Outils automatiques d'évaluation de la qualité des données pour le suivi en continu de la qualité des eaux usées

Automatic data quality assessment tools for continuous monitoring of wastewater quality

Mémoire

Romain Philippe

Sous la direction de :

Peter A. Vanrolleghem, directeur de recherche

Résumé

Aujourd'hui, la surveillance et le contrôle de la qualité des eaux usées (réseaux d'égouts, stations de récupération des ressources de l'eau – StaRRE, rivières) utilisent plusieurs capteurs installés en ligne. Une bonne stratégie de surveillance devrait être fiable et fournir une bonne qualité de données. L'utilisation des méthodes actuelles de détection de fautes a montré que des problèmes de colmatage conduisent à une perte de données comprise entre 10 et 60 %. Aider les utilisateurs à comprendre, analyser et traiter les fautes détectées (capteurs colmatés, fautes de calibration, installations et maintenances sous-optimales) permettrait de réduire le pourcentage de perte de données et d'atteindre de bonnes données sur la qualité des eaux usées. Dans ce travail de maîtrise, nous proposons deux outils modulaires complets permettant d'obtenir des informations exploitables à partir des données brutes (c'est-à-dire pour la détection des erreurs de capteurs, le contrôle ou la surveillance de processus). Ces outils ont été appliqués à des séries chronologiques des projets pil*EAU*te, bord*EAU*x et kam*EAU* collectés dans différents réseaux d'égouts et les StaRRE. Ces méthodes ont été rendues limpides dans leur applicabilité avec la rédaction de « Standard Operating Procedures (SOP) » facilitant leur utilisation. Aussi, elles sont modulaires avec la construction de blocs de fonctions, tels qu'une boîte à outils. La première méthode est un outil univarié composé de deux étapes principales: le filtrage des données (détection des valeurs aberrantes et lissage) et la détection des fautes. La deuxième méthode est un outil utilisant l'Analyse en Composantes Principales (ACP) également composée de deux étapes: Développement du modèle ACP et détection des fautes par l'ACP. Finalement, dans les cas d'étude, le traitement des données a conduit à une perte minimale de données variant de 0.1-12 %.

Abstract

Nowadays, in the wastewater field (sewers, water resource recovery facilities - WRRFs, rivers), the monitoring and control of wastewater quality is performed with several on-line sensors. However, a good monitoring strategy should be reliable and provide good data quality. The current fault detection methods have shown that problems such as fouling lead to 10-60 % of the data being discarded. However, helping users in understanding, analysing and processing detected faults (sensors clogging, faulty calibration, suboptimal installation and maintenance) will allow reducing the percentage of data loss and reaching good data on wastewater quality. In this Master thesis, we propose two full workflows allowing the collection of raw data and their transformation into actionable information (i.e. for sensor fault detection, control or process monitoring). The two full modular frameworks were applied to time series data coming from the pil*EAU*te, bord*EAU*x and kam*EAU* projects collected in sewers and WRRFs. These methods have been made more easily applicable by writing Standard Operation Procedures (SOPs) on the use of these methods. In addition, the Matlab scripts are written in a modular way by building different function blocks that are compiled in a toolbox. The first method is a univariate tool composed of two main steps: Data filtering (outlier detection and smoothing) and fault detection. The second method is a multivariate tool using Principal Component Analysis, also composed of two steps: (i) the development of the PCA model and (ii) the fault detection by the PCA. Finally, for the three aforementioned projects, data treatment has led to only 0.1-12 % of the data being discarded.

Table des matières

RÉSUMÉ II

ABSTRACT III

TABLE DES MATIÈRES IV

LISTE DES FIGURES VII

LISTE DES TABLEAUX XI

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES XIII

REMERCIEMENTS XV

INTRODUCTION 1

CHAPITRE 1 REVUE DE LITTÉRATURE SUR LE SUIVI DE LA QUALITÉ DES EAUX USÉES 2

1.1. EAUX USÉES ET TRAITEMENT 2
1.2. SUIVI DE LA QUALITÉ DE L'EAU USÉE 3
1.2.1. Variables de la qualité des eaux usées 3
1.2.2. Capteurs : types de capteurs et méthodes 3
1.2.3. Exemple de suivi usuel dans une StaRRE à l'aide de capteurs 7
1.3. FAUTES ET PROBLÈMES DES CAPTEURS 8
1.4. MÉTHODES DE TRAITEMENT DES DONNÉES 10
1.4.1. Extraction basique d'informations 10
1.4.2. Extraction avancée d'information 11
1.5. VALIDATION DES CAPTEURS 13
1.5.1. Diagramme de contrôle 13
1.5.1.1. Tests préliminaires 14
1.5.1.2. Analyse des fautes 14
1.5.2. Redondance des capteurs 15

CHAPITRE 2 PROBLÉMATIQUE ET OBJECTIFS 17

2.1. PROBLÉMATIQUE 17
2.2. OBJECTIFS 17

CHAPITRE 3 MATÉRIEL ET MÉTHODES 19

3.1. SITES D'ÉTUDE 19
3.1.1. pilEAUte 19
3.1.1.1. Système de contrôles (Station monEAU et SCADA/PLC) 23
3.1.1.2. Station monEAU 23
3.1.1.3. SCADA/PLC 24
3.1.1.1. Capteurs et emplacements 26
3.1.2. kamEAU 28
3.1.2.1. Capteurs et emplacements 30
3.1.3. bordEAUx 30
3.1.3.1. Capteurs et emplacements 32
3.2. MÉTHODES DE TRAITEMENT DE DONNÉES 33
3.2.1. Méthode univariée 33

3.2.1.1. Filtrage des données 33
a. Détection des données aberrantes 33
b. Lissage des données sans les données aberrantes 35
3.2.1.2. Détection des fautes 36
c. Calcul d'indicateurs de défaillances et leurs limites 36
d. Obtention des données acceptées et rejetées 39
3.2.2. Méthode multivariée 40
3.2.2.1. Développement du modèle ACP 40
a. Choix de données normales 40
b. Normalisation des données 41
c. Création du modèle ACP 41
d. Calcul des tests statistiques Q et T^2 et leurs limites 43
3.2.2.2. Détection des fautes 45
a. Choix de données à traiter 45
b. Normalisation des données à traiter 46
c. Projection sur le modèle ACP 46
d. Évaluation des tests T^2 et Q 46

CHAPITRE 4 RÉSULTATS ET DISCUSSION SUR LA QUALITÉ DES DONNÉES 47

4.1. RÉACTION DES CAPTEURS FACE À DEUX PROBLÈMES COURANTS 47
4.1.1. Nettoyage des capteurs 47
4.1.1.1. pilEAUte 47
4.1.1.2. kamEAU 50
4.1.1.3. bordEAUx 51
4.1.2. Réaction des capteurs à une forte charge d'un composant 51
4.1.3. Conclusion 53
4.2. MÉTHODES SIMPLES ET MODULAIRES DE TRAITEMENT DES DONNÉES 54
4.2.1. Méthode univariée 54
4.2.1.1. Structure des données 56
4.2.1.2. Présentation du script général avec ces fonctions 56
4.2.1.1. Applications 59
a. pil*EAU*te 60
- Étape 1: Détection des données aberrantes 60
- Étape 2: Lissage des données 63
- Étape 3: Détection des fautes 64
- Étape 4: Données traitées 65
- Compléments 68

b. kam*EAU* 70
- Entrée de la station 70
- Sortie de la station 76

c. Bord*EAU*x 83
- Réseau d'égout 83
- Entrée de la station 90

4.2.1.2. Conclusion 96
4.2.2. Méthode multivariée 96
4.2.2.1. Présentation du script et de ses fonctions 96
4.2.2.2. Applications 98
a. Sortie décanteur primaire 98
b. Bioréacteurs 103
c. Méthode multivariée en ligne 108

4.2.2.3. Conclusion 114
4.3. VALIDATION DES CAPTEURS : REDONDANCE DES CAPTEURS 115
4.3.1. Méthodes *115*
4.3.2. Applications *115*
4.3.2.1. pilEAUte 115
4.3.2.2. kamEAU 117
4.3.2.3. bordEAUx 119
4.3.3. Compléments *120*
4.3.4. Conclusion *122*

CONCLUSION ET PERSPECTIVES **123**

MAINTENANCE DES CAPTEURS 123
TRAITEMENT DES DONNÉES 124
VALIDATION DES DONNÉES PAR REDONDANCE 125

BIBLIOGRAPHIE **126**

ANNEXES **131**

A. Évaluation de l'effet du nettoyage d'après Plana (2015) 131
B. SOP méthode univariée 132
C. SOP méthode multivariée 159

Liste des figures

Figure 1. Exemple de contrôle et de suivi usuel dans une StaRRE (Ingildsen and Olsson, 2016) 7
Figure 2. Fautes majeures pour les données provenant d'un capteur (a) Biais, (b) Dérive (c) Défaillance (d) Dégradation de la précision (Yoo et al., 2008) 8
Figure 3. Exemple de défaillance au niveau d'un capteur (Plana, 2015) 9
Figure 4. Exemple de colmatage d'un capteur d'oxygène dissous (Tao et al., 2013) 9
Figure 5. Modèle d'identification des données aberrantes (Krajewski and Krajewski, 1989) 11
Figure 6. Diagramme de validation des données de capteurs (Thomann et al., 2002) 13
Figure 7. Diagramme de contrôle (Montgomery, 2009) 14
Figure 8. Exemple de diagramme de contrôle (Montgomery, 1996) 15
Figure 9. Suivi en continue du pH, la conductivité, la turbidité et la température avec une redondance des capteurs (Alferes et al., 2013b) 16
Figure 10. L'usine pilEAUte au sein du pavillon Adrien-Pouliot, Université Laval (Québec, Canada) 19
Figure 11. Schéma d'installation de l'usine pilEAUte de traitement des eaux usées 20
Figure 12. Configuration des réacteurs biologiques 22
Figure 13. La station monEAU avec (a) l'extérieur de la station (b) l'intérieur de la station 23
Figure 14. L'interface graphique d'utilisateur du SCADA de l'usine pilEAUte 24
Figure 15. Supervision du débit d'air comprimé au sein des réacteurs biologiques (pilEAUte : R230/240/250 ; copilEAUte : R330/340/350) 25
Figure 16. Exemple de programmation API avec la surveillance du niveau du bassin tampon avec l'implantation d'une alarme sur le niveau 25
Figure 17. Exemple d'entrée dans un API 26
Figure 18. Exemple de sortie dans un API 26
Figure 19. Surveillance de quelques variables (débits, niveau d'un bassin, conductivité, température) de l'usine pilEAUte à l'aide de SCADA 27
Figure 20. Étang aéré à Grandes-Piles (a) sans le système KAMAK (b) avec le système KAMAK (c) Vue du dessus du système (Patry et al., 2018) 29
Figure 21. Système KAMAK avec (a) le média BIONEST, (b) cellule de 3 mètres de hauteur comportant le média (c) les zones des réacteurs biologiques RX1 et RX2, (d) les zones de décantation CL1, CL2 et CL3 (d) le sens d'écoulement au sein du système (Patry et al., 2018) 29
Figure 22. L'ensemble des bassins versants de la communauté urbaine de Bordeaux 31
Figure 23. Bassin versant "Clos de Hilde" (Ledergerber et al., 2018) 31
Figure 24. Exemple théorique de détection des données aberrantes et leurs remplacement par la prédiction (Alferes et al., 2012) 35
Figure 25. Exemple théorique de le filtrage de données (Alferes et al., 2012) 36
Figure 26. Exemple de faute détectée par l'indicateur de défaillances « signe run-test » 37
Figure 27. Exemple de faute détectée par l'indicateur de défaillances « pente » 38
Figure 28. Exemple de faute détectée par l'indicateur de défaillances « déviation standard » 38
Figure 29. Exemple de faute détectée par l'indicateur de défaillances « plage » 39
Figure 30. Exemple d'un diagramme « eigenvalue scree plot » d'après Alferes et al. (2012) 42
Figure 31. Exemple de réorganisation de l'information dans ses composantes (Alferes et al., 2013b) 43
Figure 32. Représentation graphique du test T^2 (Montgomery, 2009) 43
Figure 33. Représentation graphique du test Q (Montgomery, 2009) 44
Figure 34. Nettoyage illustrant un biais entre les données de NH_4-N (ammo::lyser, affluent pilEAUte) captées avant et après le nettoyage (ligne pointillée verte) 48
Figure 35. Dérive continuelle des données de DCO soluble (spectro::lyser, affluent du pilEAUte) après un mauvais nettoyage (deuxième trait pointillé) 48
Figure 36. Diagramme général de la séquence de traitement de données pour en assurer la qualité. 49
Figure 37. Nettoyage pro-actif d'un capteur d'oxygène dissous, du réacteur aéré pilEAUte (rectangle en pointillé vert) 50

Figure 38. Colmatage rapide après le nettoyage du capteur (trait vert en pointillés) dans la série de données des MES (spectro::lyser, affluent KAMAK) 50
Figure 39. Dérive des données de MES (spectro::lyser, réseau d'égout Bordeaux) quelques temps après les nettoyages du capteur (lignes vertes pointillées) 51
Figure 40. Suivi en continue de l'ammonium avec deux capteurs Varion à quatre fortes injections de NH_4 dans un réacteur aéré du pilEAUte 52
Figure 41. Suivi en continue des nitrates avec deux capteurs spectro::lyser à quatre fortes injections de NO_3 dans un réacteur aéré du pilEAUte 53
Figure 42. Diagramme général des deux phases de traitement des données 55
Figure 43. Exemple de structure de données importées dans MATLAB a) Structure générale b) Format de la colonne channel (nom de la variable, unité de la variable) c) Format de la colonne values (temps en format MATLAB, valeurs) 56
Figure 44. Structure finale après le traitement par la méthode univariée 59
Figure 45. Diagramme général de la méthode univariée avec les outils graphiques 59
Figure 46. Données brutes d'oxygène dissous mesuré dans un réacteur aéré de boues activées du pilEAUte durant une période de 5 mois 60
Figure 47. Détection des données aberrantes pour la variable oxygène dissous mesurée au sein d'un bioréacteur du pilEAUte durant une période de cinq mois 61
Figure 48. Détection de données aberrantes pour la variable d'oxygène dissous mesurée dans un bioréacteur du pilEAUte correspondant à un nettoyage hebdomadaire du capteur (détail de la Figure 47) 61
Figure 49. Détection des données aberrantes pour la variable oxygène dissous mesurée dans un bioréacteur du pilEAUte durant une période de cinq mois avec le paramètre « param.nb_s » égale à 2 62
Figure 50. Détection de données aberrantes de la variable d'oxygène dissous mesurée dans un bioréacteur du pilEAUte correspondant à un nettoyage hebdomadaire du capteur avec le paramètre « param.nb_s » égale à 2 (détail de la Figure 49) 62
Figure 51. Données lissées d'oxygène dissous mesuré dans un bioréacteur du pilEAUte sur une période de cinq mois 63
Figure 52. Agrandissement sur une période de dix jours des données filtrées d'oxygène dissous mesuré dans un bioréacteur du pilEAUte (détail de la Figure 51) 63
Figure 53. Agrandissement sur une période de dix jours des données filtrées d'oxygène dissous mesuré dans un réacteur aéré du pilEAUte avec l'augmentation du paramètre « param.h_smoother » à 200 64
Figure 54. Détermination des indicateurs de défaillances en comparaison avec leurs limites pour la détection des fautes sur la période de cinq mois de l'oxygène dissous mesuré dans un bioréacteur du pilEAUte 65
Figure 55. Données brutes et traitées de la variable d'oxygène dissous mesuré dans un réacteur aéré du pilEAUte a) Données brutes b) Données traitées 66
Figure 56. Agrandissement sur la détection d'une défaillance complète et deux colmatages du capteur d'oxygène dissous mesuré dans un bioréacteur du pilEAUte (détail de la Figure 55) a) Données brutes b) Données traitées 67
Figure 57.Données brutes et traitées de la variable DCO mesurée à l'entrée d'un étang aéré KAMAK pour une période d'un an a) Données brutes b) Données traitées 72
Figure 58. Agrandissement sur les données brutes et traitées de la variable DCO mesurée à l'entrée d'un étang aéré KAMAK sur une période de neuf jours (détail de la Figure 57) a) Données brutes b) Données traitées 73
Figure 59. Données brutes et traitées de température mesurée à l'entrée d'un étang aéré KAMAK durant la période de trois mois a) Données brutes b) Données traitées 75
Figure 60. Agrandissement sur les données de température mesurée à l'entrée d'un étang aéré KAMAK sur une période de douze jours (détail de la Figure 59) a) Données brutes b) Données traitées 76
Figure 61. Données brutes et traitées de la variable nitrate mesurée à l'effluent d'un étang aéré KAMAK sur une période d'un an a) Données brutes b) Données traitées 78

Figure 62. Agrandissement sur les données brutes et traitées de la variable nitrate mesurées à l'effluent d'un étang aéré KAMAK sur une période de sept jours (détail de la Figure 61) a) Données brutes b) Données traitées 79
Figure 63. Données brutes et traitées de l'oxygène dissous à l'effluent d'un étang aéré KAMAK sur une période d'un an a) Données brutes b) Données traitées 81
Figure 64. Agrandissement sur les données brutes et traitées de l'oxygène dissous à l'effluent d'un étang aéré KAMAK sur une période de neuf jours (détail de la Figure 63) a) Données brutes b) Données traitées 82
Figure 65. Données brutes et traitées de la variable MES mesurée dans le réseau d'égout sur une période de deux mois à Bordeaux a) Données brutes b) Données traitées 85
Figure 66. Agrandissement sur les données brutes et traitées des MES mesurées dans le réseau d'égout sur une période d'onze jours à Bordeaux (détail de la Figure 65) a) Données brutes b) Données traitées 86
Figure 67. Données brutes et traitées de température mesurée dans le réseau d'égout sur une période de deux mois à Bordeaux a) Données brutes b) Données traitées 88
Figure 68. Agrandissement des données brutes et traitées de température mesurée dans le réseau d'égout à Bordeaux (détail de la Figure 67) a) Données brutes b) Données traitées 89
Figure 69. Données brutes et traitées de la DCO mesurée à l'entrée de la StaRRE sur une période de trois mois à Bordeaux a) Données brutes b) Données traitées 91
Figure 70. Agrandissement sur les données brutes et traitées de la DCO mesurée à l'entrée de la StaRRE de Bordeaux sur une période de neuf jours (détail de la Figure 69) a) Données brutes b) Données traitées 92
Figure 71. Données brutes et traitées de la variable pH mesurée à l'entrée de la StaRRE sur une période de trois mois a) Données brutes b) Données traitées 94
Figure 72. Agrandissement sur les données brutes et traitées de pH mesuré à l'entrée de la StaRRE à Bordeaux (Détail de la Figure 71) a) Données brutes b) Données traitées 95
Figure 73. Données prétraitées par la méthode univariée de DCO totale et soluble et MES mesurées en sortie du décanteur primaire au pilEAUte 99
Figure 74. Données prétraitées par la méthode univariée de NH_4, K, température et pH mesurés en sortie du décanteur primaire au pilEAUte 99
Figure 75. Données prétraitées par la méthode univariée normales, sélectionnées de DCO totale et soluble et MES pour la construction du modèle ACP 100
Figure 76. Données prétraitées par la méthode univariée normales, sélectionnées de NH_4, K, température et pH pour la construction du modèle ACP 100
Figure 77. Pourcentages des valeurs propres pour les composantes principales 101
Figure 78. Test Q et T^2 pour les données normales 101
Figure 79. Test Q et T^2 pour la série de nouvelles données entre février et avril 2018 obtenues à l'affluent du décanteur primaire du pilEAUte 102
Figure 80. Données prétraitées par la méthode univariée de MES dans l'effluent du décanteur primaire du pilEAUte dont une faute a été détectée par la méthode ACP 102
Figure 81. Données prétraitées par la méthode univariée de DCO soluble dans l'effluent du décanteur primaire du pilEAUte dont une faute de dérive a été détectée par la méthode ACP 103
Figure 82. Données des MES et OD mesurées dans le bioréacteur pilEAUte 104
Figure 83. Données prétraitées par la méthode univariée des débits d'air et NH_4 mesurées dans le bioréacteur pilEAUte et à l'effluent du décanteur primaire 104
Figure 84. Données normales sélectionnées de MES et d'OD pour la construction du modèle ACP 105
Figure 85. Données normales sélectionnées des débits d'air et de NH_4 pour la construction du modèle ACP 105
Figure 86. Pourcentages des valeurs propres pour les composantes principales du pilEAUte 106
Figure 87. Test Q et T^2 pour les données normales du bioréacteur pilEAUte 106
Figure 88. Test Q et T^2 pour les données à traiter bioréacteur pilEAUte 107
Figure 89. Données prétraitées par la méthode univariée du NH_4 à l'effluent du décanteur primaire dont une faute a été détectée par la méthode ACP 108

