

— Biais moyen :

$$\overline{\text{Biais}} = \frac{1}{n-m+1} \sum_{i=1}^{n-m+1} (\hat{\sigma}_i^m - \sigma_i^{m*})$$

— Ratio de la variation de $\hat{\sigma}^m$ expliqué par notre méthode :

$$\text{RVE} = 1 - \frac{\sum_{i=1}^{n-m+1} (\hat{\sigma}_i^m - \overline{\text{biais}} - \sigma_i^{m*})^2}{\sum_{i=1}^{n-m+1} (\hat{\sigma}_i^m - \overline{\hat{\sigma}^m})^2}$$

où $\overline{\hat{\sigma}^m} = \frac{1}{n-m+1} \sum_{i=1}^{n-m+1} (\hat{\sigma}_i^m)$. Il s'agit du score R^2 entre σ^{m*} et $\hat{\sigma}^m$ après avoir ajusté les deux termes à la même moyenne. Le score RVE est entre 0 et 1, et RVE = 1 indique que notre estimation explique parfaitement la variation de σ^{m*} .

Nous avons considéré 5 types de bruit :

1. $\mathcal{N}(0, \sigma_a^2(t))$ avec $\sigma_a(t) = 1 + 0.0005t$
2. $\mathcal{N}(0, \sigma_b^2(t))$ avec $\sigma_b(t) = 1$ pour $t \leq 1000$ et $\sigma_b(t) = 1 + 0.001t$ pour $t > 1000$
3. $\mathcal{U}(-d(t), d(t))$ avec $d(t) = 1 + 0.005t$
4. $\mathcal{N}(0, \sigma_a^2(t)) + \mathcal{U}(-1, 1)$
5. $\mathcal{N}(0, \sigma_c^2(t))$ avec $\sigma_c(t) = 2 + 0.0005t$.

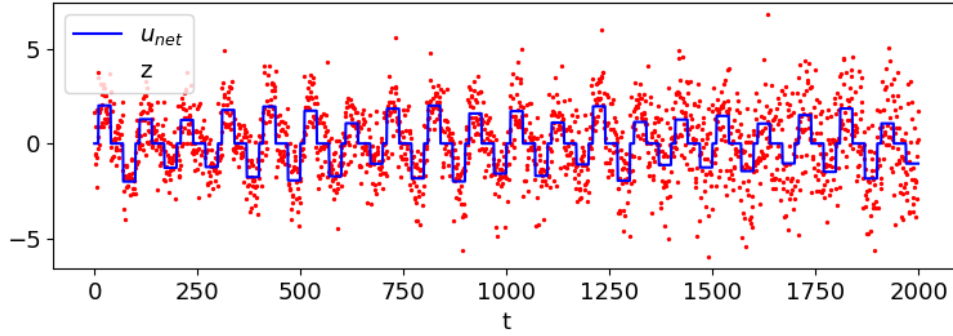
Nous avons fixé le paramètre $\log_{10}(q) = 1$ pour l'estimation de λ dans chaque fenêtre. Nous avons testé la méthode sur un signal simulé u_{net} constant par morceaux, illustré dans la Figure 4.26.

Un exemple de $z = u_{net} + \epsilon$ avec $\epsilon \sim \mathcal{N}(0, \sigma_a^2(t))$ et les estimations des variances du bruit sur une fenêtre glissante de 400 points (w_i^{400} pour tous $i = 1, \dots, 1601$) sont affichées dans la Figures 4.26. La méthode MAD (3.21) et notre méthode proposent toutes les deux une estimation similaire à la variance des réalisations de bruit $\hat{\sigma}^2$ dans chaque fenêtre. Dans cet exemple, notre méthode sous-estime la variance, mais la variation de notre proposition dans chaque fenêtre suit bien celle de $\hat{\sigma}^{400}$. La méthode MAD propose une estimation plus précise, mais l'estimation proposée ne suit pas la variation de $\hat{\sigma}^{400}$. La longueur des séquences non-droite-isolée (c.f. Définition 3.1) dans chaque fenêtre (affichée dans la Figure 4.26c) sont toutes inférieures à la taille de fenêtre $m = 400$. Cela indique que l'influence du nouvel échantillon reste dans la fenêtre actuelle, ce qui valide notre supposition de départ.