Figure 90. Données prétraitées par la méthode univariée d'oxygène dissous dans le bioréacteur pilEAUte dont des fautes ont été détectées par la méthode ACP 108
Figure 91. Données brutes de NH_4-N pour l'illustration de la méthode ACP mesurées dans un réacteur aéré du pilEAUte..... 109
Figure 92. Données brutes de NO_3-N pour l'illustration de la méthode ACP mesurées dans un réacteur aéré du pilEAUte..... 110
Figure 93. Données normales de NH_4-N pour le développement du modèle ACP mesurées dans un réacteur aéré du pilEAUte..... 110
Figure 94. Données normales de NO_3-N pour le développement du modèle ACP mesurées dans un réacteur aéré du pilEAUte..... 111
Figure 95. Pourcentages des valeurs propres en fonction des composantes principales réacteur aéré du pilEAUte 111
Figure 96. Tests Q et T^2 et leurs limites pour la série normale réacteur aéré du pilEAUte 112
Figure 97. Tests Q et T^2 pour la série de nouvelles données réacteur aéré du pilEAUte 113
Figure 98. Données brutes de NH_4 mesurées dans un réacteur aéré du pilEAUte pour trois capteurs..... 113
Figure 99. Données brutes de NO_3 mesurées dans un réacteur aéré du pilEAUte pour trois capteurs..... 114
Figure 100. Diagramme de la validation des données par la redondance de capteurs 115
Figure 101. Redondance de la variable MES mesurée dans un bioréacteur du pilEAUte 116
Figure 102. Redondance de la variable OD mesurée dans un bioréacteur du pilEAUte 116
Figure 103. Redondance de la mesure de la température à l'entrée du KAMAK 118
Figure 104. Redondance de la mesure du pH à l'entrée du KAMAK 118
Figure 105. Redondance de la mesure des MES à l'entrée du KAMAK..... 119
Figure 106. Redondance de la variable MES à l'entrée de la StaRRE à Bordeaux..... 120
Figure 107. Validation des données en ligne de MES dans un réacteur à boues activées du pilEAUte avec des données de laboratoire 121
Figure 108. Agrandissement de la Figure 107 sur la validation des données en ligne de MES dans un réacteur à boues activées du pilEAUte avec les données de laboratoire (détail de la Figure 107)... 121

Liste des tableaux

Tableau 1. Variables suivies au sein des StaRRE d'après Vanrolleghem et Lee (2003) 3
Tableau 2. Considérations dans l'implantation de capteurs d'après Lynggaard-Jensen (1999) 4
Tableau 3. Exemples de type de capteurs et leurs méthodes de mesure 5
Tableau 4. Méthodes pour la détection des fautes d'après Corominas et al., (2018) 10
Tableau 5. Capteurs au sein de l'usine pilEAUte 28
Tableau 6. Capteurs au sein du KAMAK connectés au système de surveillance monEAU 30
Tableau 7. Capteurs connectés au système de surveillance monEAU au sein des deux points de mesures (Noutary et Clos de Hilde) à Bordeaux 32
Tableau 8. Capteurs utilisés pour l'expérience et leurs variables mesurées dans un réacteur aéré du pilEAUte 52
Tableau 9. Format du fichier «.csv» 56
Tableau 10. L'ensemble des paramètres utilisés dans la méthode univariée pour chaque étape 57
Tableau 11. Quelques exemples de valeurs de paramètres pour la méthode univariée pour les capteurs du pilEAUte 68
Tableau 12. Valeurs des paramètres modifiés pour la méthode de filtrage des données pour la variable DCO mesurée à l'entrée du KAMAK 71
Tableau 13. Limites des indicateurs de défaillances pour la détection de fautes dans la série de données de la variable DCO mesurée à l'entrée du KAMAK 71
Tableau 14. Pourcentage des données aberrantes et rejetées pour la variable DCO mesurée à l'entrée d'un étang aéré KAMAK 73
Tableau 15. Valeurs des paramètres modifiées pour la méthode de filtrage des données pour la variable température mesurée à l'entrée d'un étang aéré KAMAK 73
Tableau 16. Limites des indicateurs de défaillances pour la détection de fautes dans la série de données de la variable température mesurée à l'entrée du KAMAK 74
Tableau 17. Pourcentage des données aberrantes et rejetées pour la variable température mesurée à l'entrée d'un étang aéré KAMAK 76
Tableau 18. Paramètre modifié pour la méthode de filtrage des données pour la variable nitrate mesurée à la sortie d'un étang aéré KAMAK 77
Tableau 19. Limites des indicateurs de défaillances pour la détection de fautes pour la série de données de la variable nitrate mesurée à la sortie d'un étang aéré KAMAK 77
Tableau 20. Pourcentage des données aberrantes et rejetées pour la variable nitrate mesurée à l'effluent d'un étang aéré KAMAK 79
Tableau 21. Valeurs modifiées des paramètres modifiés pour la variable d'oxygène dissous à l'effluent d'un étang aéré KAMAK 80
Tableau 22. Limites des indicateurs de défaillances pour la détection de fautes dans la série de données de la variable d'oxygène dissous mesurée à la sortie d'un étang aéré KAMAK 80
Tableau 23. Pourcentage des données aberrantes et rejetées pour l'oxygène dissous à l'effluent d'un étang aéré KAMAK 82
Tableau 24. Nouvelles valeurs des paramètres modifiés pour la méthode de filtrage des données pour la variable MES dans le réseau d'égout à Bordeaux 83
Tableau 25. Limites des indicateurs de défaillances pour la détection de fautes dans la série de données de la variable MES mesurée dans le réseau d'égout à Bordeaux 84
Tableau 26. Pourcentage des données aberrantes et rejetées pour les MES mesurées dans le réseau d'égout à Bordeaux 86
Tableau 27. Nouvelles valeurs des paramètres modifiés pour la méthode de filtrage des données pour la variable température mesurée en réseau d'égout à Bordeaux 87
Tableau 28. Limites des indicateurs de défaillances pour la détection de fautes dans la série de données de la variable température mesurée dans le réseau d'égout à Bordeaux 87
Tableau 29. Pourcentage des données aberrantes et rejetées pour la température mesurée dans le réseau d'égout à Bordeaux 90

Tableau 30. Valeurs des paramètres modifiés pour la méthode de filtrage des données pour la variable DCO mesurée à l'entrée de la StaRRE à Bordeaux 90
Tableau 31. Limites des indicateurs de défaillances pour la détection de fautes dans la série de données de la variable DCO mesurée à l'entrée de la StaRRE à Bordeaux 90
Tableau 32. Pourcentage des données aberrantes et rejetées pour la DCO mesurée à l'entrée de la StaRRE de Bordeaux 92
Tableau 33. Nouvelles valeurs des paramètres modifiés pour la méthode de filtrage de données pour la variable pH mesurée à l'entrée de la StaRRE à Bordeaux 93
Tableau 34. Limites des indicateurs de défaillances pour la détection de fautes dans la série de données de la variable pH mesurée à l'entrée de la StaRRE à Bordeaux 93
Tableau 35. Pourcentage des données aberrantes et rejetées pour la variable pH mesurée à l'entrée de la StaRRE de Bordeaux 95
Tableau 36. Paramètres utilisés dans la méthode multivariée et leur valeur par défaut 97
Tableau 37. Redondance de variables mesurées sur deux points de mesures dans le pilEAUte 115
Tableau 38. Redondance de variables mesurées sur deux points de mesures dans le KAMAK 117

Liste des abréviations, sigles et acronymes

ACP Analyse en Composantes Principales

AGV Acide Gras Volatils

AIP Automate Industriel Programmable

AQT Analyse Qualitative des Tendances

CH_4 Méthane

CO_2 Dioxyde de Carbone

COT Carbone Organique Total

CP Composante Principale

DBO_{ct} Demande Biochimique en Oxygène à court terme

DCO Demande Chimique en Oxygène

DEL Diode Électroluminescente

ESI Électrode à ions-sélectifs

H_2 Hydrogène

H_2S Hydrogène sulfuré

MES Matières en Suspension

NH_4^+ Ammonium

NO_3^- Nitrate

NO_X Oxyde d'azote

PLC Programmable Logical Controller

PO_4^{3-} Phosphate

OD Oxygène Dissous

SOP Standard Operating Procedure

StaRRE Station de Récupération des Ressources de l'Eau

STEP Stations de Traitement Des Eaux Usées

UV-vis Ultra-violet visible

« Le sport va chercher la peur pour la dominer, la fatigue pour en triompher, la difficulté pour la vaincre » Pierre de Coubertin

Remerciements

Je tiens tout d'abord à remercier mon professeur de maîtrise Peter Vanrolleghem, professeur-chercheur au département de génie civil et de génie des eaux de l'Université Laval pour son encadrement dynamique, son expertise, sa qualité pédagogique, sa passion pour la science et sa disponibilité durant ces deux années. Sans lui tout cela n'aurait pas pu débuter ni aboutir. Enfin, je le remercie aussi pour son temps passé sur la correction de mon mémoire.

Merci à Cyril Garneau, postdoctorant, pour son encadrement dynamique et sa grande disponibilité, ses réponses à mes nombreuses questions, pour son soutien sur le logiciel MATLAB, et aussi ses nombreux commentaires constructifs lors de la rédaction de ce mémoire.

Merci à Elena Torfs, postdoctorante, pour son aide et ses réponses durant le début de ma maîtrise sur le projet pil*EAU*te.

Je remercie aussi l'ensemble de l'équipe pil*EAU*te composé de Gamze, Maryam, Jean-David, Feiyi et les stagiaires pour leur soutien et leur aide.

Je remercie aussi toute l'équipe model*EAU* de m'avoir accueilli à nouveau avec une grande courtoisie et sympathie.

Enfin, je voudrais aussi remercier mes parents, mes frères, ma belle-sœur, mon neveu et l'ensemble de ma famille pour leur soutien indéfectible et leur profonde affection, et de m'avoir remis sur la bonne voie durant certaines périodes difficiles pendant ces deux années même si la distance qui nous sépare est importante.

Introduction

Une recrudescence des changements climatiques ainsi qu'une pollution élevée des milieux récepteurs (lacs, rivières…) est actuellement observée. Ceci place donc l'eau au centre des débats et particulièrement sa gestion qui est importante pour éviter des pénuries futures ou des contaminations diverses qui auraient un impact direct sur l'Homme. Cette gestion inclue le traitement des eaux usées, effectué par des stations de traitement des eaux usées (STEP).

Récemment, les eaux usées ont été définies comme une ressource et non plus comme un déchet, puisqu'elles permettent notamment la production d'énergie, d'engrais pour le domaine de l'agriculture ou bien d'eau potable. Ce changement de paradigme a induit une redéfinition des STEP en Stations de Récupération des Ressources de l'eau (StaRRE). Cette transition a aussi conduit à une meilleure gestion des ressources et des coûts dans les stations. Pour effectuer ceci, le contrôle des processus est devenu plus strict en mettant en place des limites de qualité plus sévères (Vanrolleghem and Vaneeckhaute, 2014).

Pour contrôler ces processus tels que l'aération dans des bioréacteurs, l'utilisation de capteurs est primordiale. Cependant, l'utilisation de ces derniers mène à une collecte importante de données qui sont transférées dans des bases de données. Par exemple, un capteur utilisé pendant trois mois et demi et prenant une mesure toutes les 5 secondes génèrera environ 2 millions de données.

Cependant, les eaux usées peuvent rendre l'utilisation de ces capteurs difficile. Ceci est dû aux importantes quantités de particules et autres matières présentes. Ainsi, plusieurs auteurs indiquent des pourcentages de pertes de données comprises entre 5 et 60 % (Alferes et al., 2013a). Par exemple, au sein des expériences de la Chaire de recherche du Canada model*EAU*, Alferes et al. (2013a) rapportent une perte de données variant de 8 à 14 %.

Ces pertes de données résultent de bruit de fond, de données aberrantes ou manquantes. Yoo et al. (2008) avaient défini quatre façons dont la qualité des séries de données peut être affectée : les biais, les dérives, les défaillances complètes du capteur et la dégradation de la précision.

Afin d'éviter ces problèmes, des outils automatiques ont été mis en place pour détecter, isoler et diagnostiquer les pannes (anomalies ou fautes). Cependant, l'utilisation de ces outils nécessite une expertise qui doit être inscrite sur une documentation, ce qui n'est pas le cas à ce jour.

C'est pour répondre à ce problème que s'inscrit cette maîtrise. Deux grands thèmes seront abordés par la suite : l'utilisation des outils automatiques d'évaluation de la qualité des données et le suivi en continu de la qualité des eaux usées. Ceci permettra, *in fine*, d'obtenir des outils précédemment développés clairs, documentées et modulaires afin de faciliter les utilisateurs futurs.

Chapitre 1 Revue de littérature sur le suivi de la qualité des eaux usées

1.1. Eaux usées et traitement

Aujourd'hui, la croissance continue de la population mondiale et les dérèglements climatiques, ont transformé l'eau potable en denrée épuisable. En effet, en 2017, le comité national français du Fonds des Nations unies estimait que 2,1 milliard de personnes n'avaient pas accès à l'eau potable et 4,4 milliards de personnes à un système d'assainissement convenable (Unicef France, 2017). Une des manières d'améliorer l'accès à l'eau potable consiste à traiter les eaux usées.

Le traitement des eaux usées a été mise en place depuis longtemps pour limiter les risques potentiels pour la santé humaine dans les agglomérations urbaines (Henze et al., 2008). Des recherches archéologiques ont identifié la présence de systèmes de traitement et d'assainissements dans les civilisations antérieures. Ces dernières ont montré, par exemple, dans des anciennes ruines de Moenjodaro (2600 -1700 AC) (ville localisée au Sud du Pakistan), que des salles de bains et des toilettes étaient directement connectées à des systèmes de drainage dans les rues ou à des réseaux d'égouts (Checkley and Checkley, 2008; Gray, 1940). A l'ère moderne industrielles, les premières recherches et la mise en place de systèmes de traitement ont commencé aux États-Unis et au Royaume-Uni aux 19 ème siècle (Henze et al., 2008). Le tout premier système à avoir vu le jour a été le filtre biologique ou lit bactérien en 1893 à Salford près de Manchester au Royaume-Uni. Il a par la suite été implanté aux États-Unis, dès 1901, à Madison dans le Wisconsin. Puis entre 1902 et 1920, cette technologie a été installée dans plusieurs autres villes pour le traitement des eaux usées. Au même moment, le traitement par boue activée, inventée en 1913, a eu des difficultés à s'implanter dues aux faibles investissements financiers comparé à celui mis en place pour le filtre biologique (Henze et al., 2008). Les deux technologies citées précédemment permettent d'éliminer des eaux usées, les matières organiques, azotées et phosphorées.

En parallèle, durant la seconde partie du 20ème siècle, un nouveau problème a vu le jour dû au rejet des eaux usées traitées dans les eaux de surface réceptrices (Lac, rivière, cours d'eau…). En effet, le phénomène d'eutrophisation apparaît (Henze et al., 2008). Ce processus se définit par la prolifération des algues et autres plantes aquatiques découlant de la fertilisation de la matière azotée et phosphorée effectuée par les bactéries dans les eaux de surfaces (Ramalho, 2012). Actuellement pour limiter ce phénomène, beaucoup de recherches sont mises en place pour éliminer ces matières (Henze et al., 2008). Mais, depuis plusieurs années, avec l'augmentation du secteur industriel, d'autres polluants ont vu le jour tels que les métaux lourds (arsenic, plomb, etc…) et les micropolluants (pesticides, médicaments, produits de combustion, etc…). Ces derniers s'accumulent dans les écosystèmes naturels (cours d'eau, plans d'eau, etc..) et ont par la suite des effets toxiques sur les écosystèmes et ont un potentiel de perturbateur endocrinien sur l'être humain (Henze et al., 2008). Afin de solutionner ce problème, des recherches dans le suivie du trai-

tement des eaux usées ont été entreprises. Cependant, les solutions proposées qui reposent sur l'acquisition et le traitement d'un nombre important de données produisent beaucoup de données. Par conséquent, la recherche dans le domaine du suivie de la qualité des eaux usées basée sur des données fiables est essentielle et permettra de minimiser les impacts des polluants sur les milieux récepteurs et leurs faunes.

1.2. Suivi de la qualité de l'eau usée

Le suivi de la qualité de l'eau usée commence par une bonne compréhension des variables à suivre et des capteurs utilisés. Dans cette section, ces deux aspects seront développés ainsi que deux exemples de suivi usuel dans des StaRRE.

1.2.1. Variables de la qualité des eaux usées

Les variables de la qualité des eaux usées sont multiples et augmentent au fur et à mesure que les méthodes de détection s'améliorent. Yoo et al. (2003) définissent les variables couramment surveillées dans le domaine des eaux usées au sein des StaRRE présentés dans le Tableau 1. Ces diverses variables sont classées en trois catégories : Mesures physiques, mesures physico-chimiques et mesures (bio)-chimiques.

Tableau 1. Variables suivies au sein des StaRRE d'après Vanrolleghem et Lee (2003)

Type de mesures		
Physique	**Physico-chimique**	**(Bio)-chimique**
Température	pH	Respirométrie
Pression	Conductivité	Toxicité
Niveau de l'eau	Oxygène dissous	DBO_{ct} (Demande Biologique en Oxygène à court terme)
Débit	Fluorescence	
MES (Matière en suspension)	Redox	DCO (Demande Chimique en Oxygène)
Niveau de boues	NH_4^+ (Ammonium)	COT (Carbone Organique Total)
Volume de boues	NO_3^- (Nitrate)	NH_4^+ (Ammonium)
Vitesse de sédimentation	Gaz (CH_4 (Méthane), H_2S (Hydrogène sulfuré), H_2 (Hydrogène), CO_2 (Dioxyde de carbone))	NO_3^- (Nitrate)
Morphologie de la boue		NOX (Oxyde d'azote)
Calorimétrie		PO_4^{3-} (Phosphate)
Absorption UV (Ultraviolet)		Bicarbonate
		Alcalinité
		AGV (Acide Gras Volatils)

1.2.2. Capteurs : types de capteurs et méthodes

Généralement, l'implantation de capteurs (appareils de mesure) dans le secteur de l'eau usée n'est pas aussi développée que dans d'autres secteurs industriels tels que l'agroalimentaire ou le biomédical en raison d'un environnement néfaste pour ces appareils. Cependant, avant l'installation de capteurs, plusieurs

considérations nécessitent d'être prises en compte (Bourgeois et al., 2001; Lynggaard-Jensen, 1999). Lynggaard-Jensen (1999) explique les différents aspects à prendre en compte tels que la méthode de mesure, la fiabilité, la précision du capteur (Tableau 2). Bonastre et al. (2005) reprennent les mêmes considérations concernant la localisation du capteur pour leur analyse.

Tableau 2. Considérations dans l'implantation de capteurs d'après Lynggaard-Jensen (1999)

Propriétés	**Exemples**
Localisation du capteur	In-situ, en ligne, hors-ligne,…
Principe d'échantillonnage	Échantillonnage externe ou non
Principe de filtration	Filtrage ou non
Principe de traitement de l'échantillon	Continu, ponctuel
Méthode de mesure	Photométrie, colorimétrie, titrimétrie,…
Type de mesure	Mono-paramètre, multi-paramètre
Consommables	Utilisation de réactifs chimiques, filtres, etc…
Intervalle de mesure	Minute, heure,...

De plus, la connaissance des méthodes de mesure est un aspect important. Ceci a pour but d'assurer une efficacité de traitement des eaux et aussi de prévenir les anomalies (Bourgeois et al., 2001).
Chaque variable a une méthode de mesure différente, le Tableau 3 propose quelques exemples de capteurs et leurs méthodes de mesure.

Tableau 3. Exemples de type de capteurs et leurs méthodes de mesure

Variable suivie	***Modèle de capteurs***	***Méthode de mesure***
pH	pHD sc Digital Differential (HACH*) 	**Électrochimique :** le pH est le logarithme négatif de l'activité ionique de l'hydrogène et une mesure de l'acidité ou l'alcalinité d'une solution. Deux électrodes sont installées au sein du capteur, une électrode en verre et une électrode de référence. L'électrode en verre agit en tant que transducteur en convertissant l'énergie chimique (l'activité ionique de l'hydrogène) en potentiel électrique (mesurée en millivolts). La réaction est équilibrée et le circuit électrique est complété par le flux d'ions depuis la solution de référence à la solution testée (Hach, 2006a).
Oxygène dissous	LDO (Luminescent Dissolved Oxygen) (HACH*) 	**Luminescence :** le principe de la luminescence est basé sur une méthode optique mesurant la durée de la luminescence après une impulsion d'excitation. Plus généralement, le capteur est recouvert d'un cap luminescent. La lumière bleue d'une DEL (Diode Électroluminescente) excite les composantes luminescentes du cap. Lorsque ces dernières reviennent à un état d'équilibre, une lumière rouge est émise par fluorescence et détectée par une photodiode. Le temps d'émission de la lumière rouge est mesuré. Ce dernier est inversement proportionnel à la concentration en oxygène dissous (Hach, 2006b).
Ions (NH_4^+, NO_3^+)	IQ SensorNet VARION (Xylem*) ammo::lyser (s::can*) 	**EIS (Électrodes à Ions-sélectifs) :** une électrode à ions-sélectifs est un capteur qui convertit l'activité d'un ion spécifique dissous dans une solution en un potentiel électrique mesurable. La tension dépend du logarithme de l'activité ionique se basant sur l'équation de Nernst : $$E = E_0 + \left(\frac{RT}{nF}\right) ln \frac{a_{ox}^x}{a_{red}^x}$$ *Équation 1* Avec : E_0 : le potentiel standard [Volt] T : la température absolue [Kelvin] R : la constante des gaz parfaits : 8.3144621 J.mol-1.K^{-1} a : l'activité chimique de l'oxydant et du réducteur F : la constante de Faraday : 96 485 C.mol^{-1} = 1 F n : le nombre d'électrons transférés dans la demi -réaction Au sein de chaque capteur, des membranes à ions spécifiques y sont installées. Les potentiels mesurés sont convertis en concentration à l'aide de l'équation de Nernst (s::can, 2007; Xylem, 2012).

Matière organique (DCO)	*spectro::lyser (s::can*)* 	**Absorbance UV-vis :** le capteur mesure l'absorbance dans des longueurs d'ondes ultra-violettes et visibles (190 à 720 nm). Le principe de mesure repose sur une émission d'un faisceau lumineux diffusé par une lampe sur le milieu. Ce dernier est capté par un détecteur situé à 180° de l'émetteur. Chaque molécule d'un composant dissout dans le milieu absorbe des radiations à une certaine longueur d'onde. La concentration des substances dans le milieu augmente l'intensité d'absorption. Ainsi, la longueur d'onde mesurée a pour but de déterminer la concentration des divers composants dans le milieu *(s::can, 2011)*.
Conductivité	3700sc Digital Conductivity Sensor (HACH*) 	**Induction :** le principe de mesure est basé sur l'induction électromagnétique. Cette dernière est mesurée à l'aide de deux bobines. Un courant alternatif est envoyé dans la première bobine (référence) qui induit un courant dans l'échantillon d'eau. Ce courant est par la suite mesuré par la deuxième bobine afin de créer un courant proportionnel à la conductivité (Hach, 2008).
Turbidité	Solitax sc (HACH*) 	**Diffusion :** le principe de mesure est basé sur la diffusion de la lumière sur des particules. Le capteur émet une lumière infrarouge et capte la lumière diffusée latéralement à un angle de 90°, la néphélométrie (Hach, 2004).
Hauteur d'eau	Débitmètre « Sigma 950 » (capteurs bulle à bulle ou capteur à vitesse Doppler) (HACH*) 	**Hauteur d'eau :** Le principe de mesure est basé sur une mesure d'une hauteur d'eau en poussant une série de bulles d'air continue à travers un tube de plastique immergé au fond de la colonne d'eau. La pression dans ce dernier est proportionnelle à la hauteur du liquide au point où l'extrémité du tube est mise en place. Le capteur mesure cette pression et la convertie en hauteur d'eau. La hauteur d'eau est convertie en débit à l'aide de la loi hauteur-débit en connaissant l'hydraulique de l'ouvrage. Sinon, un capteur à vitesse Doppler permet de mesurer la vitesse de l'écoulement. Des ondes sonores à hautes fréquences sont transmises dans le milieu et sont réfléchies par les particules en mouvement. La vitesse des particules est alors déterminée à partir de la fréquence de retour. Le débit (Q) est ainsi déterminé à partir de l'équation regroupant la vitesse des particules (v_{par}) et la hauteur d'eau (h_{eau}) (Hach, 2014) : $Q = H_{eau}\ x\ v_{par}$ *Équation 2*

* compagnies commercialisant les capteurs

1.2.3. Exemple de suivi usuel dans une StaRRE à l'aide de capteurs

La question pouvant se poser est « pourquoi utiliser des capteurs pour contrôler et suivre le traitement dans une StaRRE ». Afin de répondre à cette question, prenons l'exemple d'une StaRRE qui traite les eaux usées d'une usine de lait en poudre. Cette station s'est dotée d'un module de bioréacteur à membrane (MBM). Le système est robuste et facilement installable. L'avantage de ce dernier est aussi le rejet d'un effluent d'une bonne qualité dans le milieu récepteur (rivière, lac) servant par exemple à l'irrigation, ou même pour la consommation humaine (Ingildsen and Olsson, 2016).

Dans ce système, une surveillance en temps réel est mise en place afin de rendre le système le plus autonome possible et d'avoir le moins de maintenances manuels de la part d'un opérateur. Dans le même ordre idée, l'oxygène dissous est mesuré dans les bassins aérobies et la teneur en NH_4 et en NO_3 dans l'effluent. A partir de la concentration en NH_4, les opérateurs peuvent contrôler et ajuster l'oxygène fourni dans les bassins aérobies. La Figure 1 montre un exemple de changement de la valeur désirée (set point) d'oxygène dans les bassins aérobies de 1,5 à 2 mg/L en réaction à une forte augmentation de la teneur en NH_4 dans l'effluent qui met en danger le respect de la norme. Ce changement en oxygène dissous se réalise instantanément et l'effet sur la teneur en NH_4 commence à être observable après 2 heures.

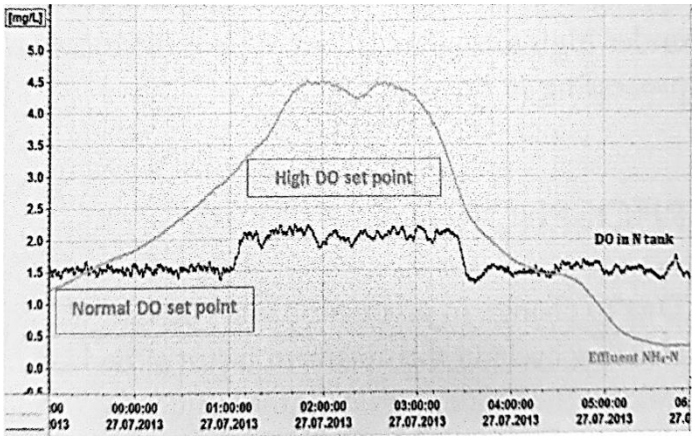


Figure 1. Exemple de contrôle et de suivi usuel dans une StaRRE (Ingildsen and Olsson, 2016)

Le résultat de ce système de suivi et contrôle usuel est un effluent de très bonne qualité. Les capteurs d'oxygène dissous et NH_4 montrent une autonomie dans l'opération du système avec des maintenances mineures par un opérateur telles que le nettoyage et la validation des capteurs.

Les limites dans ce suivi usuel sont le coût important lors de l'installation d'un tel système (contrôleur hautement performant, plusieurs capteurs). De plus, l'opérateur du système doit être une personne qualifiée afin de correctement opérer et résoudre des problèmes occasionnels.

1.3. Fautes et problèmes des capteurs

De nombreux auteurs ont rappelé que les méthodes de mesure doivent être simples, fiables et rapides à un coût de maintenance relativement faible (Bourgeois et al., 2001). Mais, dans le contexte des eaux usées, les capteurs peuvent subir une multitude de contraintes pouvant fausser la qualité et la fiabilité des données. D'après Mirin et Wahab (2013), ces conditions anormales ou défectueuses provenant d'un composant, d'un processus ou d'équipements conduisent à une faute du système. Les exemples courants de contraintes incluent le colmatage, le vieillissement des électrodes, l'abrasion des éléments optiques, les pannes matérielles (Alferes et al., 2013a; Chow et al., 2018; Mirin and Wahab, 2013; Plana, 2015; Yoo et al., 2008).
Les fautes provoquées par ces contraintes sont classées d'après Yoo et al. (2008) en quatre groupes (Figure 2), le biais, la dérive, la défaillance complète du capteur et la dégradation de la précision.

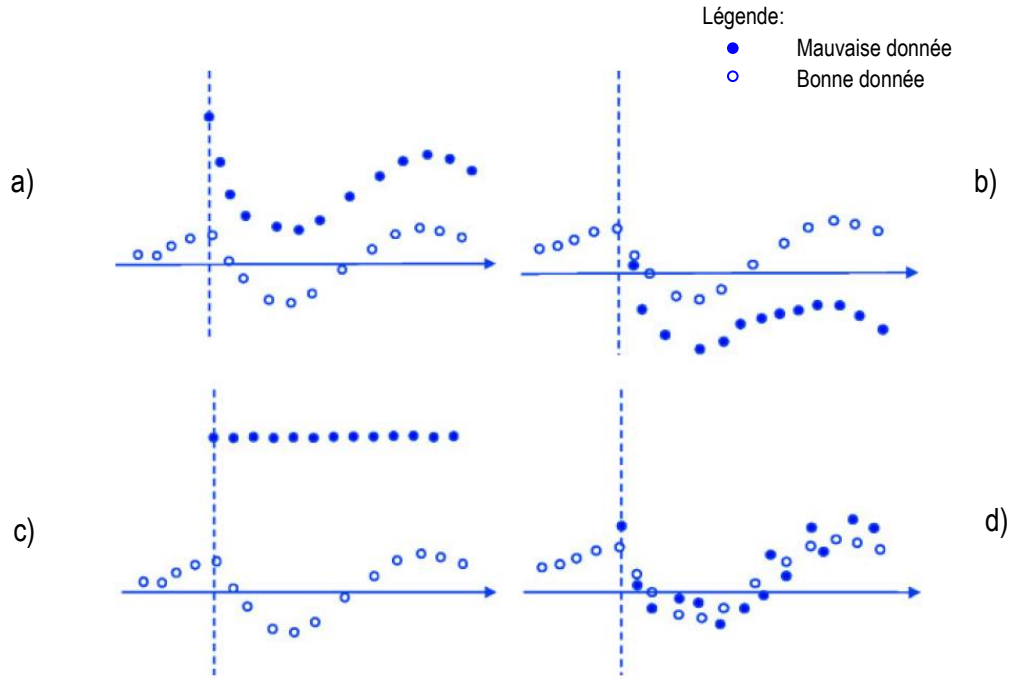


Figure 2. Fautes majeures pour les données provenant d'un capteur (a) Biais, (b) Dérive (c) Défaillance (d) Dégradation de la précision (Yoo et al., 2008)

Plana (2015) présente quelques fautes au sein de son mémoire avec des exemples de données réelles. La Figure 3 montre un exemple de défaillance complète au niveau du capteur. Pendant environ 10 jours, le capteur n'enregistrait pas de données à cause d'une interruption électrique sur le site d'étude. Tao et al., (2013) observent aussi une défaillance d'un capteur mesurant l'oxygène. En effet, une forte diminution est observée au niveau de la ligne rouge (à la mesure 190) (Figure 4). Cette baisse importante démontre bien la présence d'une défaillance du capteur.

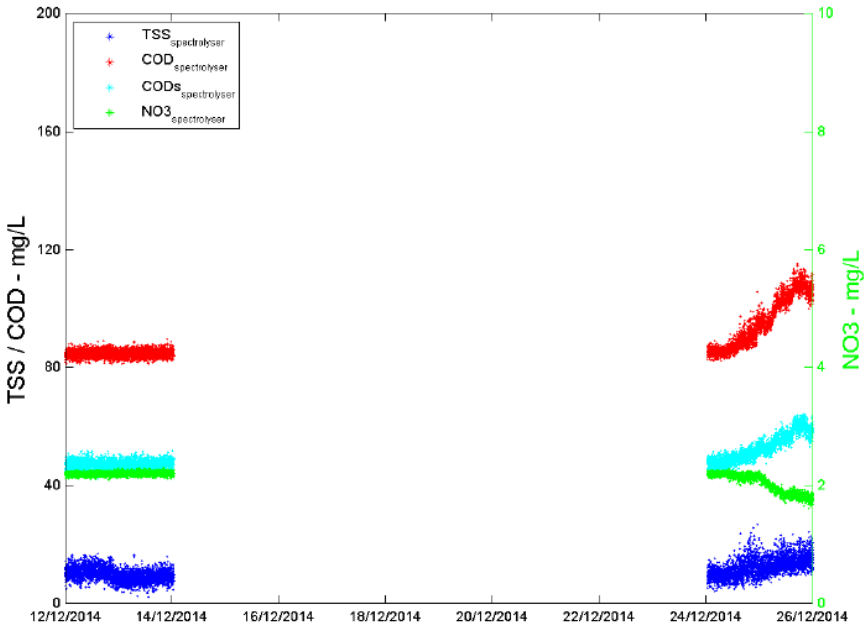


Figure 3. Exemple de défaillance au niveau d'un capteur (Plana, 2015)

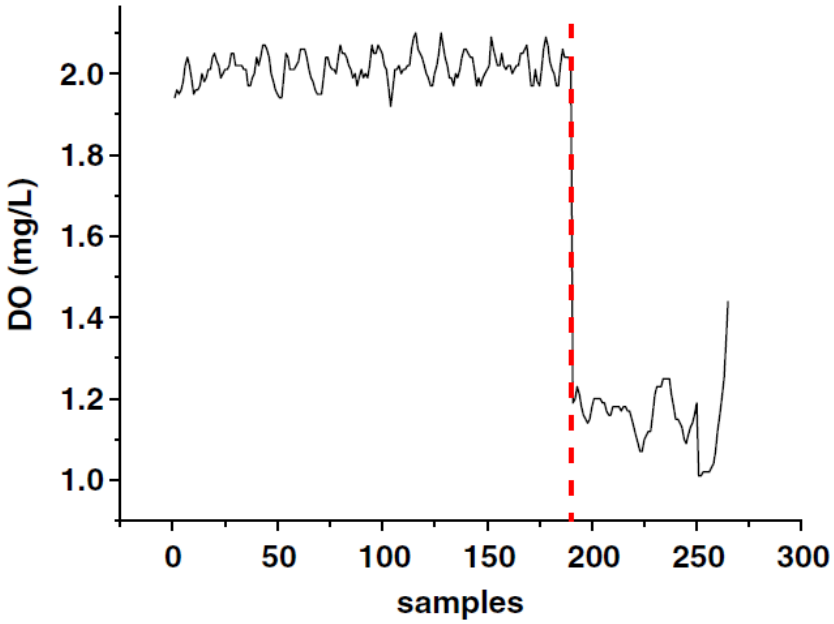


Figure 4. Exemple de colmatage d'un capteur d'oxygène dissous (Tao et al., 2013)

1.4. Méthodes de traitement des données

Afin de détecter les fautes explicitées dans la partie précédente et pouvant causer des erreurs de mesure ou des coûts de maintenance élevés, des méthodes de traitement de données sont développées depuis plusieurs années. Elles peuvent être appliquées directement sur le signal en ligne ou sur des séries de données préexistantes dans une base de données. Le but de ces méthodes est la détection, l'isolement et l'identification des fautes. Ces trois principes ont pour but le diagnostic des fautes. La détection permet de déceler les dysfonctionnements (les fautes). L'isolation a pour but de trouver la cause première de celles-ci. L'identification permet d'estimer le type ou la nature de la faute. Les méthodes permettent aussi l'amélioration de la qualité des données et la fiabilité du signal. Depuis plusieurs années, une multitude de méthodes sont développées afin d'analyser les données pour améliorer les opérations effectuées au sein des stations (Corominas et al., 2018). Parmi ces techniques, certaines ont pour but le traitement des données afin de détecter les fautes, les anomalies, les pannes de capteurs. Corominas et al., (2018) proposent une revue des méthodes dans le domaine de l'analyse des données afin d'améliorer les opérations dans les stations. Au sein de la revue, ils proposent une classification de ces méthodes sous trois grands niveaux : extraction basique d'information, extraction avancée d'informations et extraction d'informations humainement interprétables. Dans cette importante classification, quelques méthodes ont été choisies pour leur capacité à détecter les fautes qui sont reprises au Tableau 4.

Tableau 4. Méthodes pour la détection des fautes d'après Corominas et al., (2018)

Niveaux	***Techniques***
Extraction basique d'informations	- Diagramme de contrôle univarié
Extraction avancée d'informations	- Réduction de la dimension : Analyse en Composantes Principales (ACP) - Méthode de détection de prévision qualitative : Analyse qualitative des tendances (AQT)

1.4.1. Extraction basique d'informations

La première classe des méthodes s'intitule l'extraction basique de l'information. Elle a pour but la détection d'erreurs dans les séries de données, par exemple les données aberrantes, les anomalies ou les fautes, en ne nécessitant qu'une variable d'entrée (Corominas et al., 2018). D'après Ni et al., (2009), une donnée aberrante est une donnée isolée ou anormalement distante des modèles développés par des experts.

L'une des méthodes utilisées dans cette classe, est la méthode univariée qui a pour but de détecter les données aberrantes. Par exemple, Krajewski et Krajewski (1989) ont mis en place un modèle physique afin d'identifier ces données aberrantes, modèle présenté à la Figure 5.

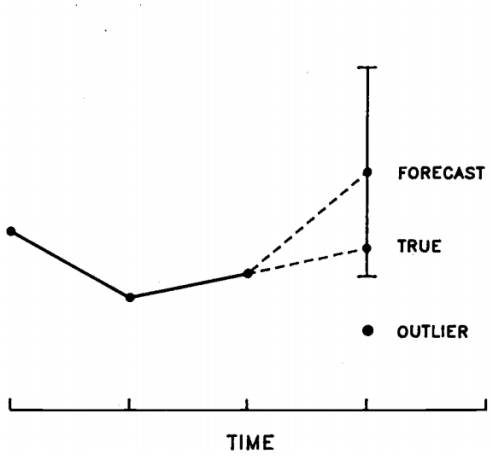


Figure 5. Modèle d'identification des données aberrantes (Krajewski and Krajewski, 1989)

Hill et Minsker (2010) ont développé un autre modèle, l'autorégressif univarié. Ce dernier se base sur une approche de comparaison entre les données mesurées et des données simulées. Si une donnée se situe à l'extérieur d'un intervalle défini au préalable par le modèle, elle est identifiée comme une donnée aberrante. D'autres méthodes univariées permettent de détecter les fautes en utilisant des modèles simulant le comportement des données en calculant sur un intervalle, un paramètre tel que la pente, la variance, la corrélation.

Un intervalle d'acceptabilité est positionné pour chaque paramètre à l'aide, par exemple, de la moyenne, de la médiane ou des quartiles (Ni et al., 2009). Mourad et Bertrand-Krajewski (2002) ont défini cet intervalle en choisissant des percentiles à partir de séries de données historiques afin de détecter la présence de fautes. De plus, ils ont mis en place une méthode permettant de détecter les pentes anormales dues aux données aberrantes. Pour ce faire, à l'aide d'une moyenne mobile, les données sont filtrées et ces dernières sont comparées aux données brutes.

Enfin, une étude récente a été effectuée par Alferes et Vanrolleghem (2016) où une évaluation efficace automatique de la qualité des données est réalisée sur des données provenant de capteurs mesurant la conductivité, la turbidité et le pH au sein d'une rivière. Deux parties sont identifiées dans la méthode. Premièrement, le filtrage permet une détection des données aberrantes avec un modèle autorégressif et un lissage de ces dernières à l'aide d'une moyenne mobile. Deuxièmement, une détection des fautes est effectuée en déterminant des paramètres tels que la pente et l'écart type ainsi que leurs limites d'acceptabilités. Cette méthode sera développée plus succinctement dans le chapitre 3 (section 3.2.1).