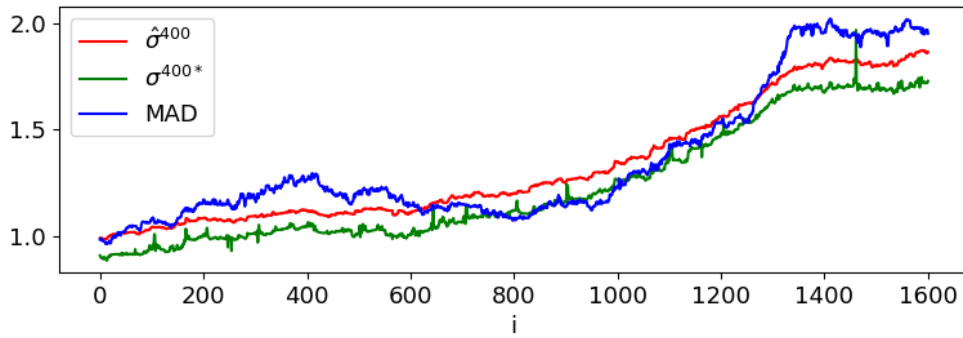
Nous avons effectué 100 simulations pour chaque type de bruit considéré. Les résultats des quatre premiers types (faibles bruits) sont montrés dans la Figure 4.27. Pour la plupart des simulations, notre méthode a un score RVE $> 0,95$ et dépasse l'estimateur MAD. Un grand score RVE indique que la variation proposée par notre méthode s'adapte bien à celle des valeurs réelles, nous permettant de surveiller la variation de la variance du bruit pour les signaux réels collectés de différents procédés. Cependant, les résultats de score $\overline{\text{Biais}}$ montrent que notre méthode sous-estime toujours la variance du bruit, et que le biais dépend du type de bruit. Nous ne pouvons pas calculer un décalage universel pour débiaiser nos estimations pour un cas général.

Nous avons aussi testé notre méthode dans les cas de fort bruit, et les résultats de 100 simulations sont montrés dans la Figure 4.28. Notre méthode fournit encore une performance acceptable avec RVE $> 0,85$ pour la plupart des simulations, tandis que la méthode MAD n'explique pas la variation de $\hat{\sigma}^{400}$.

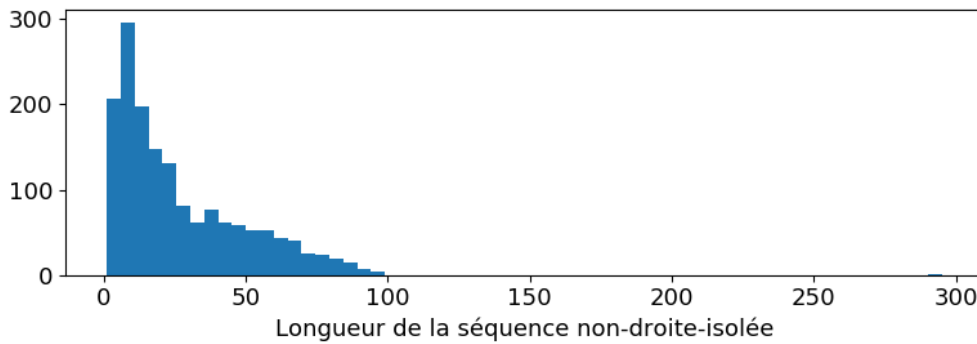
La taille de fenêtre joue un rôle prépondérant dans la qualité de restauration et aussi dans la surveillance de la variance du bruit. Une petite fenêtre permet de capturer la variation locale de la variance du bruit, mais le nouvel échantillon (i.e. $(i+m)^e$ point) pourrait encore avoir une forte influence aux échantillons passés (i.e. $i-1, i-2, \dots$) et limiter la performance de restauration pour ces points passés. Nous avons testé $m \in \{200; 400; 600\}$ avec $\epsilon \sim \mathcal{N}(0, \sigma_a^2(t))$, et les résultats sont affichés dans la Figure 4.29. Une fenêtre plus longue fournit une meilleure performance de la surveillance. Cependant, la variation locale de σ^2 serait négligée, et l'hypothèse de la stationnarité locale n'est plus bien fondée.