1.4.2. Extraction avancée d'information

La deuxième classe s'intitule l'extraction avancée d'information, elle permet une analyse multivariée des séries de données en prenant en compte plusieurs variables d'entrées en même temps. Elles peuvent être effectuées à l'aide de plusieurs méthodes, par exemple en utilisant des méthodes simples multivariées telles qu'exposées par certains auteurs. C'est le cas de Mourad et Bertrand-Krajewski (2002) qui utilisent cette

méthode en installant deux capteurs en redondance pour trouver des tendances ou des différences anormales dans les données. Le résultat permet ainsi de détecter les fautes. Dans leur article, ils proposent aussi d'exploiter les variables en corrélation telles que la vitesse d'écoulement et la hauteur d'eau pour trouver ces anomalies. Par la suite, d'autres méthodes multivariées plus complexes ont été développées. Premièrement, des méthodes qui permettent de représenter une information de plusieurs variables en un jeu de variables réduit tout en gardant un maximum d'information issues des données originales (Corominas et al., 2018). C'est notamment le cas de l'Analyse en Composantes Principales (ACP), qui est la technique de réduction de dimension la plus connue à ce jour. Cette dernière a par exemple été utilisée dans l'analyse de la composition physico-chimique des eaux usées mais aussi dans la description de la distribution de données correspondant à des conditions opérationnelles normales. Dans la détection des fautes, elle a aussi montré sa robustesse pour détecter les anomalies dans des séries de données (Alferes et al., 2013b, 2013a; Rosen and Olsson, 1998; Tao et al., 2013; Yoo et al., 2003). Rosen et Olsson (1998) ont par exemple utilisé la PCA afin de détecter des fautes de capteurs dans des séries de données en ligne provenant de l'affluent d'une station. Une étude récente d'Alferes et al., (2013b) , utilisant aussi l'ACP, a permis de détecter certaines fautes dans des séries de données tels que le bruit et la dérive des données. La méthode de détection des fautes par l'ACP sera développée plus en détail dans le chapitre 3 (section 3.2.2).

En revanche, l'ACP est une méthode de nature quantitative et nécessite soit une connaissance précise du processus soit une grande quantité de données. En revanche, de nombreux auteurs ont récemment développé des méthodes pour le diagnostic des capteurs, de la détection des fautes, de la caractérisation de la décantation des boues de décanteur et dans les systèmes automatiques de débits continusm, qui sont des données qualitatives. Ces méthodes sont basées sur une identification automatique de prévisions visuellement identifiables et compréhensibles dans des séries de données (Derlon et al., 2017; Thürlimann et al., 2018; Villez & Habermacher, 2015; Villez et al., 2012). Cette identification se base sur l'analyse qualitative des tendances (AQT). Villez et Habermacher (2015) ont fait la comparaison entre la méthode AQT et la méthode multivariée ACP pour la détection des fautes dans les séries de données. L'AQT a démontré des avantages comparativement à l'ACP.

1.5. Validation des capteurs

Afin d'assurer une fiabilité et une précision des données fournies par les capteurs, les séries de données doivent être validées. Deux méthodes sont connues :

- Diagramme de contrôle
- Redondance des capteurs

1.5.1. Diagramme de contrôle

La méthode du diagramme de contrôle développée par Walter A. Shewhart permet de valider des données provenant de capteurs (Montgomery, 2009; Plana, 2015). Thomann et al. (2002) reprennent et explicitent plusieurs étapes pour valider des données provenant de capteurs assurant le suivi de l'efficacité du traitement. Le schéma dans la Figure 6 explique les diverses étapes dans la validation de données.

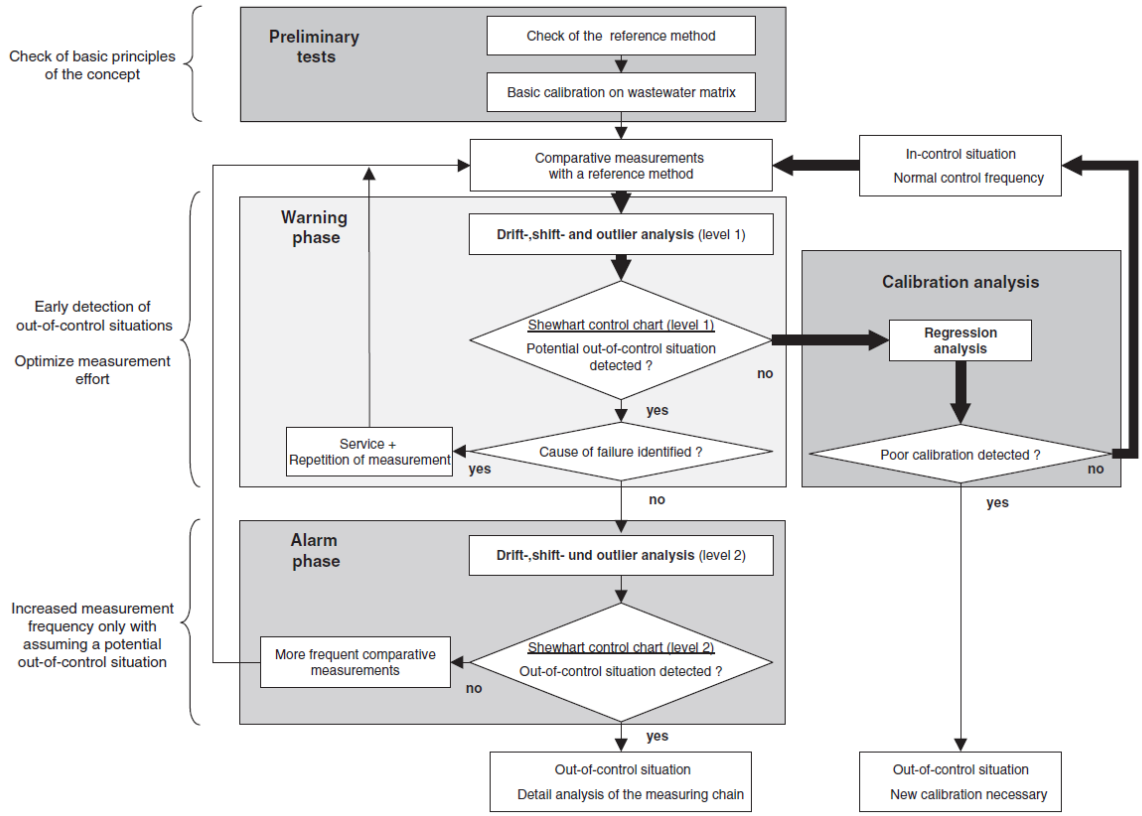


Figure 6. Diagramme de validation des données de capteurs (Thomann et al., 2002)

La méthode de Thomann (2002) permet de détecter les fautes de capteurs. Au sein de cette méthode de validation, deux sous-étapes sont observées :

- Les tests préliminaires
- L'analyse des fautes : diagramme de contrôle

1.5.1.1. *Tests préliminaires*

Au sein de cette sous-étape, différents tests sont effectués, basés sur des méthodes de référence (Thomann, 2008; Thomann et al., 2002). Ces dernières sont au nombre de trois :

- L'analyse de la calibration des capteurs avec des solutions standards ;
- La comparaison des données de capteurs avec celle d'une sonde portable ;
- La comparaison des données de capteurs avec des mesures de laboratoire.

1.5.1.2. *Analyse des fautes*

C'est dans cette sous-étape que le diagramme de contrôle montré en Figure 7 est utilisé dans l'analyse des pannes (Montgomery, 2009). Sur cette dernière, trois lignes horizontales sont présentes et correspondent au niveau général du processus (ligne centrale) et aux limites basses et hautes du processus. Si la mesure dépasse l'une des limites, le processus n'est plus considéré sous contrôle (Montgomery, 2009).

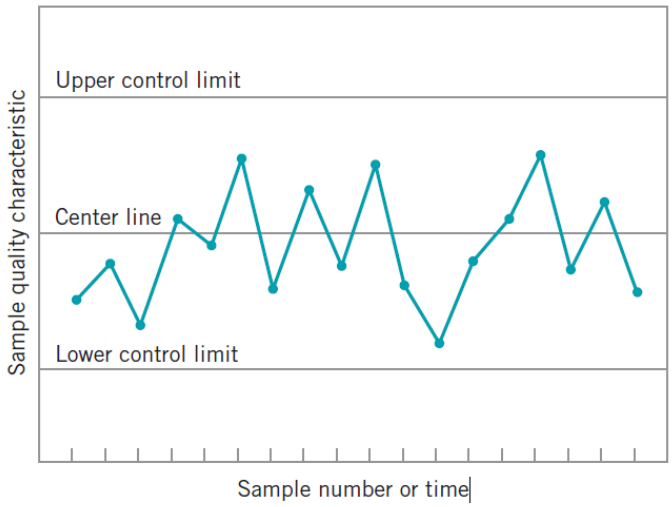


Figure 7. Diagramme de contrôle (Montgomery, 2009)

Mais, avant d'obtenir la courbe ci-dessus, plusieurs étapes doivent être effectuées :

- La détermination de la différence entre la valeur de référence (mesure en laboratoire, solution standard, mesure d'une sonde portable) et les valeurs du capteur grâce à l'équation ci-dessous:

$$D = C_{ref} - C_{mesures} \qquad \text{Équation 3}$$

- La détermination de la déviation standard s_D à l'aide de la formule suivante :

$$s_D = \sqrt{\frac{\sum_{i=1}^{n}\left(D_i - \check{D}\right)^2}{n-1}} \quad avec\ \check{D} = \frac{1}{n}\sum_{i=1}^{n} D_i \qquad \text{Équation 4}$$

Avec :

D : la différence entre la valeur de référence et la valeur du capteur;

$\check{D}$: la moyenne des différences;

n : le nombre d'échantillons.

Dans la littérature, ces équations peuvent être utilisées seulement lorsque le nombre d'échantillons est supérieur à 10 (Plana, 2013, 2015; Thomann et al., 2002).

La détermination de la limite haute (LH) et basse (LB) est réalisée à l'aide des équations suivante :

$$LH = +L \times s_D \qquad \text{Équation 5}$$

$$LB = -L \times s_D \qquad \text{Équation 6}$$

Avec:

L: le paramètre.

Montgomery (2009) montre que « L » peut prendre une valeur entre 2 ou 3 qui correspond un intervalle de probabilité de 95 à 99 % que les valeurs soient acceptables.

Sur la Figure 8 un exemple de diagramme de contrôle comprenant les différences calculées et les différentes limites est présenté.

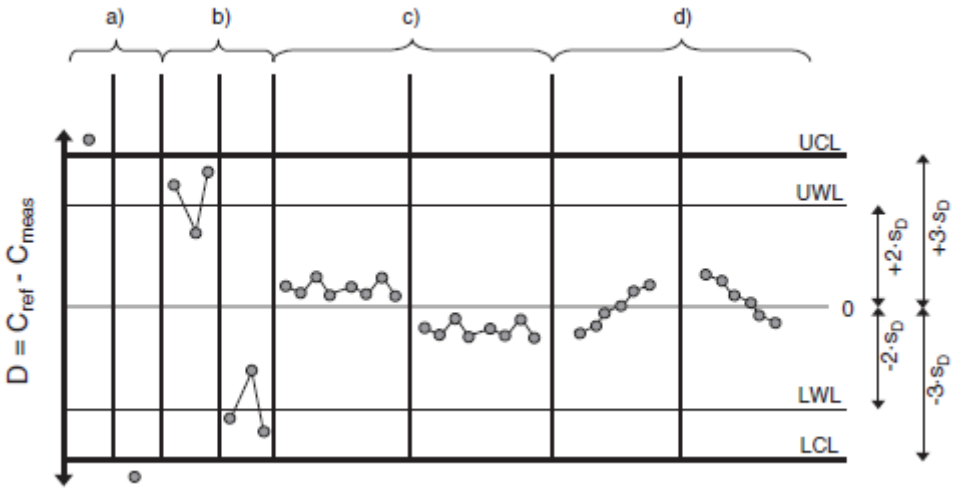


Figure 8. Exemple de diagramme de contrôle (Montgomery, 1996)

Dans cet exemple, les limites hautes (UCL) et basses critiques (LCL) sont représentées par deux lignes épaisses noires correspondantes à la valeur L égale à 3. Les limites hautes et basses d'avertissement notées UWL et LWL correspondent à la valeur L, 2. Certains auteurs ont utilisé ces limites afin d'être plus restrictif dans leurs analyses de données. D'autres cas peuvent aussi être présentés, comme indiquer sur la Figure 8 (les cas a), b), c) et d)). Le cas a) montre deux points consécutifs hors des limites critiques UCL et LCL. Le cas b) montre quatre points sur six hors des limites d'avertissement UWL et LWL. Enfin, les cas c) et d) sont une accumulation de points dans les limites et au niveau de la ligne centrale. Ces deux derniers montrent un capteur correctement calibré.

1.5.2. Redondance des capteurs

Une deuxième méthode de validation des données de capteurs est basée sur le principe de la redondance des capteurs installés à une même localisation. Cette dernière est similaire à un processus utilisé en informatique portant le nom de « RAID ». « RAID » désigne « Redundant Array of Independent Disks ». On en

retrouve plusieurs variantes tels que RAID1, RAID0 et RAID5. Par exemple, RAID1 est un système regroupant deux disques ayant les mêmes données à l'intérieur. Ceci a pour but de préserver un des disques avec les données si l'autre tombe en panne (Patterson et al., 1989).

Dans la littérature, quelques auteurs ont aussi développé l'idée de la redondance du matériel et ceci dans le domaine du traitement des eaux usées. Villez et al. (2013) démontre l'importance du placement de deux capteurs à une même localisation. Ceci permet de comparer les séries des deux capteurs et ainsi de détecter une ou plusieurs fautes dans les séries de données. Ce qui peut permettre à l'identification d'un dysfonctionnement d'un des deux capteurs. Ils suggèrent aussi que chaque paramètre doit être validé par plus d'une mesure même si tous les capteurs sont à des endroits différents. Cependant, cette technique de validation des capteurs peut être coûteuse puisque plusieurs capteurs sont nécessaires (Villez et al., 2013). Alferes et al., (2013b) démontre aussi l'importance de la redondance des capteurs, au sein d'une rivière pour le suivi de différentes variables tels que le pH, la turbidité, la température et la conductivité à l'aide de plusieurs capteurs. Pour chaque chacun d'entre eux, deux capteurs ont été installés. La Figure 9 montre ce suivi pour les quatre variables. Mais, l'augmentation du nombre capteur présentent un coût ce qui n'est pas gérable d'un point de vue financier dans certaines études visant le contrôle de la qualité de l'eau. En effet, dans certains cas, le contrôle de la qualité des données peut restreindre le contrôle de la qualité des eaux usées en plaçant à une même localisation deux mêmes capteurs et en abandonnant d'autres localisations.

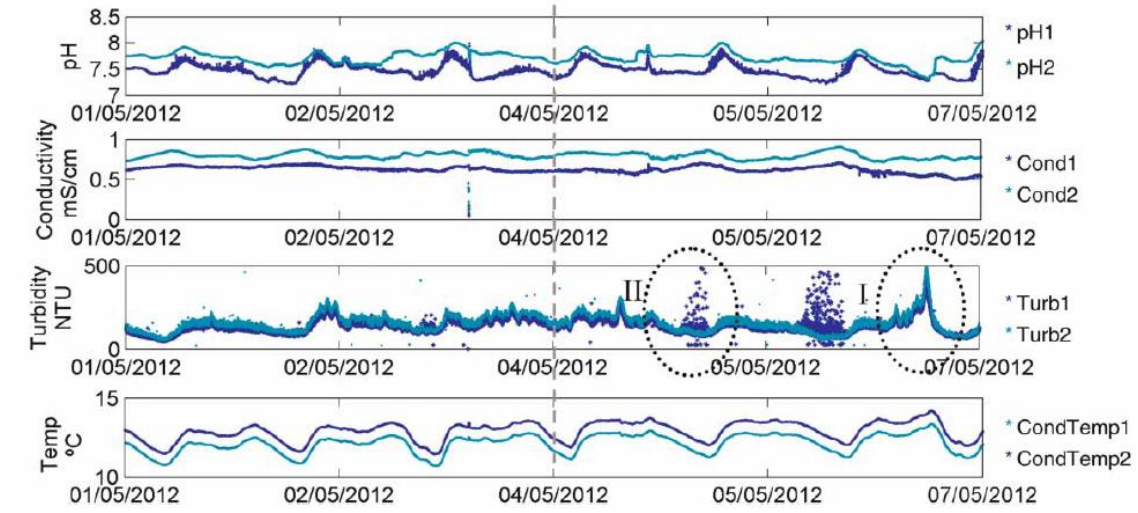


Figure 9. Suivi en continue du pH, la conductivité, la turbidité et la température avec une redondance des capteurs (Alferes et al., 2013b)

Chapitre 2 Problématique et objectifs

Le traitement des eaux usées passe par la surveillance de l'effluent. Ceci est effectué soit en contrôlant la qualité d'échantillons ponctuels grâce à des mesures de laboratoire ou soit en utilisant des capteurs, tels que vu au chapitre 1. Concernant ces deux méthodes, l'utilisation de capteurs semble la plus pertinente car elle permet d'effectuer des mesures sans la présence constante d'un opérateur et en continu (par exemple toutes les secondes). Cependant, cela mène à une collecte très importante de données. Par exemple, un capteur installé pendant trois mois et demi et prenant une mesure toutes les cinq secondes génèrera environ deux millions de données. Le domaine des eaux usées est aussi un milieu néfaste pour les capteurs dus, entre autres, à une importante quantité de particules et autres matières négatives pour les capteurs. Ainsi, des colmatages de capteurs, des pannes, des anomalies et autres problèmes peuvent mener à des pertes de données généralement comprises entre 5 et 60 % (Alferes et al., 2013a). Afin de minimiser ces pertes, un programme de maintenance régulier où des systèmes de nettoyage automatiques doivent être mis en place.

Assurer la qualité des données prélevées est une tâche assez complexe due à la grande variété de capteurs et de variables mesurées. Des outils automatiques ont donc été développés afin de détecter, isoler et diagnostiquer ces pannes et anomalies. Cependant, ces outils ont traditionnellement été développés et appliqués à des cas spécifiques.

2.1. Problématique

C'est dans ce cadre-là que s'inscrit le projet de maîtrise, de rendre les outils automatiques d'évaluation de la qualité des données développés par le passé, plus simple, modulaire, et applicable à une multitude de cas.

2.2. Objectifs

Afin de répondre à cette problématique, l'objectif principal de ce projet de maîtrise est de rendre ces outils plus clairs et documentés afin de faciliter leur utilisation. Concernant la modularité et l'applicabilité, elles traduisent une reprogrammation des fonctions pour chaque étape des outils et une recherche des paramètres généraux pour différents capteurs.

Afin d'atteindre cet objectif principal, plusieurs objectifs spécifiques devront être validés :

- Comprendre des outils de détection des fautes mis en place précédemment afin de se familiariser avec le logiciel MATLAB et de comprendre les divers scripts construits par le passé;

- Développer des fonctions d'uniformisation des données pour qu'elles aient un format de sortie unique;

- Créer des blocs de fonctions pour chaque étape des outils afin de les rendre modulaires et plus intelligibles dans leur utilisation;
- Tester les fonctions développées obtenues à l'étape précédente, sur plusieurs cas d'étude afin de montrer l'applicabilité des outils pour diverses séries de données;
- Créer des Standard Operating Procedures (SOPs) afin de permettre une utilisation adéquate des outils de traitement de données.

Chapitre 3 Matériel et Méthodes

Premièrement, les sites d'études d'où proviennent l'ensemble des données, seront évoqués ainsi que les systèmes de surveillance et les capteurs. Par la suite, les méthodes de traitement de données étudiées seront exposées et évaluées pour l'ensemble des sites d'étude. Ces méthodes seront utilisées afin de détecter les fautes au sein des diverses séries de données. Les données traitées provenant des projets bord*EAU*x et kam*EAU* seront employées dans le but de calibrer et valider deux modèles développés par les étudiants au doctorat Julia Ledergerber et Bernard Patry respectivement. Les données traitées serviront aussi à la compréhension des fautes des capteurs afin d'éviter qu'elles se reproduisent dans le futur (Chapitre 2).

3.1. Sites d'étude

Dans ce travail de maîtrise, les données étudiées proviennent de trois projets de recherche :

- pil*EAU*te
- kam*EAU*
- bord*EAU*x

Chacun d'eux sera développé succinctement ci-dessous ainsi que les systèmes de surveillance présents et les capteurs.