(a) Exemple d'un signal simulé : $z = u_{net} + \epsilon$ avec $\epsilon \sim \mathcal{N}(0, \sigma_a^2(t))$. Couleurs : signal original u_{net} et échantillons simulés z .

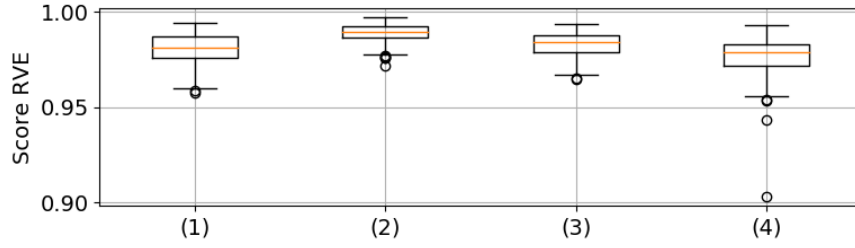


(b) Estimations de la variance du bruit dans des fenêtres glissantes de 400 points. Couleurs : variance des réalisations de bruit, notre estimation avec $\log_{10}(q) = 1$ et estimation MAD.

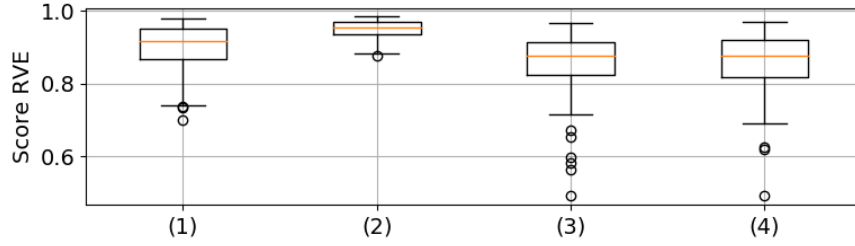


(c) Histogramme de la longueur des séquences non-droite-isolées

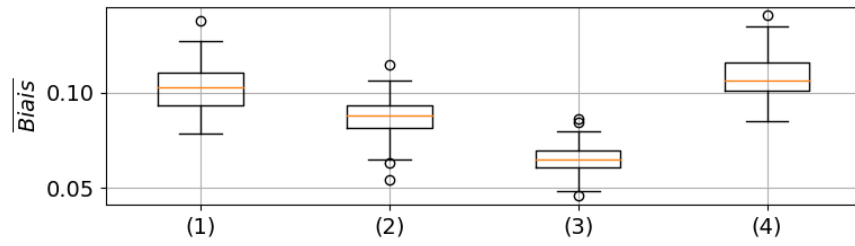
FIGURE 4.26 – Un exemple de l'estimation de la variance du bruit avec une fenêtre de 400 points.



(a) Score RVE de notre méthode sous les différents types de bruit



(b) Score RVE de la méthode MAD sous les différents types de bruit



(c) Score biais de notre méthode sous les différents types de bruit

FIGURE 4.27 – Résultats de 100 simulations sous les différents types de bruit. (1) : $\mathcal{N}(0, \sigma_a^2(t))$, (2) : $\mathcal{N}(0, \sigma_b^2(t))$, (3) : $\mathcal{U}(-d(t), d(t))$ et (4) : $\mathcal{N}(0, \sigma_a^2(t)) + \mathcal{U}(-1, 1)$.

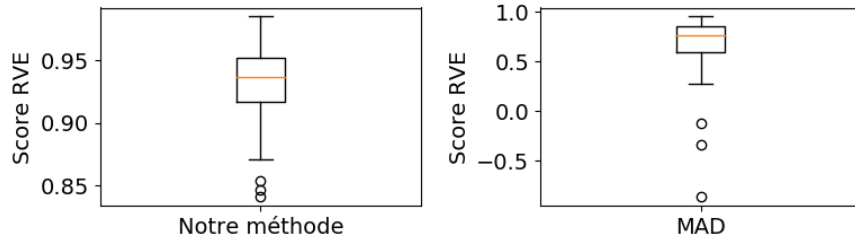


FIGURE 4.28 – Score RVE de 100 simulations de notre méthode et MAD avec $\epsilon \sim \mathcal{N}(0, \sigma_c^2(t))$ et $\sigma_c(t) = 2 + 0.0005t$.

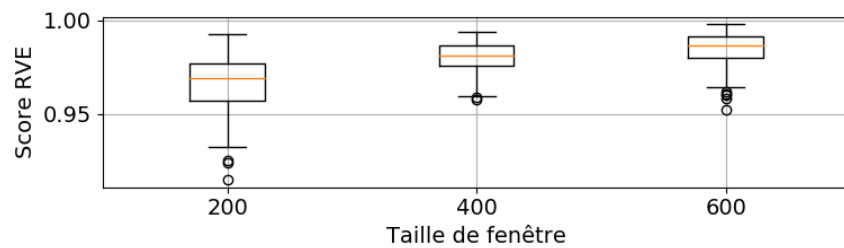


FIGURE 4.29 – Score RVE de 100 simulations de notre méthode avec les différentes tailles de fenêtre (m). $\epsilon \sim \mathcal{N}(0, \sigma_a^2(t))$.

4.4.3 Conclusion

Nous avons proposé une méthode pour estimer la variance du bruit fondée sur la VT-restauration dans une fenêtre glissante. La méthode ne fournit pas une estimation précise de la variance, mais notre estimation suit bien la variation de la variance des réalisations de bruit avec une erreur d'estimation qui dépend du type de bruit et de la forme du signal original.

Une des applications de notre méthode serait de détecter la dérive de la variance du bruit. La variance des résidus de restauration pourrait rester stable (i.e $\sigma_i^{m*} = \sigma$) pendant le bon fonctionnement du capteur, et la dérive de la variance du bruit pourrait être détectée par un seuillage prédéfini sur σ_i^{m*} : e.g $\sigma_i^{m*} > 1.2\sigma$ pour plusieurs fenêtres successives.

L'application de la méthode proposée sur les données réelles est en cours d'investigation, et le choix de la taille de fenêtre fait partie des prochaines étapes.

Chapitre 5

Conclusions générales et perspectives

5.1 Conclusions

De nos jours, la prise de décision automatique dans un procédé industriel complexe, faisant partie de ce qu'on appelle "l'industrie 4.0", est une évolution cruciale pour que les procédés industriels puissent rester compétitifs. En ce moment, il n'existe pas une méthode universelle applicable aux différents procédés existants, et il est nécessaire d'effectuer une analyse approfondie du fonctionnement du procédé considéré et des données enregistrées pour comprendre les problématiques spécifiques à celui-ci. Dans cette thèse, nous nous sommes attelés à établir une méthodologie pour analyser des procédés industriels variés ainsi que leurs données, en vue de la proposition de lois de contrôle de ceux-ci.

Pour cela, nous avons travaillé sur une grande quantité de données procédé issues de différentes usines utilisant le même procédé Saint-Gobain (*Process 1*). Une étape préliminaire a été d'appliquer des méthodes statistiques classiques pour comprendre la structure des données. Nous avons d'abord analysé les paramètres procédé à l'aide d'outils statistiques (l'analyse en composantes principales, la classification non-supervisée, etc) et cherché à relier les informations obtenues lors de ces analyses au fonctionnement du procédé. Nous avons ensuite tenté d'analyser une mesure de qualité du produit final en fonction des paramètres opératoires utilisés lors de sa fabrication, l'objectif final étant d'obtenir un modèle prédictif de cette mesure de qualité en fonction des paramètres procédé remontés sur le *Process 1*. L'hypothèse nulle de la modélisation linéaire a été rejetée. Un ensemble restreint de paramètres procédé contribuant à l'explication de la mesure de qualité a été sélectionné par les méthodes statistiques utilisées, ce qui a été d'ailleurs validé auprès de l'expert procédé. Nous avons à partir de ce constat étudié des alternatives non-linéaires (par exemple les modèles additifs généralisés) qui ont amélioré la performance prédictive par rapport aux modèles linéaires simples. Les résultats de nos études statistiques montrent que les données industrielles, possédant souvent un fort bruit de mesure, sont problématiques dans le cadre de la modélisation, et que des analyses plus fines sur la relation non-linéaire entre la mesure de qualité et les paramètres procédé doivent être effectuées.