3.1.1. pil*EAU*te

L'usine pil*EAU*te, située à l'Université Laval, est une usine pilote de traitement de la ressource en eau construite en 2015 pour des fins de recherche (Figure 10). Elle est située au pavillon Adrien-Pouliot, Département de génie civil et de génie des eaux de l'Université Laval (Québec, Canada). La Figure 11 représente l'ensemble des installations s'y trouvant ainsi que les processus s'y effectuant.

Figure 10. L'usine pilEAUte au sein du pavillon Adrien-Pouliot, Université Laval (Québec, Canada)

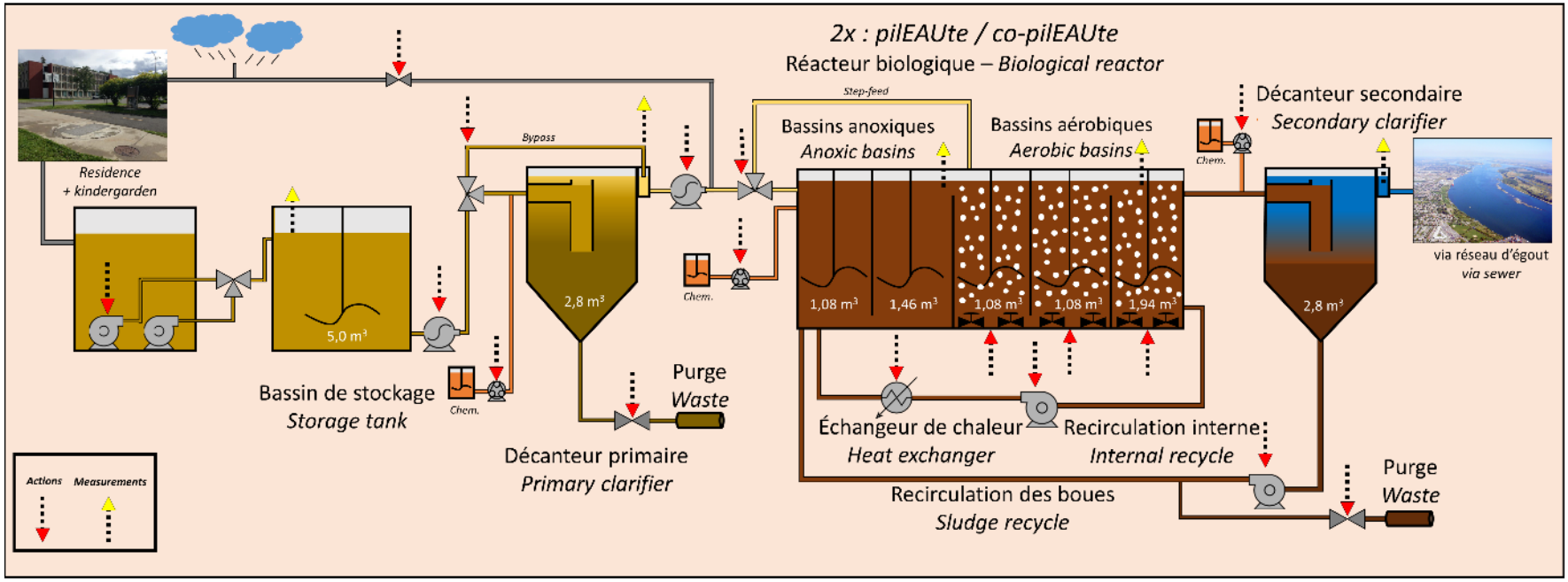


Figure 11. Schéma d'installation de l'usine pilEAUte de traitement des eaux usées

Pil*EAU*te est alimenté par l'eau usée domestique provenant d'une résidence universitaire ainsi que de deux garderies (Figure 11). La résidence peut héberger en pleine session universitaire 480 personnes. Les deux garderies peuvent accueillir un total de 150 enfants. L'eau alimentant l'usine comporte aussi l'eau de ruissellement se drainant dans le système tel que l'eau du stationnement et des toits (Figure 11).

L'usine comporte plusieurs installations telles qu'une station de pompage, un bassin de stockage, un décanteur primaire, deux chaînes de réacteurs biologiques parallèles appelées pil*EAU*te et co-pil*EAU*te et deux décanteurs secondaires (Figure 11). Plus précisément, l'eau usée provenant de la résidence passe avant par une station de pompage où deux pompes déchiqueteuses prétraitent l'eau et ses composants non solubles (Figure 11). Ces pompes déchiqueteuses permettent de prétraiter l'eau brute sans dégrilleur, afin de protéger les pompes et les tuyaux situés dans l'usine. De cette station de pompage, l'eau arrivant dans l'usine, est stockée dans un bassin tampon d'un volume de 5 m^3 (Figure 11). Ce dernier permet d'homogénéiser les eaux d'entrée à l'aide d'un mélangeur (Figure 11). Il est aussi utilisé comme une réserve lors des périodes nocturnes quand il n'y a pas assez d'eau produite. Cependant, durant ces périodes, l'homogénéisation est un défaut car l'affluent est moins représentatif.

Par la suite, l'eau stockée se dirige vers le décanteur primaire (premier bassin de la chaine de traitement) à l'aide d'une pompe à un débit fixe de 1,1 m^3/h. Ce décanteur dont le volume est de 2,8 m^3 et de 1,2 m^2 de surface permet aux particules de décanter au fond du bassin par le phénomène de gravité. Certains auteurs ont montré que ce décanteur pouvait éliminer en moyenne 55 à 70 % de la MES et 35 % de la DCO (Philippe, 2016; Ponzeli, 2018; Tohidi, 2018). Ces valeurs sont en accord avec les performances théoriques pour un décanteur primaire tel que 60 % pour les MES et 30 % pour la DCO (Nathanson, 2003).

L'effluent clarifié des particules en sortie du décanteur primaire est redirigé vers deux lignes de traitement biologique (pil*EAU*te et co-pil*EAU*te) par deux pompes ayant un débit de 0,5 m^3/h. Ces lignes de traitement ont les mêmes conceptions et permettent d'éliminer la matière carbonée et azotée en utilisant une configuration pré-dénitrification (Élimination des nitrates en zone anoxique au début des lignes de traitement) comportant deux bassins anoxiques et trois bassins aérobiques (Figure 11 et Figure 12). Des mélangeurs sont présents au sein des deux premiers bassins afin d'avoir une homogénéisation de l'eau (Figure 11). Au contraire, dans les bassins aérobiques, des diffuseurs d'air comprimé permettent d'homogénéiser l'eau et de fournir de l'oxygène à la flore bactérienne. Enfin, deux recirculations internes ayant un débit de 1,5 m^3/h, du 5^e bassin vers le premier, permettent la dénitrification (Figure 11).

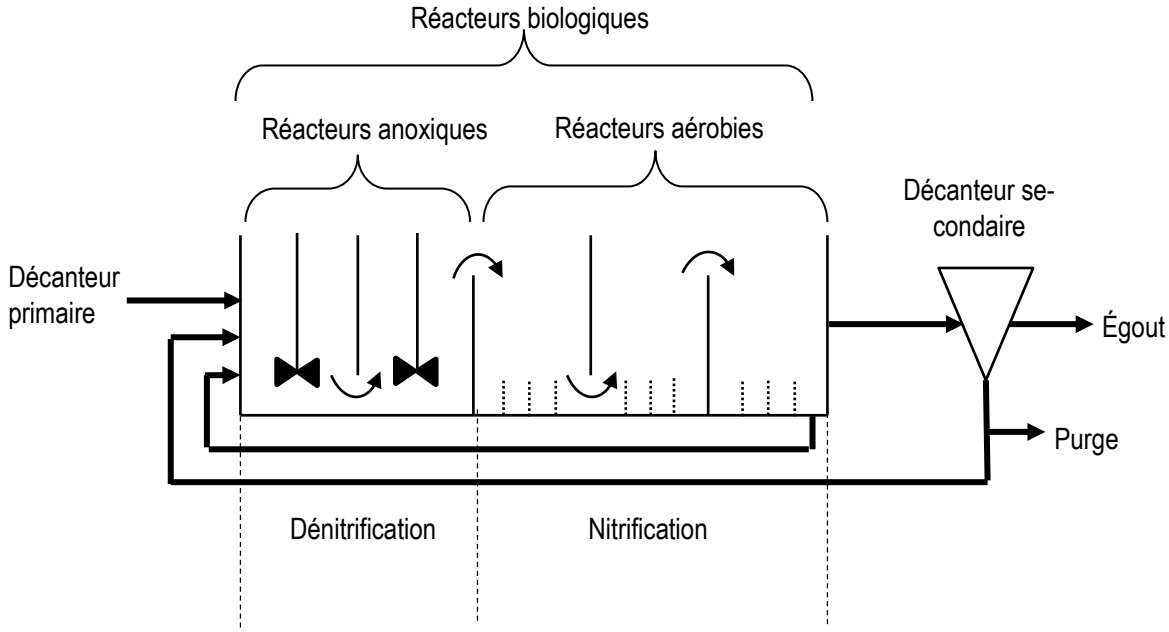


Figure 12. Configuration des réacteurs biologiques

Les effluents du pil*EAU*te et du co-pil*EAU*te sont dirigés vers deux décanteurs secondaires où l'eau est séparée des boues par gravité. Ils ont un volume de 2,8 m^3 et 1,2 m^2 de surface. Afin de garder la concentration en bactéries au sein des bassins biologiques à un niveau acceptable, une boucle de recirculation des boues avec un débit de 0,5 m^3/h entre les clarificateurs secondaires et les premiers bassins biologiques est mis en place (Figure 11 et Figure 12). Pour réduire la quantité de boues en excès au sein du décanteur, une purge manuelle ou automatique permet de vidanger les clarificateurs. Enfin, l'ensemble des eaux est rejeté dans le réseau d'égout puisque pil*EAU*te n'est utilisé que pour la recherche.

L'usine pil*EAU*te offre une importante flexibilité dans son opération dans le but de pouvoir tester diverses conditions. Par exemple, les débits de recirculation interne ou de boues peuvent être soit diminués soit augmentés afin d'observer leurs effets sur le traitement.

3.1.1.1. Système de contrôles (Station monEAU et SCADA/PLC)

Au sein de cette usine, l'ensemble des instruments tels que les capteurs, les pompes, etc. est contrôlé par des deux systèmes :

- Station mon*EAU*
- SCADA/PLC

3.1.1.2. Station monEAU

La station « mon*EAU* » est un système d'acquisition de l'ensemble des données provenant de divers capteurs. La compagnie Primodal Inc. (Hamilton, Ontario, Canada) la commercialise sous le nom RSM30. Rieger et Vanrolleghem (2008) ont explicité les points importants qui ont défini sa conception :

- Un système flexible
- Un système ouvert et modulaire
- Une base de données
- Une connectivité à distance
- Une évaluation automatique de la qualité des données
- Un logiciel facilement utilisable et orienté pour l'utilisateur

Sa flexibilité et sa connectivité à distance permettent son implantation dans divers environnements pouvant être difficiles d'accès (rivières, réseaux d'égout, StaRREs, etc…). Cette même flexibilité permet aussi la connexion d'une multitude de capteurs de diverses compagnies (HACH, s::can, xylem, etc…) grâce à plusieurs protocoles de communication tels que profibus, HART, 4-20 mA, USB, etc… Au sein de la station, le logiciel mon*EAU* appelé « basestation » est facilement utilisable dû à sa modularité. Des outils de visualisations, d'évaluation de la qualité des données peuvent y être ajoutés à la demande de l'utilisateur. L'information disponible dépend des qualifications de l'utilisateur, il est orienté vers l'utilisateur (opérateur, chercheur, doctorant, etc…). Cependant, l'ensemble des concepts n'est pas mis en place. Par exemple, l'évaluation automatique de la qualité des données étudiées dans ce projet n'est pas installée. La Figure 13 montre la station.

Figure 13. La station monEAU avec (a) l'extérieur de la station (b) l'intérieur de la station

3.1.1.3. SCADA/PLC

Un système SCADA (Supervisory Control and Data Acquisition) est un système de télégestion ayant pour but de contrôler et de surveiller en temps réel des variables provenant de procédés (Daneels and Salter, 1999). L'usine pil*EAU*te est contrôlée par un système SCADA (Figure 14). Ce dernier est programmé à l'aide du logiciel « FactoryTalk ». Sur l'interface montrée en Figure 14, différents onglets peuvent être observés :

- Vue générale : cet onglet permet d'avoir une vue d'ensemble de l'usine-pilote dans toute son intégralité;
- Système : cet onglet permet de voir le schéma informatique et d'automation de l'usine-pilote;
- Consignes : cet onglet permet de contrôler les délais en lien avec les alarmes, l'alimentation électrique et le débit;
- Alarmes : cet onglet affiche toutes les alarmes déclenchées possibles dans le procédé;
- Reconnaissances de toutes les alarmes : cet onglet permet de reconnaître les alarmes déclenchées soudainement;
- Sécurité : cet onglet permet de se connecter à une session avec un des trois niveaux de contrôle : Administrateur, superviseur et opérateur, de fermer une session et de changer de mot de passe;
- Tendances : cet onglet permet de visualiser les différentes courbes de tendances des variables mesurées.

SCADA permet d'opérer le pil*EAU*te. L'opérateur peut, par exemple modifier le débit d'une pompe ou augmenter l'aération au sein des bassins (Figure 15). Aussi, le système permet de surveiller certaines variables en temps réel tels que la conductivité, le niveau dans le bassin tampon, etc. (Figure 19).

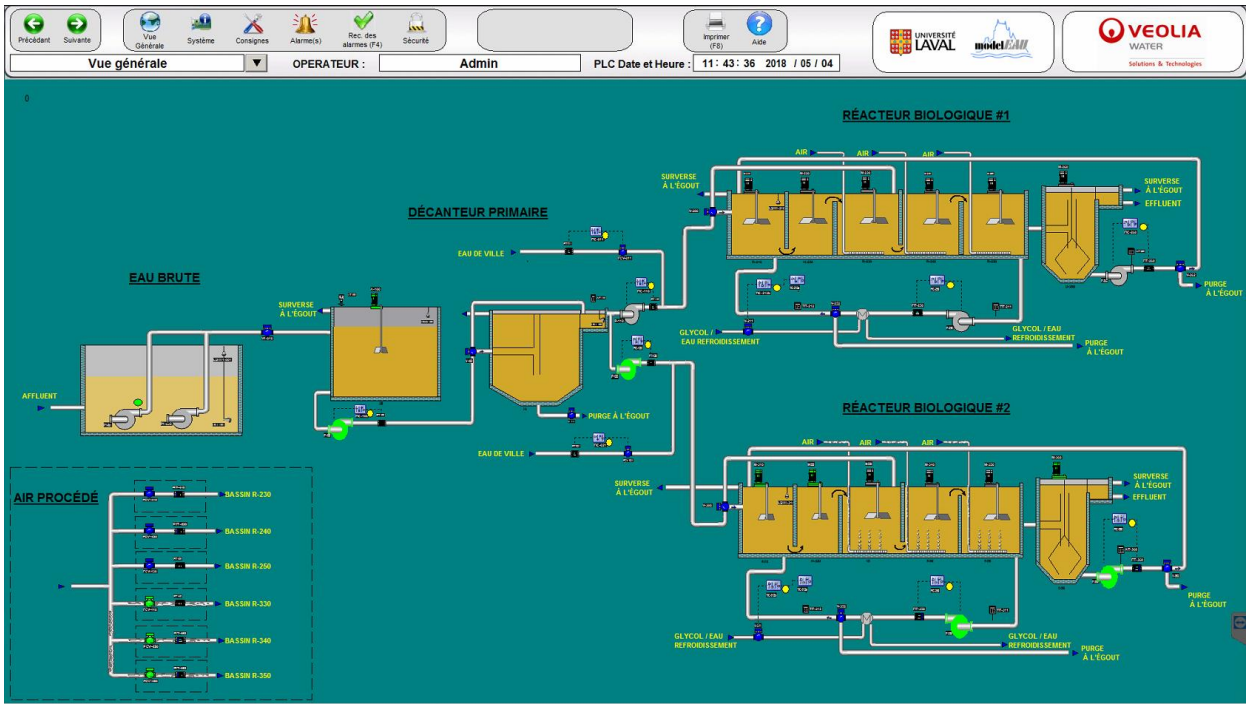


Figure 14. L'interface graphique d'utilisateur du SCADA de l'usine pilEAUte

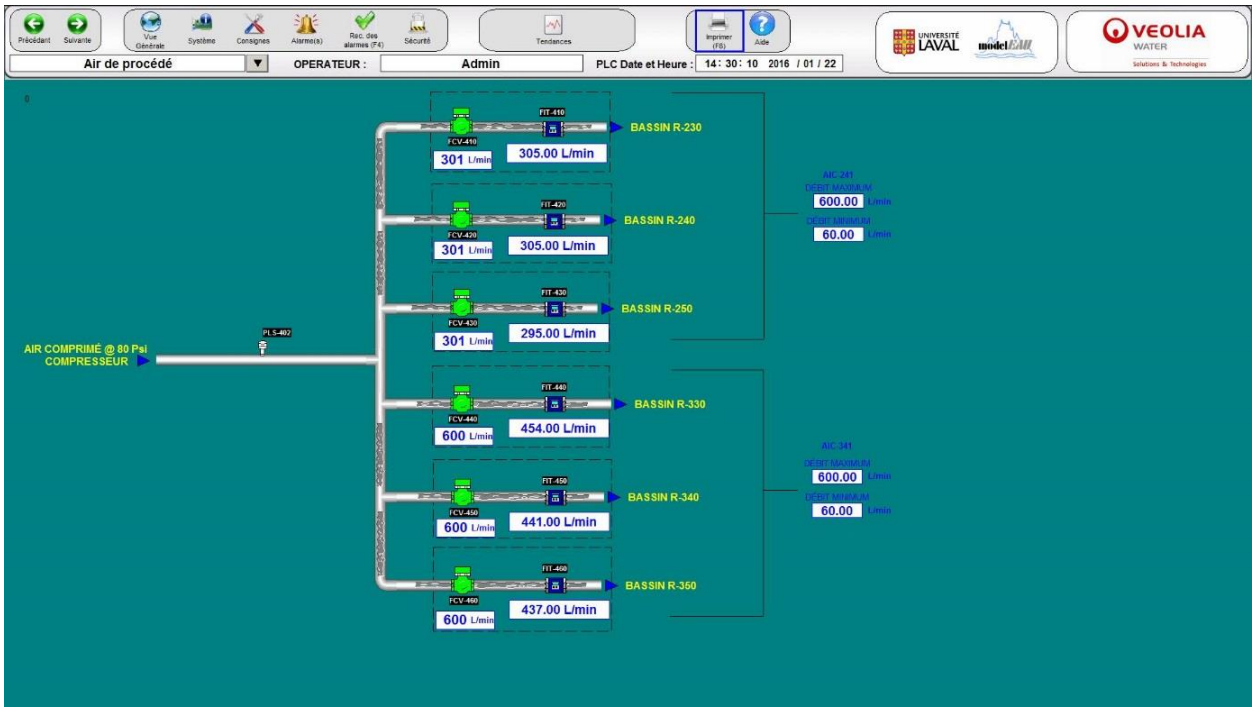


Figure 15. Supervision du débit d'air comprimé au sein des réacteurs biologiques (pilEAUte : R230/240/250 ; copilEAUte : R330/340/350)

Afin de programmer le système SCADA, la programmation PLC est utilisée. PLC provient de « Programmable Logic Controllers ». En français, PLC se nomme Automate Programmable Industriel (API). Au sein du pil*EAU*te, la programmation API est réalisée à l'aide du logiciel « Logix5000 » de Allen Bradley (Bradley, 2018). La Figure 16 représente un exemple de commande d'API au sein du pil*EAU*te. Sur cet exemple, la surveillance du niveau du bassin tampon a été implantée tout en mettant en place une alarme sur ce niveau. Lorsque le niveau du bassin tampon tombe en-dessous d'un seuil, la pompe P100 s'arrête de fonctionner.