Concernant le *Process 1*, outre une meilleure connaissance de celui-ci et des interdépendances entre les paramètres suivis jusqu'ici, l'une des conclusions de nos travaux a été la nécessité d'implémenter de nouveaux capteurs à certains endroits bien spécifiques afin d'approfondir les analyses de la mesure de qualité. Cette partie a fait l'objet d'une communication dans un congrès national [26] et de deux rapports internes chez Saint-Gobain Recherche.

La modélisation plus fine de la non-linéarité est le sujet de la deuxième partie de cette thèse. Afin d'analyser un paramètre ayant potentiellement une influence sur la qualité du produit, nous avons développé une approche variationnelle pour reconstituer les informations potentiellement intéressantes à partir d'échantillons bruités de mesures de capteurs. Les résultats concrets de cette partie sont des algorithmes applicables en temps réel et en ligne de la restauration d'un signal 1D par la régularisation de variation totale (VT) avec la détermination automatique de l'hyper-paramètre λ pondérant le poids entre l'attachement aux données et la régularisation imposée. Ce travail a donné lieu à un article qui a été soumis dans un journal international avec comité de lecteur [27].

La méthode de restauration en temps réel a été ensuite appliquée à plusieurs procédés de Saint-Gobain :

- Pour le *Process 1*, nous avons analysé un signal clé d’après l’intuition des opérateurs et de l’ingénieur procédé impliqués avec la méthode de restauration proposée. Les informations reconstituées n’améliorent pas a priori les modèles prédictifs de la mesure de qualité. Cependant, des analyses effectuées nous indiquent qu’il est probable que certains facteurs de premier ordre ne soient pas entièrement remontés dans la base de données. Ceci a donné lieu à des actions d’implémentation de nouveaux capteurs sur les lignes de production.
- Nous avons analysé une mesure de qualité 2D enregistrée en continu par une série de 24 capteurs dans un deuxième procédé de Saint-Gobain (*Process 2*). La restauration des quantiles de mesures de qualité nous a permis d’identifier une variation synchronisée des quantiles avec une période caractéristique au cours de la production, ce qui a un impact non-négligeable sur la qualité du produit et le rendement du procédé. Notre méthode de restauration a montré ici son intérêt dans la détection automatique de motifs récurrents, potentiellement déclenchés par certains paramètres opératoires. L’analyse des causes de ces motifs est actuellement en cours au sein des équipes de Saint-Gobain.
- Sur un troisième procédé de Saint-Gobain, nous avons proposé une méthode en ligne de détection des changements brusques d’un signal bruité issu d’un nouvel instrument de mesure en cours de déploiement. La méthode proposée a une meilleure performance que celle utilisée actuellement en usine. Les résultats sont validés par l’ingénieur de recherche en charge de l’implémentation du nouveau système de mesure, et la méthode proposée sera implémentée dans un outil de visualisation utilisé par les opérateurs dans les prochains mois.
- Une autre application envisagée de la méthode de débruitage développée dans cette thèse consiste à détecter en ligne la dérive de la variance du bruit de mesure afin de faciliter la maintenance préventive des lignes de production. Ce travail a donné lieu à un article qui a été soumis à une conférence internationale [28].