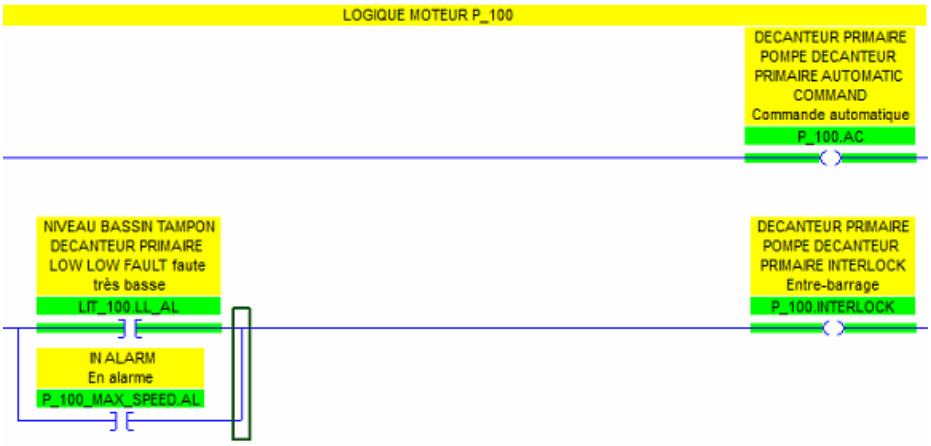


Figure 16. Exemple de programmation API avec la surveillance du niveau du bassin tampon avec l'implantation d'une alarme sur le niveau

Plus généralement, un API fonctionne avec une entrée et une sortie reliées par une branche (Figure 16). L'entrée donne une certaine condition ou l'état dans le système. La Figure 17 représente une entrée dans un API : « Observer si l'alarme indique un très bas niveau d'eau dans le bassin de stockage ».

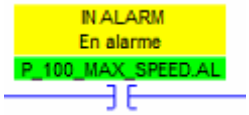


Figure 17. Exemple d'entrée dans un API

La sortie quant à elle, utilise l'information de l'entrée et prend une certaine action. La Figure 18 fournit une sortie dans un API : « Si l'alarme indique un très bas niveau d'eau dans le bassin de stockage, la pompe P-100 s'arrêtera ».

Figure 18. Exemple de sortie dans un API

3.1.1.1. Capteurs et emplacements

Au sein de l'usine pil*EAUt*e, un ensemble de capteurs mesurant en continu ou en discontinu est présent. Le Tableau 5 résume les variables mesurées par les quatorze capteurs installés. Cette multitude de variables est aussi marquée par une diversité de méthodes de mesure. Ainsi, l'ensemble des données collectée par ces capteurs permet d'une part de comprendre l'état de leur fonctionnement et d'autre de surveiller l'efficacité du traitement des eaux usées.

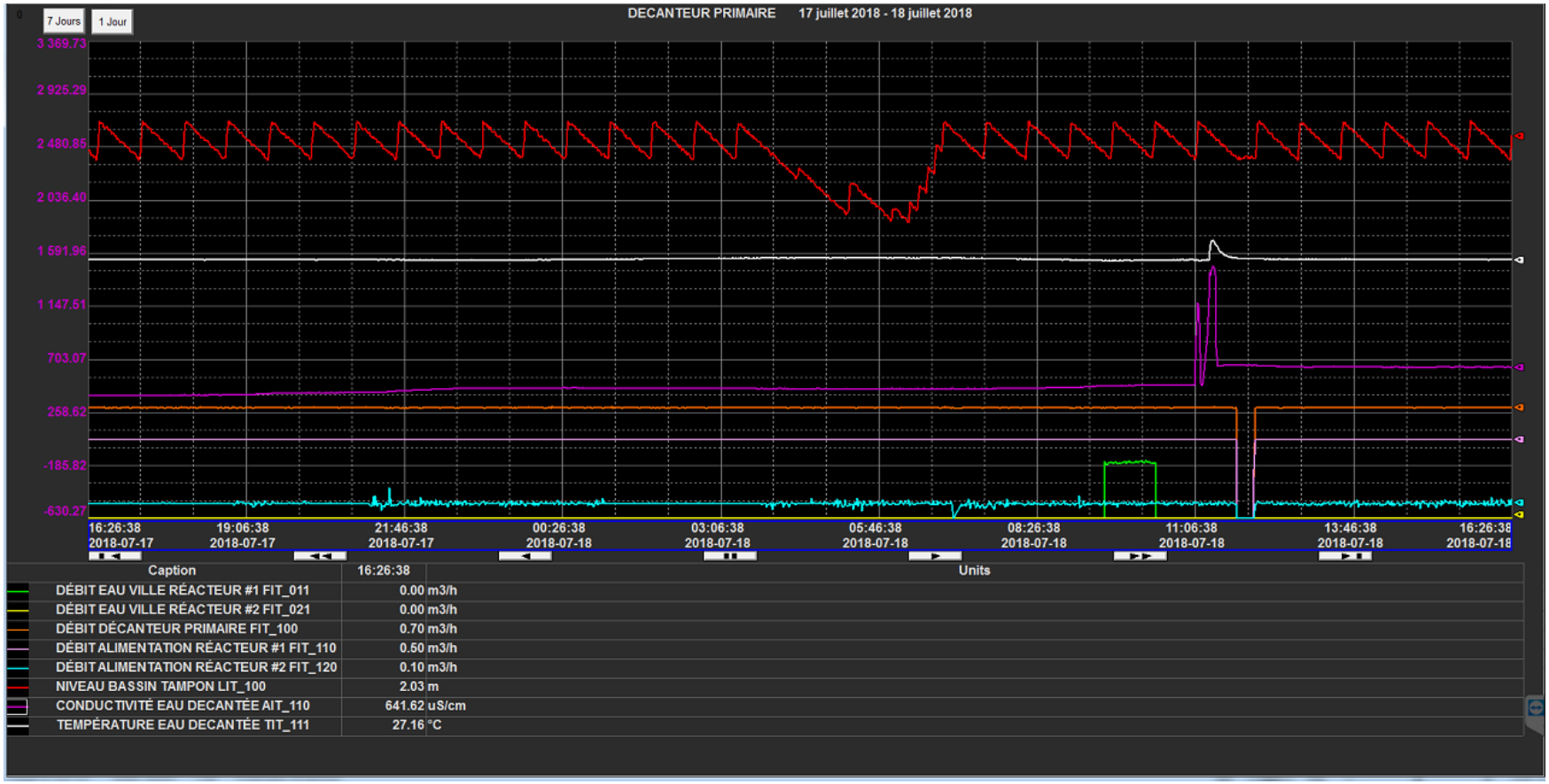


Figure 19. Surveillance de quelques variables (débits, niveau d'un bassin, conductivité, température) de l'usine pilEAUte à l'aide de SCADA

Tableau 5. Capteurs au sein de l'usine pilEAUte

Localisation des capteurs	***Nom du capteur (Fabricant)***	***Variables surveillées***	***Principe de mesure***	***Système de surveillance***
Effluent du décanteur primaire	spectro::lyser (s::can)	DCO_{total}, $DCO_{soluble}$, MES, NO_3	Absorption UV-VIS spectrophotométrie	mon*EAU*
	ammo::lyser (s::can)	NH_4-N, K, température, pH	Électrodes d'Ion Sélective (EIS)	mon*EAU*
	IQ SensorNet VARION (Xylem)	NH_4-N, K et température	Électrodes d'Ion Sélective (EIS)	mon*EAU*
	Conductimètre 3700sc Digital (HACH)	Conductivité et température	Induction	SCADA
	RODTOX NG (Kelma)	DBO_{ct} et toxicité	Respirométrie	mon*EAU*
Réacteur biologique Bassin 2	Solitax sc (HACH)	MES	Diffusion	SCADA
Réacteur biologique Bassin 4	LDO (Luminescent Dissolved Oxygen) (HACH)	Oxygène dissous et température	Luminescence	SCADA
Réacteur biologique Bassins 2&5	Analyseur simple et multi-variables Trescon/ Membrane Système de préparation d'échantillons Purcon (Xylem)	NH_4-N, NO_2-N, NO_X-N (NO_2-N & NO_3-N total)	Colorimétrie	mon*EAU*
Ligne de retour des boues	Solitax sc (HACH)	MES	Diffusion	SCADA
Effluent du décanteur secondaire	Turbidimètre VisoTurb® (Xylem)	Turbidité	Diffusion	mon*EAU*
	IQ SensorNet VARION (Xylem)	NH_4-N, NO_3-N, K, Cl et température	Électrodes d'Ion Sélective (EIS)	mon*EAU*
	pHmètre SensoLyt® 700 IQ (Xylem)	pH et température	Potentiométrie	mon*EAU*

3.1.2. kam*EAU*

Le projet de recherche kam*EAU* étudie le procédé KAMAK de la compagnie Bionest installé au sein d'un étang aéré (système de traitement des eaux usées) depuis septembre 2014 à Grandes-Piles. Les eaux usées alimentant ce système proviennent de la municipalité de Grandes-Piles (415 habitants en 2016). L'étang aéré d'origine est montré sur la Figure 20a. Le système KAMAK occupe le premier tiers de l'étang afin d'obtenir des charges de pollution élevées, c'est-à-dire trois fois la charge originale du système (Figure 20b, Figure 20c) (Patry et al., 2018). Plus en détail, la technologie KAMAK utilise un support inerte immergé et auto-supporté pouvant être déployé dans des lagunes aérées (Figure 21b). Sur ce support, un biofilm se développe et ainsi augmente la capacité de traitement biologique. La conception du KAMAK comprend deux zones de réacteurs à biofilm aérés, soit RX1 et RX2 (Figure 21c) ainsi que trois zones de clarification où les matières particulaires (particules se retrouvant dans les eaux usées ou le biofilm se détachant) sédimentent et s'accumulent (Figure 21d). Ces zones sont nommées CL1, CL2 et CL3 (Figure 21d). Enfin, la Figure 21e

représente le sens d'écoulement de l'eau au sein du système. Cette technologie a pour but de permettre d'augmenter de la capacité de traitement sans modifier le volume de l'étang.

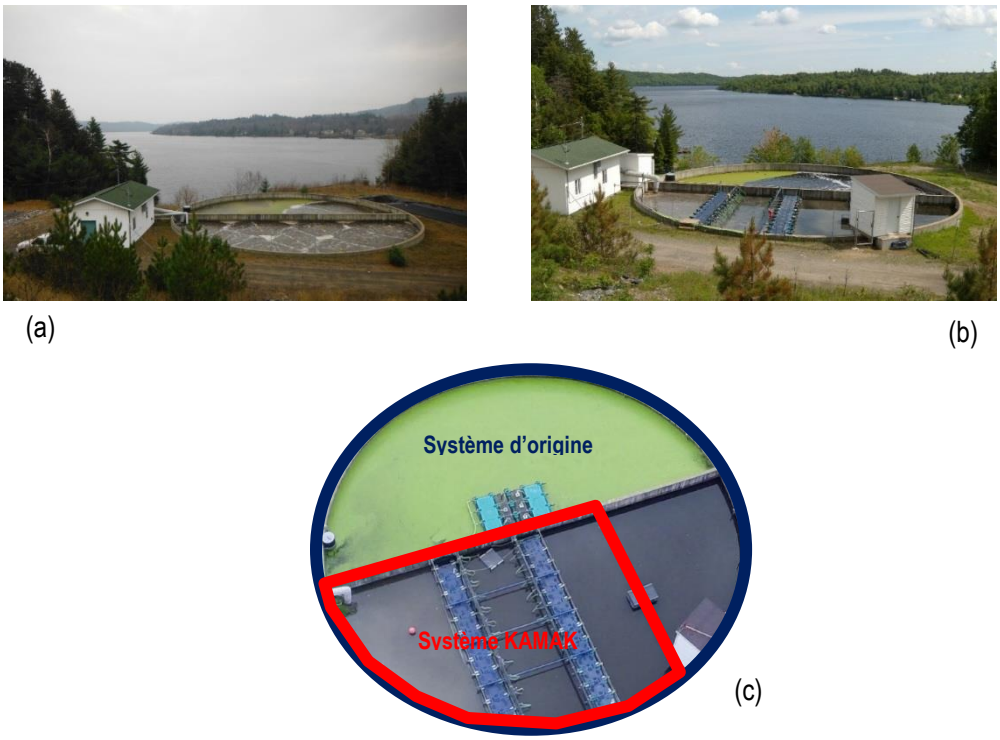


Figure 20. Étang aéré à Grandes-Piles (a) sans le système KAMAK (b) avec le système KAMAK (c) Vue du dessus du système (Patry et al., 2018)

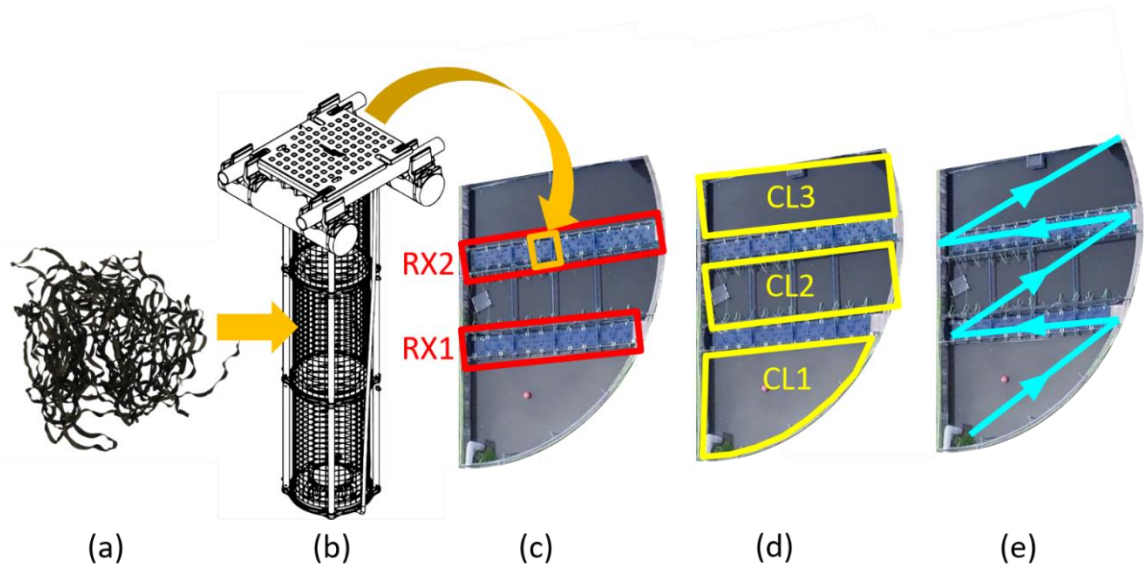


Figure 21. Système KAMAK avec (a) le média BIONEST, (b) cellule de 3 mètres de hauteur comportant le média (c) les zones des réacteurs biologiques RX1 et RX2, (d) les zones de décantation CL1, CL2 et CL3 (d) le sens d'écoulement au sein du système (Patry et al., 2018)

3.1.2.1. Capteurs et emplacements

Les capteurs installés dans le cadre de ce projet sont connectés à une station mon*EAU* similaire à celle présentée à la section 3.1.1.2. Le Tableau 6 résume l'ensemble des 11 capteurs présent au sein du système KAMAK. Comme pour l'usine pil*EAU*te, la grande diversité des capteurs mis en place permet de suivre en temps réel l'efficacité du processus, la compréhension du procédé et sa modélisation par la suite.

Tableau 6. Capteurs au sein du KAMAK connectés au système de surveillance monEAU

Localisation des capteurs	***Nom du capteur (fabricant)***	***Variables surveillées***	***Méthode de mesure***
Entrée	spectro::lyser (s::can)	DCO_{total}, $DCO_{soluble}$, MES, NO_3-N	Absorption UV-VIS spectrophotométrie
	ammo::lyser (s::can)	NH_4-N, K, pH et température	Électrodes d'Ion Sélective (EIS)
	pHmètre pHD sc Digital Differential (HACH)	pH et température	Potentiométrie
	Conductimètre 3700sc Digital (HACH)	Conductivité et température	Induction
RX1	Solitax sc (HACH)	MES	Diffusion
RX2	Solitax sc (HACH)	MES	Diffusion
Sortie	spectro::lyser (s::can)	DCO_{total}, $DCO_{soluble}$, MES, NO_3-N	Absorption UV-VIS spectrophotométrie
	ammo::lyser (s::can)	NH_4-N, K, pH et température	Électrodes d'Ion Sélective (EIS)
	pHmètre pHD sc Digital Differential (HACH)	pH et température	Potentiométrie
	Conductimètre 3700sc Digital (HACH)	Conductivité et température	Induction
	LDO (Luminescent Dissolved Oxygen) sensor (HACH)	Oxygène dissous, Température	Luminescence

3.1.3. bord*EAU*x

Le projet bord*EAU*x comme le nom l'indique se réalise à Bordeaux (France). Ce projet utilise la modélisation intégrée afin d'optimiser simultanément la gestion du réseau des égouts et de la StaRRE (Ledergerber et al., 2017, 2018). L'objectif du projet est d'élaborer et d'évaluer divers scénarios de gestion en vue de minimiser l'impact de la pollution sur le milieu naturel de la Garonne (Ledergerber et al., 2017, 2018). La Figure 22 montre l'ensemble des bassins versants se localisant au sein de la ville de Bordeaux. Pour ce projet, le bassin versant « Clos de Hilde » est étudié (Figure 23).

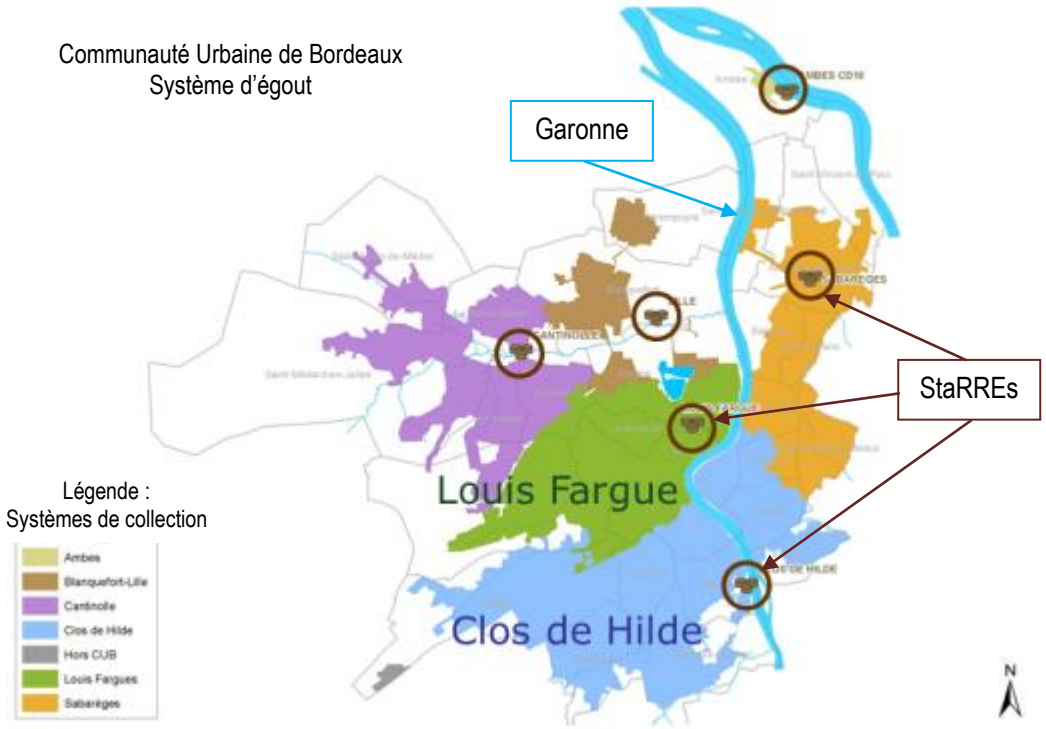


Figure 22. L'ensemble des bassins versants de la communauté urbaine de Bordeaux

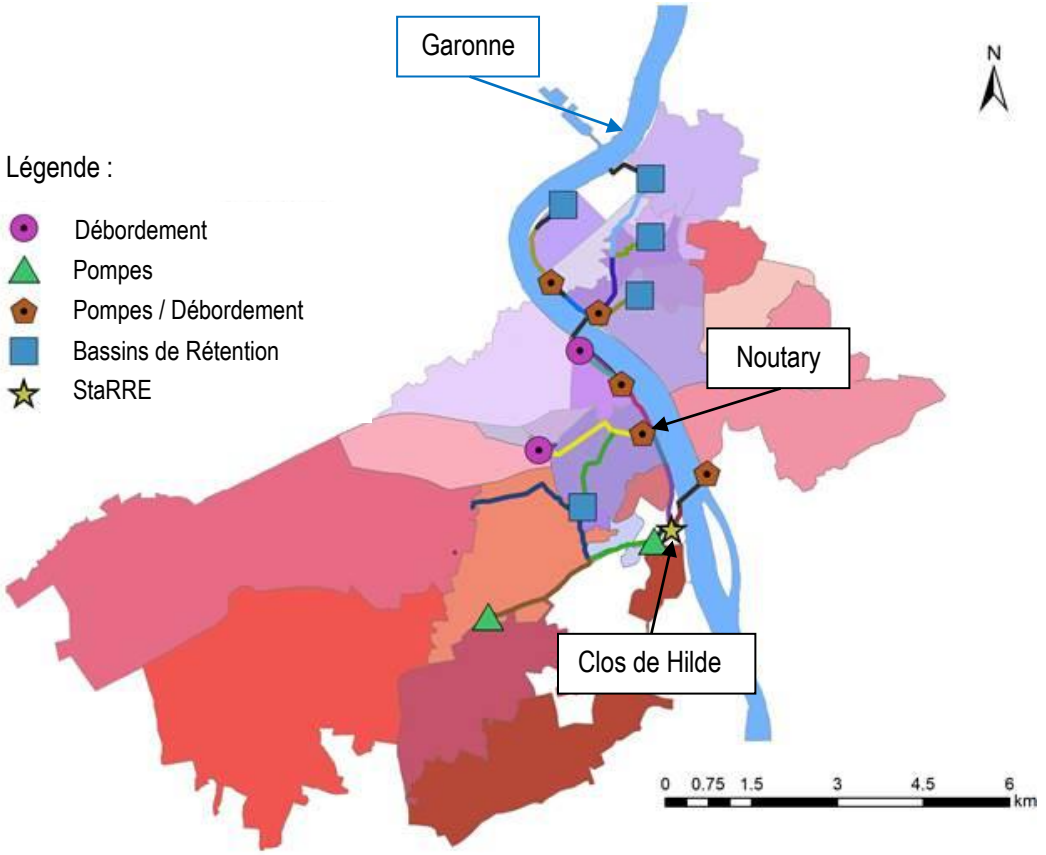


Figure 23. Bassin versant "Clos de Hilde" (Ledergerber et al., 2018)

Le bassin versant « Clos de Hilde », d'une superficie de 8000 ha, comporte un ensemble de structures tels que des bassins de rétention, des stations de pompage et des surverses (Figure 23). Il est aussi composé de systèmes d'égouts unitaires et séparatifs dont les eaux proviennent d'industries et de résidences. Sur la Figure 23, les deux localisations Noutary et Clos de Hilde sont les deux systèmes à l'étude dans le projet. L'un est au niveau d'un réseau d'égout (Noutary) et l'autre se situe au niveau de l'entrée de la StaRRE de Clos de Hilde (Figure 23).