Les deux axes développés sont complémentaires : les méthodes statistiques permettent d’identifier les paramètres clés liés au fonctionnement de procédés et de caractériser la relation multi-variée entre une mesure de qualité et des paramètres identifiés ; tandis que les approches de restauration reconstituent les informations potentiellement intéressantes noyées dans les bruits de mesure, ce qui pourrait améliorer ensuite les modèles prédictifs des mesures de qualité, aider à la compréhension du fonctionnement de procédés, ou encore faciliter la mise en place voire robustifier des lois de contrôle utilisant ces signaux débruités. Les contributions de ces deux axes permettent de proposer une méthodologie pour construire des modèles prédictifs des mesures de qualité à partir des paramètres opératoires enregistrés, ce qui fait partie des travaux préliminaires pour la proposition de lois de régulation de procédé efficaces.

Il faut remarquer que l’estimation des hyper-paramètres est un problème crucial dans l’analyse des données. Nous avons développé une approche déterministe dans le cadre de la restauration par la régularisation de variation totale. L’estimation des hyper-paramètres reste un problème ouvert pour les autres méthodes statistiques et variationnelles.

5.2 Perspectives

Ces travaux de thèse ouvrent plusieurs perspectives de recherche et d’application, liées à la modélisation statistique d’un procédé industriel et à la restauration automatique d’un signal bruité.

Une première perspective consiste à chercher d’autres facteurs clés pour l’optimisation du fonctionnement du *Process 1* analysé dans la première partie du manuscrit, notamment parmi les informations collectées par les nouveaux capteurs qui sont en cours d’implémentation. D’autre part, une dépendance entre les produits successivement réalisés a été observée lors des différentes analyses, et les méthodes de type auto-régressif pourraient améliorer les modèles prédictifs.

Plusieurs autres perspectives concernent l'application de la méthode de restauration en temps réel. La restauration localement lisse proposée par la VT-restauration est particulièrement adaptée aux signaux collectés sur différents procédés industriels, car possédant souvent des changements brusques en raison de la modification des réglages de machine. D'un point de vue applicatif, la méthode proposée pourrait être appliquée sur différents procédés automatisés ayant toutefois encore certains réglages manuels proposés par les opérateurs : par exemple la fabrication du verre plat par Saint-Gobain Glass, de pare-brises par Saint-Gobain Sekurit ou encore de laine de verre par Saint-Gobain Isover. Le faible coût en temps de calcul montre un intérêt particulier pour la surveillance du fonctionnement de capteurs en temps réel. Un des travaux en cours consiste à appliquer cette méthode aux signaux non-stationnaires à l'aide de fenêtres glissantes. Le choix de la taille de fenêtre reste une question importante pour garantir le bon fonctionnement de la méthode et le bien fondé des hypothèses imposées par celle-ci.

Une autre perspective serait d'adapter notre méthode d'estimation de l'hyper-paramètre à des données en deux dimensions (comme par exemple une image, une mesure de qualité spatialisée, etc). Deux problèmes sont à résoudre pour la 2D VT-restauration :

- Estimer les hyper-paramètres afin de restaurer correctement les données 2D : la question clé est d'estimer efficacement $g(\lambda)$, le nombre d'extremums locaux de la restauration en fonction de λ , pour des données 2D. Les auteurs de [40] ont proposé une méthode pour estimer Λ (la liste de λ qui évoque un changement de la topologie de restauration). Cependant, l'estimation efficace de $g(\lambda)$ à partir de Λ n'est pas un problème simple car la structure de voisinage est plus compliquée en 2D. Cela fait partie de nos travaux en cours, notamment concernant des algorithmes d'estimation en temps réel. Par ailleurs, les méthodes stochastiques de type MCMC [38], [42] pour estimer les hyper-paramètres font aussi partie de nos pistes de recherche.
- Adapter la méthode de restauration en ligne à des données 2D : les méthodes existantes [7], [15] et [4] estiment la restauration d'une image d'une manière hors ligne en considérant l'ensemble de pixels. L'idée de la méthode en ligne est de mettre à jour l'image restaurée lorsqu'une nouvelle série d'échantillons est mesurée. Elle est efficace pour traiter les mesures de qualité en 2D enregistrées en continu, comme par exemple celle du *Process 2*, présentée dans la Section 4.2.