3.1.3.1. Capteurs et emplacements

Dans le projet bord*EAU*x, les capteurs sont connectés à deux stations mon*EAU*. Le Tableau 7 résume l'ensemble des huit capteurs se localisant au sein du site d'étude de Bordeaux. Il y a une similarité des capteurs au niveau des deux points permettent de faciliter leur maintenance (nettoyages, calibrations) en réduisant le nombre de méthodes de mesure et de calibration.

Tableau 7. Capteurs connectés au système de surveillance monEAU au sein des deux points de mesures (Noutary et Clos de Hilde) à Bordeaux

Localisation des capteurs	***Nom du capteur (Fabricant)***	***Variables surveillées***	***Méthode de mesure***
Noutary	spectro::lyser (s::can)	DCO_{total}, $DCO_{soluble}$, TSS, NO_3-N	Absorption UV-VIS spectrophotométrie
	Turbidimètre VisoTurb® (Xylem)	Turbidité	Néphélométrie
	Conductimètre TetraCon® 700 IQ (Xylem)	Conductivité et température	Induction
	pH-mètre SensoLyt® 700 IQ (Xylem)	pH et température	Potentiométrie
Clos de Hilde	spectro::lyser (s::can)	DCO_{total}, $DCO_{soluble}$, TSS, NO_3-N	Absorption UV-VIS spectrophotométrie
	Turbidimètre VisoTurb® (Xylem)	Turbidité	Néphélométrie
	Conductimètre TetraCon® 700 IQ (Xylem)	Conductivité et température	Induction
	pH-mètre SensoLyt® 700 IQ (Xylem)	pH et température	Potentiométrie

3.2. Méthodes de traitement de données

Les deux méthodes de traitement de données étudiées dans ce projet sont :

- La méthode univariée
- La méthode multivariée (ACP)

Elles ont été développées par Alferes et al. (2013a). Divers articles et mémoires exposent ces méthodes (Alferes and Vanrolleghem, 2016; Alferes et al., 2012, 2013b, 2013a; Plana, 2015; Saberi, 2015).

3.2.1. Méthode univariée

La méthode univariée se classe dans les méthodes d'extraction basiques d'information (Corominas et al., 2018) (section 1.4.1). Elle a pour objectif de détecter les erreurs, les données aberrantes ainsi que les fautes au sein des séries de données. Plus précisément, la méthode mise en place par Alferes et al. (2012) est basée sur deux grandes étapes :

- Filtrage des données
- Détection des fautes

3.2.1.1. Filtrage des données

Au sein de cette première étape, deux sous-parties sont effectuées dans le but de filtrer des données :

- Détection des données aberrantes
- Lissage des données sans les données aberrantes

a. Détection des données aberrantes

Dans certaines situations, au sein de séries de données, les mesures prennent de très faibles ou de très fortes valeurs dues par exemple à un nettoyage du capteur ou une interférence (Alferes & Vanrolleghem, 2016). Elles sont aberrantes parce qu'elles ne représentent pas le phénomène que l'on veut mesurer. D'après Judd et al., (2018), ces données sont aberrantes si pour une raison ou une autre, elles ne font pas partie du même lot que les autres observations. Une méthode basée sur la prédiction de la valeur attendue et la comparaison avec la valeur mesurée permet de les détecter. En général, les systèmes de prédiction sont robustes, précis, rapides et simples à mettre en œuvre (Alferes & Vanrolleghem, 2016). Alferes et al., (2012) ont utilisé une exponentielle statistique lissée de premier, deuxième et troisième ordre fréquemment employée dans le domaine de la finance et l'économie. Les Équation 7, Équation 8 et Équation 9 représentent les trois exponentielles statistiques (S_T, $S_T^{[2]}$ et $S_T^{[3]}$).

$$S_T = \alpha x_T + (1-\alpha) S_{T-1} \qquad \textit{Équation 7}$$

$$S_T^{[2]} = \alpha x_T + (1-\alpha) S_{T-1}^{2} \qquad \textit{Équation 8}$$

$$S_T^{[3]} = \alpha x_T + (1-\alpha) S_{T-1}^{3} \qquad \textit{Équation 9}$$

Avec

- x_T : la valeur actuelle de la donnée

- S_{T-1} : la valeur estimée ou prédite pour le temps précédent T-1
- α : la constante de lissage

La donnée prédite à un temps T+1 est alors calculée à l'aide du modèle de lissage de troisième ordre. L'Équation 10 donne la formule théorique de la valeur prédite calculée au temps T+1 :

$$\hat{x}_{T+1} = \hat{a}_T + \hat{b}_T + \frac{1}{2}\hat{c}_T \qquad \textit{Équation 10}$$

Avec :

- $\hat{a}_T, \hat{b}_T$ et $\hat{c}_T$: les coefficients du modèle se calculent à l'aide des trois exponentielles statistiques (S_T, $S_T^{[2]}$et $S_T^{[3]}$) et des équations :

$$\hat{a}_T = 3S_T - 3S_T^{[2]} + S_T^{[3]} \qquad \textit{Équation 11}$$

$$\hat{b}_T = \frac{\alpha}{2(\alpha-1)^2}[(6-5\alpha)S_T - 2(5-4\alpha)S_T^{[2]} + (4-3\alpha)S_T^{[3]}] \qquad \textit{Équation 12}$$

$$\hat{c}_T = (\frac{\alpha}{\alpha-1})^2(S_T - 2S_T^{[2]} + S_T^{[3]}) \qquad \textit{Équation 13}$$

Enfin, des limites minimales et maximales autour de la valeur prédite $\hat{x}_{T+1}$ sont calculées :

- Limite haute :

$$x \lim_T U = \hat{x}_{T+1} + K \times \hat{\sigma}_{e,T+1} \qquad \textit{Équation 14}$$

- Limite basse :

$$x \lim_T L = \hat{x}_{T+1} - K \times \hat{\sigma}_{e,T+1} \qquad \textit{Équation 15}$$

Avec :

K : la constante proportionnelle afin d'ajuster les limites plus ou moins restrictif

$\hat{\sigma}_{e,T+1} = 1.25 \times \hat{\Delta}_T$: l'erreur sur la prédiction

$\hat{\Delta}_T = \beta|e_T(1)| + (1-\beta) \times \hat{\Delta}_{T-1}$: l'estimation de la déviation standard

Avec

$e_T(1) = x_T - \hat{x}_T$: l'erreur de la prédiction

Avec

$\hat{x}_T$: la valeur prédite

Ces deux limites permettent de calculer un intervalle de prédiction. Si à un instant T+1, une donnée est en dehors de cet intervalle, la donnée est alors appelée une valeur aberrante. Elle sera remplacée par sa valeur prédite ($\hat{x}_{T+1}$). En fin de processus, une nouvelle série de données est créée. Elle est composée des données acceptées incluant les données remplacées.

La Figure 24 représente la méthode avec un exemple théorique de détection d'une donnée aberrante en utilisant la méthode exposée auparavant.

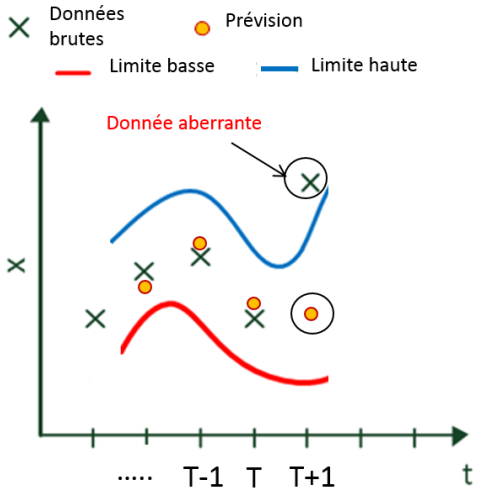


Figure 24. Exemple théorique de détection des données aberrantes et leurs remplacement par la prédiction (Alferes et al., 2012)

b. Lissage des données sans les données aberrantes

Le lissage des données après la détection des données aberrantes est réalisée en utilisant un estimateur Kernel Nadadya-Watson, aussi appelé lissage par moyenne mobile pondérée (Alferes and Vanrolleghem, 2016; Plana, 2015; Saberi, 2015). Les Équation 16 et Équation 17 résument le principe de lissage par la moyenne mobile pondérée :

$$\hat{y}_h(x_0) = \sum_{i=1}^{n} W(x_0, x_i; h) \times y(x_i) \qquad \textit{Équation 16}$$

Avec :

$\hat{y}_h(x_0)$: la moyenne mobile pondérée d'un point observé à x_0

n : le nombre de points considérés

h : le paramètre pour calculer la bande passante du filtre

$y(x_i)$: les observations à un points x_i

$W(x_0, x_i; h)$: la fonction poids

$$W(x_0, x_i; h) = \frac{K\left(\frac{x_0 - x_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{x_0 - x_i}{h}\right)} \qquad \textit{Équation 17}$$

Avec :

K : la fonction noyau

Dans la littérature, Alferes et Vanrolleghem (2016) proposent l'utilisation d'une fonction « Gaussian kernel » pour le calcul de K, présentée par l'Équation 18 (Alferes & Vanrolleghem, 2016; Plana, 2015).

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \qquad \textit{Équation 18}$$

Avec :

$x = \frac{x_0 - x_i}{h}$

Finalement, la Figure 25 représente un exemple de données filtrées en détectant les données aberrantes et en lissant le signal (courbe en verte).

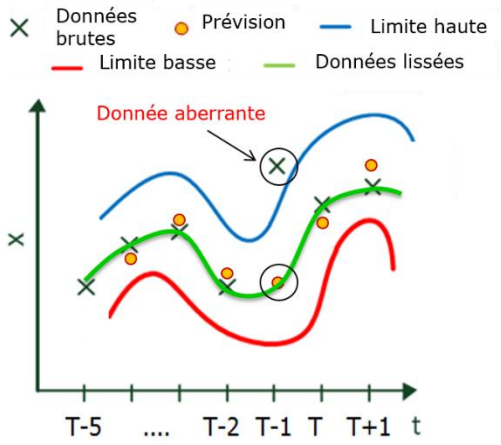


Figure 25. Exemple théorique de le filtrage de données (Alferes et al., 2012)

3.2.1.2. Détection des fautes

La deuxième étape de la méthode, la détection des fautes, a pour but de détecter les biais, les dérives, les défaillances de capteurs ainsi que la dégradation de la précision. Ces différents types de fautes ont été exposés dans la revue de la littérature à la section 1.3 « Fautes et problèmes des capteurs ». Cette sous-étape se divise en deux parties:

- Calcul d'indicateurs de défaillances et leurs limites
- Obtention des données acceptées

c. Calcul d'indicateurs de défaillances et leurs limites

La méthode permet de déterminer quatre indicateurs de défaillances dans le but de détecter les fautes et nettoyer les séries de données. La détermination de ces quatre indicateurs de défaillances est réalisée en analysant les données acceptées et lissées, obtenues dans l'étape de le filtrage des données à chaque pas de temps. Les quatre indicateurs de défaillances sont présentés ci-dessous avec leurs équations correspondantes :

1) **Sign Run-test :** Cet indicateur de défaillances a pour but de vérifier si le bruit de mesure est distribué aléatoirement autour de la série de données (Dochain and Vanrolleghem, 2001). Pour ce test, deux équations sont utilisées. Premièrement, le signe de la différence entre les données acceptées et lissées est évalué (Équation 19) et une série de signes est obtenue. La série de données a un bruit de mesure aléatoire si la séquence de signes respecte certaines caractéristiques.

$$\text{Signe}\,(\text{données acceptées} - \text{données lissées}) \qquad \textit{Équation 19}$$

Deuxièmement, l'indicateur de défaillances Q_{corr} du Sign Run-test et calculé avec la série de signes obtenue et de Équation 20 :

$$Q_{corr} = \frac{R - \frac{N}{2}}{\sqrt{\frac{N}{2}}} \qquad \text{Équation 20}$$

Avec:

- R : le nombre de changement de signes dans la série de signes
- N : le nombre de données dans la fenêtre sélectionnée

La Figure 26 montre un exemple de faute détectée. Dans les premiers temps, les données sont constantes à cause d'une défaillance complète du capteur. Ainsi, le signe Run-Test va permettre de détecter cet intervalle de temps dû à un changement de signes constant.

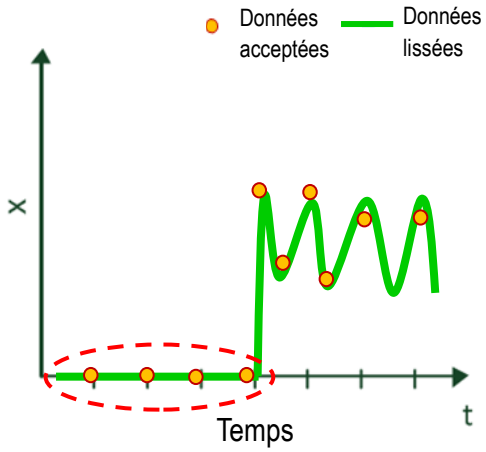


Figure 26. Exemple de faute détectée par l'indicateur de défaillances « signe run-test »

2) **Pente :** Cet indicateur de défaillances informe sur la dynamique des données et aide à la détection des variations soudaines dans les séries de données. Mathématiquement, cet indicateur de défaillances est déterminé à partir de l'équation suivante :

$$\frac{dx}{dt} \cong \frac{((Donnée\ lissée)_i - (Donnée\ lissée)_{i-1})}{dt} \qquad \text{Équation 21}$$

Si la pente est trop élevée par rapport à la variation maximale que la variable est censée donner, la mesure sera considérée comme défectueuse. En termes d'exemple, la Figure 27 montre deux changements soudains des données lissées pouvant être détectés par cet indicateur de défaillances. Ces deux fautes peuvent être dues à un colmatage du capteur.

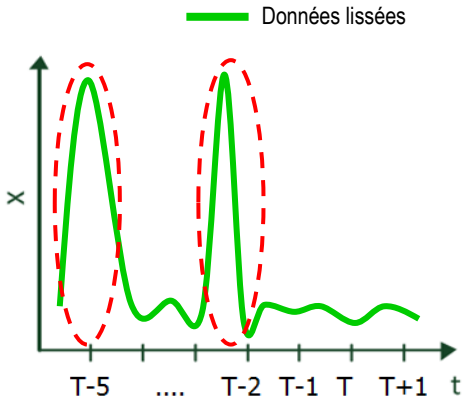


Figure 27. Exemple de faute détectée par l'indicateur de défaillances « pente »

3) **Déviation Standard :** Cet indicateur de défaillances se base sur le niveau du bruit de mesure estimé. Une importante déviation peut indiquer la présence d'une faute. Sur la Figure 28, deux déviations importantes sont notées en début et au milieu de la série de données et ainsi indiquer une faute de capteur.

$$\mathrm{dst} \cong \frac{\sum(x_i - \hat{x}_c)^2}{N-1} \qquad \textit{Équation 22}$$

Avec

x_i : valeur de la donnée à un temps i

$\hat{x}_c$: la moyenne des données dans la fenêtre

N : nombre de données dans la fenêtre sélectionnée

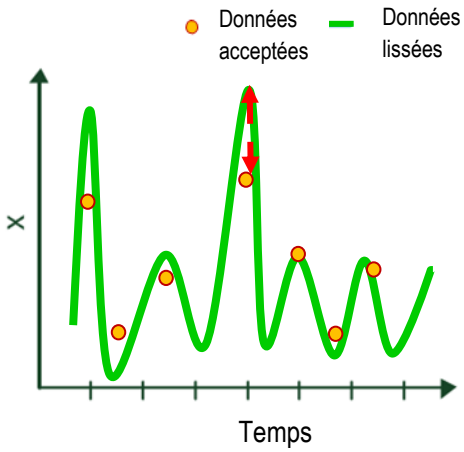


Figure 28. Exemple de faute détectée par l'indicateur de défaillances « déviation standard »

4) **Plage des valeurs :** cet indicateur de défaillances évalue si les données sont hors des étendues de valeurs réalistes. Connaissant le procédé, des limites minimum et maximum permettent de vérifier si

les données sont à l'intérieur de cette plage (Équation 23). La Figure 29 montre un exemple de détection d'une faute par cet indicateur de défaillances où une partie des données lissées sont au-dessus de la limite maximum.

$$x_i > Max \; ou \; x_i < Min \qquad \text{Équation 23}$$

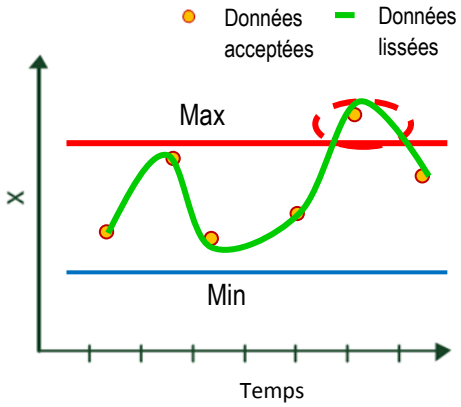


Figure 29. Exemple de faute détectée par l'indicateur de défaillances « plage »

Enfin, des limites minimum et maximum à ces indicateurs de défaillances peuvent être déterminées afin d'obtenir les données nettoyées. Ce choix de limites est effectué par l'utilisateur de la méthode, à l'aide d'une aide graphique, de la connaissance du procédé et des événements passés documentés. Dans la section résultats et discussions, plusieurs séries de données de trois études de cas ont été traitées en donnant à chaque fois les limites choisies. Cela aidera l'utilisateur futur dans ses choix.

d. Obtention des données acceptées et rejetées

L'étape finale de la méthode est la détermination de données acceptées et rejetées. Pour ce faire, les limites minimum et maximum des quatre indicateurs de défaillances permettent de déterminer les données acceptées et rejetées dans les séries de données. Ces dernières se définissent comme :

- **Donnée acceptée :** Donnée n'étant pas une donnée aberrante et ayant respectée l'ensemble des quatre indicateurs de défaillances.
- **Donnée rejetée :** Donnée étant une donnée aberrante ou n'ayant pas respectée un ou plusieurs des quatre indicateurs de défaillances.