Plusieurs problèmes théoriques issus de la méthode de restauration proposée font aussi partie de nos perspectives, comme par exemple l'analyse théorique de notre proposition de l'hyper-paramètre à partir de la fonction $g(\lambda)$ et le choix optimal de point de découpe $\hat{\lambda}$ pour garantir la bonne performance des méthodes en ligne proposées.

A plus long terme, il serait également intéressant de se pencher sur l'estimation des hyper-paramètres de pondération pour les autres méthodes statistiques et variationnelles. Le choix des hyper-paramètres joue un rôle crucial sur la performance de méthodes appliquées, et reste un des obstacles pour les applications réelles.

Durant cette thèse, nous avons pu faire un pas vers l'industrie 4.0, qui est un domaine de recherche attractif pour le secteur industriel de nos jours, en proposant une méthodologie pour la modélisation statistique d'un procédé industriel. La perspective à plus long terme consiste à générer les lois de contrôle efficaces en utilisant les modèles statistiques construits à partir d'une grande quantité de données procédé provenant de différents procédés industriels, afin d'optimiser ceux-ci que ce soit en termes de consommation de matières premières, d'énergie ou en termes d'émissions de CO₂.

Références

- [1] Samaneh Aminikhanghahi and Diane J Cook. A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2) :339–367, 2017. [70](#)
- [2] Antoniadis Anestis. *Régression non linéaire et applications / Anestis Antoniadis, Jacques Ber-ruyer, René Carmona*. Collection Economie et statistiques avancées série Ecole nationale de la statistique et de l’administration économique et du Centre d’études des programmes économiques. Economica, Paris. [6](#), [9](#)
- [3] Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4 :40–79, 2010. [33](#)
- [4] Jean-François Aujol. Some first-order algorithms for total variation based image restoration. *Journal of Mathematical Imaging and Vision*, 34(3) :307–327, 2009. [79](#)
- [5] José M Bioucas-Dias. Bayesian wavelet-based image deconvolution : a gem algorithm exploiting a class of heavy-tailed priors. *IEEE Transactions on Image Processing*, 15(4) :937–951, 2006. [15](#)
- [6] Leo Breiman. Random forests. *Machine learning*, 45(1) :5–32, 2001. [6](#)
- [7] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2) :89–97, 2004. [15](#), [79](#)
- [8] Antonin Chambolle and Pierre-Louis Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2) :167–188, 1997. [15](#)
- [9] L. Condat. A direct algorithm for 1-d total variation denoising. *IEEE Signal Processing Letters*, 20(11) :1054–1057, Nov 2013. [15](#)
- [10] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3) :273–297, 1995. [6](#)
- [11] N. Cressie. *Statistics for Spatial Data*. Wiley Series in Probability and Statistics. Wiley, 1993. [9](#)
- [12] Susmita Datta and Somnath Datta. Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics*, 19(4) :459–466, 2003. [6](#)
- [13] David L Donoho and Iain M Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432) :1200–1224, 1995. [22](#)
- [14] David L Donoho and Jain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3) :425–455, 1994. [15](#), [36](#), [37](#)
- [15] Jalal M. Fadili and Gabriel Peyré. Total Variation Projection with First Order Schemes. In IEEE, editor, *ICIP’09*, pages 1325–1328, Cairo, Egypt, November 2009. IEEE. [79](#)
- [16] Carlo Gaetan and Xavier Guyon. *Spatial Statistics and Modeling*. Springer Series in Statistics. Springer Science+Business Media, LLC, New York, NY, 2010. [9](#)

- [17] Zaid Harchaoui and Céline Lévy-Leduc. Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association*, 105(492) :1480–1493, 2010. [15](#)
- [18] SayedMasoud Hashemi, Soosan Beheshti, Richard SC Cobbold, and Narinder Paul. Adaptive updating of regularization parameters. *Signal Processing*, 113 :228–233, 2015. [15](#)
- [19] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning : data mining, inference, and prediction*. Springer Science & Business Media, 2009. [6](#), [9](#)
- [20] Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*, volume 43. CRC press, 1990. [6](#)
- [21] François Husson, Sébastien Lê, and Jérôme Pagès. *Exploratory multivariate analysis by example using R*. CRC press, 2017. [5](#)
- [22] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013. [6](#)
- [23] Angela Kempe. *Statistical analysis of discontinuous phenomena with Potts functionals*. PhD thesis, lmu, 2004. [17](#)
- [24] Vladimir Kolmogorov, Thomas Pock, and Michal Rolinek. Total variation on a tree. *SIAM Journal on Imaging Sciences*, 9(2) :605–636, 2016. [15](#)
- [25] Zhanhao Liu. Modélisation statistique d’un procédé de centrifugation, Note technique interne Saint-Gobain Recherche Paris - O2M/Datalab - Confidentiel, 2021. [5](#)
- [26] Zhanhao Liu, Marion Perrodin, Thomas Chambrion, and Radu S. Stoica. Modélisation statistique d’un procédé de centrifugation. In *Journées de Statistique*, Nancy, France, 2019. [6](#), [77](#)
- [27] Zhanhao Liu, Marion Perrodin, Thomas Chambrion, and Radu S. Stoica. Revisit 1D Total Variation restoration problem with new real-time algorithms for signal and hyper-parameter estimations. December 2020. [77](#)
- [28] Zhanhao Liu, Marion Perrodin, Thomas Chambrion, and Radu S. Stoica. Windowed total variation denoising and noise variance monitoring. January 2021. [78](#)
- [29] Cécile Louchet and Lionel Moisan. Total Variation as a local filter. *SIAM Journal on Imaging Sciences*, 4(2) :651–694, 2011. [15](#)
- [30] Vladimir Alekseevich Morozov. *Methods for solving incorrectly posed problems*. Springer Science & Business Media, 2012. [15](#)
- [31] Francesco Ortelletti, Sara van de Geer, et al. On the total variation regularized estimator over a class of tree graphs. *Electronic Journal of Statistics*, 12(2) :4517–4570, 2018. [15](#)
- [32] Ilya Pollak, Alan S Willsky, and Yan Huang. Nonlinear evolution equations as fast and exact solvers of estimation problems. *IEEE Transactions on Signal Processing*, 53(2) :484–498, 2005. [15](#), [16](#), [17](#), [39](#)
- [33] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D : nonlinear phenomena*, 60(1-4) :259–268, 1992. [15](#)
- [34] Nassim Sahki, Anne Gégout-Petit, and Sophie Wantz-Mézières. Performance study of change-point detection thresholds for cumulative sum statistic in a sequential context. *Quality and Reliability Engineering International*, 36(8) :2699–2719, 2020. [70](#)
- [35] Sylvain Sardy and Hatéf Monajemi. Threshold selection for total variation denoising. *arXiv preprint arXiv :1605.01438*, 2016. [33](#), [34](#)

- [36] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8) :1627–1639, 1964. [14](#)
- [37] Charles M Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981. [33](#)
- [38] Radu S Stoica, Anne Philippe, Pablo Gregori, and Jorge Mateu. Abc shadow algorithm : a tool for statistical analysis of spatial patterns. *Statistics and computing*, 27(5) :1225–1238, 2017. [79](#)
- [39] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society : Series B (Methodological)*, 58(1) :267–288, 1996. [6](#)
- [40] Ryan J. Tibshirani and Jonathan Taylor. The solution path of the generalized lasso. *The Annals of Statistics*, 39(3) :1335–1371, Jun 2011. [15](#), [17](#), [21](#), [33](#), [39](#), [79](#)
- [41] You-Wei Wen and Raymond H Chan. Parameter selection for total-variation-based image restoration using discrepancy principle. *IEEE Transactions on Image Processing*, 21(4) :1770–1781, 2011. [15](#)
- [42] Gerhard Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods : A Mathematical Introduction (Stochastic Modelling and Applied Probability)*. Springer-Verlag, Berlin, Heidelberg, 2006. [14](#), [17](#), [79](#)