3.2.2. Méthode multivariée

La seconde méthode étudiée au sein de ce projet est l'Analyse en Composantes Principales (ACP) qui est une méthode multivariée (Montgomery, 2009). Cette méthode se situe dans la deuxième catégorie de l'extraction avancée d'informations avec la réduction de la dimension (section 1.4.2). Son objectif est d'exploiter la redondance d'informations présente dans des variables fortement corrélées afin de réduire le nombre de dimensions tout en gardant un maximum d'informations des données originales. Dans le domaine de la qualité des données, l'ACP permet l'analyse de variables multiples. Elle utilise des données collectées dites « normales » pour créer un modèle ACP. Par la suite, deux tests statistiques permettront d'évaluer la concordance, à l'intérieur de limites, de nouvelles séries de données dans le but de détecter les fautes (Lee and Vanrolleghem, 2004; Rosen and Lennox, 2001; Villez, 2008). Cette méthode développée par Alferes et al. (2013b) est composée de deux grandes étapes :

- Le développement du modèle ACP
- La détection des fautes

3.2.2.1. Développement du modèle ACP

Le but de cette étape est de construire un modèle ACP. Les sous-étapes réalisées sont :

- Choix des données
- Normalisation de ces données
- Création du modèle ACP
- Calcul des tests statistiques Q et T^2 et leurs limites

a. Choix de données normales

Les données utilisées pour construire le modèle d'ACP doivent représenter au mieux le corportement du système (Alferes et al., 2013b). L'ensemble des données brutes d'une ACP est représenté sous forme matricielle. L'Équation 24 donne la matrice de données brutes d'une ACP :

$$\mathrm{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,j} & \dots & x_{1,M} \\ x_{2,1} & x_{2,2} & \dots & x_{2,j} & \dots & x_{2,M} \\ \vdots & \vdots & & \vdots & & \vdots \\ x_{i,1} & x_{i,2} & \dots & x_{i,j} & \dots & x_{i,M} \\ \vdots & \vdots & & \vdots & & \vdots \\ x_{N,1} & x_{N,2} & \dots & x_{N,j} & \dots & x_{N,M} \end{bmatrix} \qquad \textit{Équation 24}$$

Avec :

- X : la matrice théorique des données brutes
- N : le nombre de valeurs par variable
- M : le nombre de variables

b. Normalisation des données

La normalisation des données ou plus précisément le centrage-réduction permet de contourner les problèmes d'ordre de grandeur tels que les différences d'unités entre les variables. Le résultat de l'étape montre une indépendance aux unités des variables.

Le centrage-réduction s'opère en soustrayant la moyenne de la variable de chaque valeur $x_{i,j}$ et en divisant chaque valeur centrée par l'écart-type de la variable. Les variables centrées et réduites ont donc une moyenne de zéro et un écart type de 1.

L'Équation 25 représente la variable centrée-réduite associée à la valeur $x_{i,j}$:

$$Z_{ij} = \frac{(x_{ij} - \overline{x_j})}{\sigma_j} \qquad \textit{Équation 25}$$

Avec :

x_{ij} : la valeur brute

$\overline{x_j}$: la moyenne de chaque variable j

σ_j : l'écart type de chaque variable j

c. Création du modèle ACP

La création du modèle ACP permet d'oobtenir les composantes principales. **Une composante principale (CP)** se définit par une combinaison linéaire des variables initiales qui conserve un maximum de variance. Premièrement, la matrice de covariance Cx est calculée à l'aide de l'équation suivante (Johnson and Wichern, 2002) :

$$\mathrm{Cx} = \frac{1}{\mathrm{N}-1}\mathrm{Z}^T\mathrm{Z} \qquad \textit{Équation 26}$$

Avec :

Cx : la matrice de covariance

N : le nombre d'échantillons

Z' : la matrice centrée-réduite transposée

Z : la matrice centrée-réduite

Deuxièmement, de cette matrice Cx, une décomposition en valeurs singulières est effectuée à partir de l'Équation 27 (Johnson and Wichern, 2002) :

$$\mathrm{Cx} = \mathrm{V}\Lambda\mathrm{V}^{\mathrm{T}} \qquad \textit{Équation 27}$$

Avec :

V: les vecteurs propres de Cx

V^{T} : la matrice transposée de V

Λ : la matrice diagonale des valeurs propres de Cx triées par ordre décroissant ($\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_M \geq 0$)

Troisièmement, une matrice de transformation P [M x a] est créée en choisissant « a » vecteurs propres (colonnes de V correspondant à « a » valeurs principales propres) (Alferes et al., 2013b; García et al., 2010;

Mirin and Wahab, 2014). L'indice « a » représente le nombre de composantes principales de la matrice V. Le choix de la valeur « a » est effectué en utilisant la méthode « eigenvalue scree plot » proposée par Jollife (2002). Lorsque la somme des pourcentages de variance (Axe vertical) de «a» premières composantes approche une valeur choisie par l'utilisateur la valeur de « a » est égale au nombre de composantes principales (Axe horizontal) sélectionnées. La Figure 30 montre un exemple de diagramme « eigenvalue scree plot ».

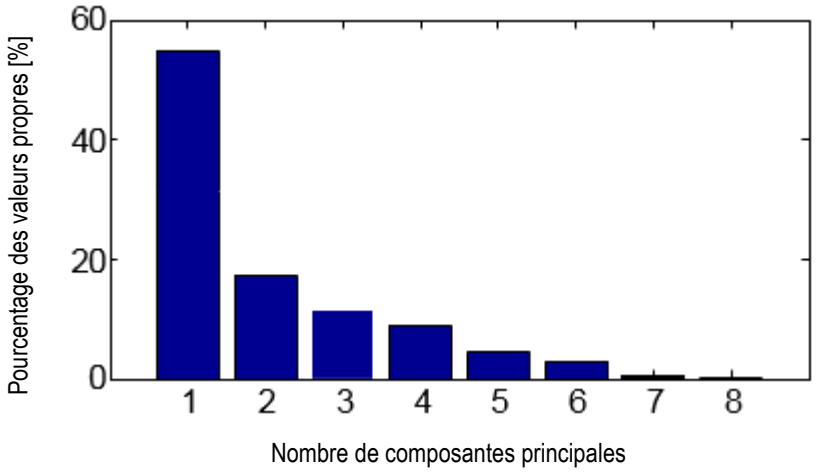


Figure 30. Exemple d'un diagramme « eigenvalue scree plot » d'après Alferes et al. (2012)

Sur la Figure 30, la somme de pourcentage de la variance que l'on désire conserver est de 90 % (choix de l'utilisateur) ce qui correspond à un nombre de CP égal à quatre.

Ainsi, le modèle ACP est obtenu à partir de l'équation suivante :

$$\overline{\mathrm{X}} = \mathrm{TP}^{\mathrm{T}} + \mathrm{E} \qquad \textit{Équation 28}$$

Avec :

$\overline{\mathrm{X}}$: la matrice du modèle centrée-réduite

P : la matrice de chargement

E : la matrice résiduelle

T : la matrice des scores de l'ACP. Elle est équivalente aux données originales dans le nouveau système de coordonnées. Cette matrice se détermine à partir de l'équation :

$$T = \mathrm{XP} \qquad \textit{Équation 29}$$

Ayant effectué l'ensemble des transformations matricielles ci-dessus, un graphique multidimensionnel est obtenu où l'information est réorganisée dans ses composantes principales. La Figure 31 montre un exemple de la réorganisation des données dans ses deux composantes principales. Chaque variable est représentée dans le nouvel espace de coordonnées (PC1, PC2) par un vecteur (une certaine longueur et direction). Cette longueur et cette direction indique la contribution de la variable aux deux composantes PC1 et PC2. A titre d'exemple, la variable pH1 a une plus forte contribution à PC2 qu'à PC1 à l'inverse de la variable

Cond1 (Figure 31). Chaque point (rouge) représente une donnée originale dans le nouveau système de coordonnées.

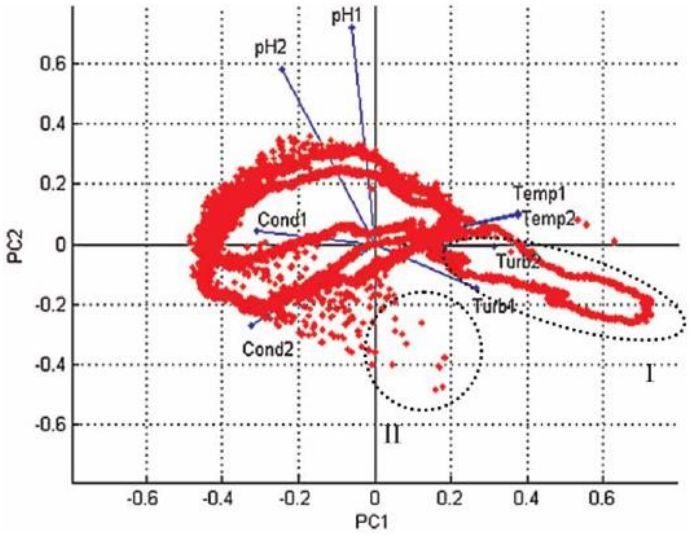


Figure 31. Exemple de réorganisation de l'information dans ses composantes (Alferes et al., 2013b)

d. Calcul des tests statistiques Q et T^2 et leurs limites

L'analyse de l'ACP se fait à l'aide de deux tests statistiques : le test T² d'Hotelling et le test statistique de l'erreur quadratique moyenne Q (Alferes et al., 2013b; García et al., 2010; Mirin and Wahab, 2014; Montgomery, 2009). La mise en place de ces tests et leurs limites s'effectuent automatiquement.

Tout d'abord, le test T² d'Hotelling représente la variation majeure dans les données (Garcia-Alvarez, 2009). Cette variation peut être due à un colmatage d'un capteur et donc une augmentation soudaine de la valeur de la variable à ce moment-là. Dans un espace géométrique, tel que montré en Figure 32, ce test se définit comme la distance entre un point et le centre du plan représentant le modèle ACP (cercle bleu).

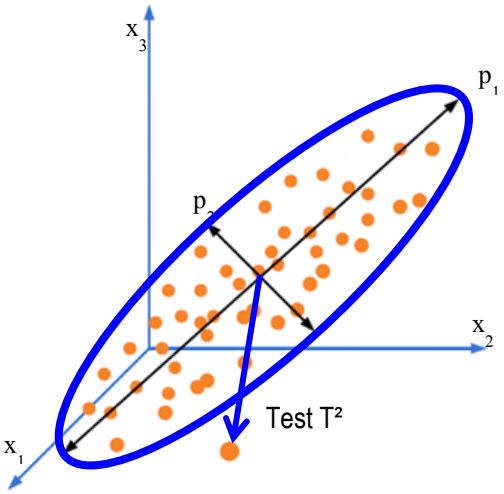


Figure 32. Représentation graphique du test T² (Montgomery, 2009)

En niveau mathématique, la valeur du T^2 peut être calculée à partir de l'équation suivante pour chaque point du vecteur Z (Johnson and Wichern, 2002) :

$$T^2 = Z^T P \Lambda_a^{-1} P^T Z \qquad \textit{Équation 30}$$

Avec :

Z^T : la transposée de la matrice centrée-réduite

P : la matrice de chargement

Λ_a^{-1} : la matrice diagonale contenant les « a » valeurs propres associées avec les « a » vecteurs propres (CP retenues)

Z : la matrice centrée-réduite

Ensuite, le test statistique de l'erreur quadratique moyenne Q quantifie le bruit aléatoire dans les séries de données (Garcia-Alvarez, 2009). Au niveau géométrique, le test Q se définit comme la distance perpendiculaire entre un point et le plan (cercle bleu) (Figure 33).

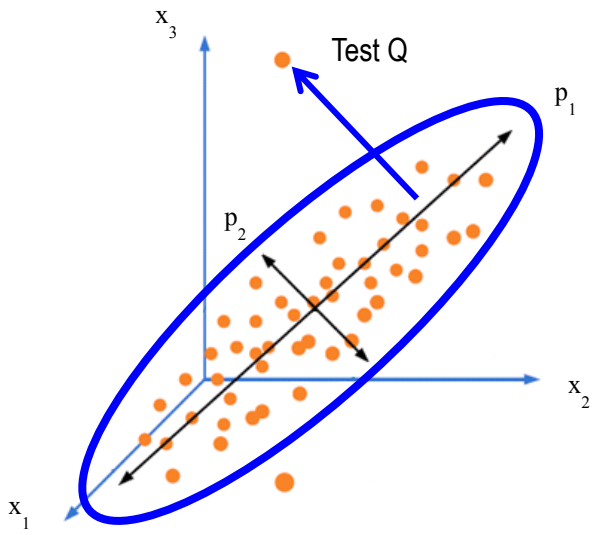


Figure 33. Représentation graphique du test Q (Montgomery, 2009)

Le test est déterminé mathématiquement par l'équation suivante pour chaque point du vecteur Z (Johnson & Wichern, 2002):

$$Q = Z^T \left(I - PP^T\right) \times Z \qquad \textit{Équation 31}$$

Avec :

I : la matrice d'identité

Enfin, ayant effectué les deux tests pour une série de données normales, leur limite haute est déterminée afin d'analyser la normalité de nouvelles données. Des fautes éventuelles de capteurs peuvent y être détectées. Premièrement, la limite haute pour le test T² est évaluée par l'Équation 32 :

$$T_\alpha^2 = \frac{(N-1)\times(N+1)\times C}{N\times(N-C)} F(\alpha, N-C) \qquad \textit{Équation 32}$$

Avec :

N : le nombre de variables

C : le nombre de composantes principales choisies

F : la distribution de Fisher-Snedecor

α : le niveau de signification

Deuxièmement, la limite haute pour le test Q est donnée par l'Équation 33 (Jackson and Mudholkar, 1979):

$$Q_\alpha = \theta_1 \times [t_\alpha \times \frac{\sqrt{2\times\theta_2\times h_0^2}}{\theta_1} + 1 + \frac{\theta_2 \times h_0 \times (h_0-1)}{\theta_1^2}]^{\frac{1}{h_0}} \qquad \textit{Équation 33}$$

Avec :

t_α : le percentile supérieur pour une distribution standard normale N(0,1) et une significativité α

$h_0 = 1 - \frac{2\times\theta_1\times\theta_3}{3\times\theta_2^2}$

$\theta_i = \sum_{j=a+1}^{M} \lambda_j^i$ avec i=1,2,3

Avec :

a : le nombre de composantes principales choisies

M : le nombre de variable

λ_j : la j-ième valeur propre

3.2.2.2. Détection des fautes

La deuxième grande étape de la méthode est la détection de fautes au sein des nouvelles séries de données. Plusieurs étapes doivent être effectuées :

- Le choix de données à traiter
- La normalisation des données à traiter
- La projection sur le modèle ACP
- L'évaluation des tests statistiques T² et Q

a. Choix de données à traiter

La personne utilisant cette méthode choisit des données qu'elle veut traiter. Point important, l'ensemble des variables choisies doit rester le même que l'ensemble des variables utilisées pour la construction du modèle ACP.

b. Normalisation des données à traiter

Les données doivent être centrées et réduites tel qu'expliqué précédemment (Normalisation des données).

c. Projection sur le modèle ACP

La projection des nouvelles données sur le modèle ACP est effectuée en utilisant l'Équation 34 :

$$T = Z_{new} \times P \qquad \textit{Équation 34}$$

Avec :

T : la matrice des scores des données à traiter

Z_{new} : la matrice centrée-réduite des données à traiter

P : la matrice de chargement du modèle ACP

d. Évaluation des tests T^2 et Q

Dans cette dernière étape, les tests statistiques T² et Q exposés précédemment sont appliqués aux données à traiter. Lors du développement du modèle ACP, des limites maximales à ces tests ont été calculées. En prenant en compte ces limites, une détection des fautes peut être réalisée en observant les périodes où les tests T² et Q sont au-dessus des limites :

$$T^2_{new} > T^2_{lim}$$
$$Q_{new} > Q_{lim}$$

Chapitre 4 Résultats et discussion sur la qualité des données

Ce chapitre présente et discute les résultats obtenus sur la qualité des données. La Figure 36 présente le diagramme général de la séquence de traitement de données pour en assurer la qualité des données et les principales activités effectuées à chaque étape :

- Réactions des capteurs
- Traitement des données
- Validations des données

4.1. Réaction des capteurs face à deux problèmes courants

Cette section expose la réaction des capteurs face à deux problèmes courants tels que le nettoyage des capteurs et l'effet à une forte charge d'un constituant.

4.1.1. Nettoyage des capteurs

La maintenance et le nettoyage des capteurs sont des étapes essentielles pour récolter des données de qualité en continue et pour maximiser leur durée de vie. Plana (2015) expose dans son mémoire une méthode d'évaluation de l'effet du nettoyage qui part du principe qu'il ne devrait pas être visible dans les séries de données. Ainsi, les données obtenues entre deux nettoyages peuvent être considérées valides. L'annexe A (Évaluation de l'effet du nettoyage d'après Plana (2015)) présente cette méthode en s'inspirant du diagramme de contrôle développé dans le chapitre 1 (section 1.5.1). Les prochaines trois sous-parties montrent l'influence du nettoyage sur les séries de données pour les projets pil*EAU*te, bord*EAU*x et kam*EAU* où des nettoyages étaient réalisés une à deux fois par semaine.

4.1.1.1. pilEAUte

Un nettoyage hebdomadaire est effectué pour l'ensemble des 14 capteurs installés dans la station. Ce dernier a, à certains moments, des effets sur la qualité des données. La Figure 34 expose un exemple typique de biais entre les données du capteur avant et après nettoyage de ce dernier. Il résulte du fait que le capteur devrait être nettoyé plus régulièrement. Ainsi, les données précédant le nettoyage ne peuvent pas être utiles (données non valides) puisqu'un biais important est mis en évidence par le nettoyage. Un mauvais nettoyage ou non régulier peut aussi entrainer une dérive des mesures. La Figure 35 montre une dérive continuelle des données due à un colmatage graduel commencé quelques jours après le premier nettoyage (la première ligne verte). Celui-ci a eu pour effet un biais. Le deuxième nettoyage devait rétablir la situation mais il n'a pas eu l'effet escompté à cause d'une mauvaise procédure de nettoyage. Ainsi, la dérive a continué jusqu'au troisième nettoyage où un biais important est observable (Figure 35). La conclusion pouvant être apportée ici, est le rejet de l'ensemble des données entre le premier et le troisième nettoyage.

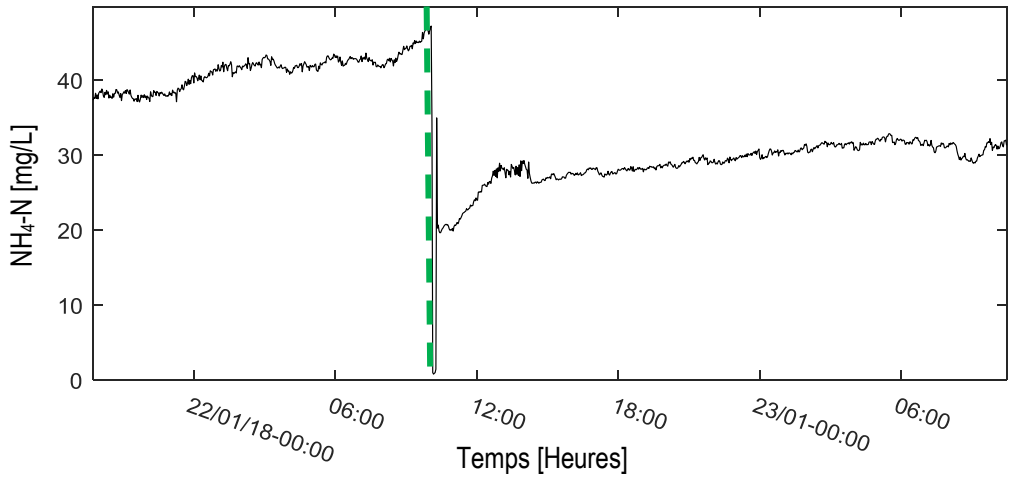


Figure 34. Nettoyage illustrant un biais entre les données de NH_4-N (ammo::lyser, affluent pilEAUte) captées avant et après le nettoyage (ligne pointillée verte)

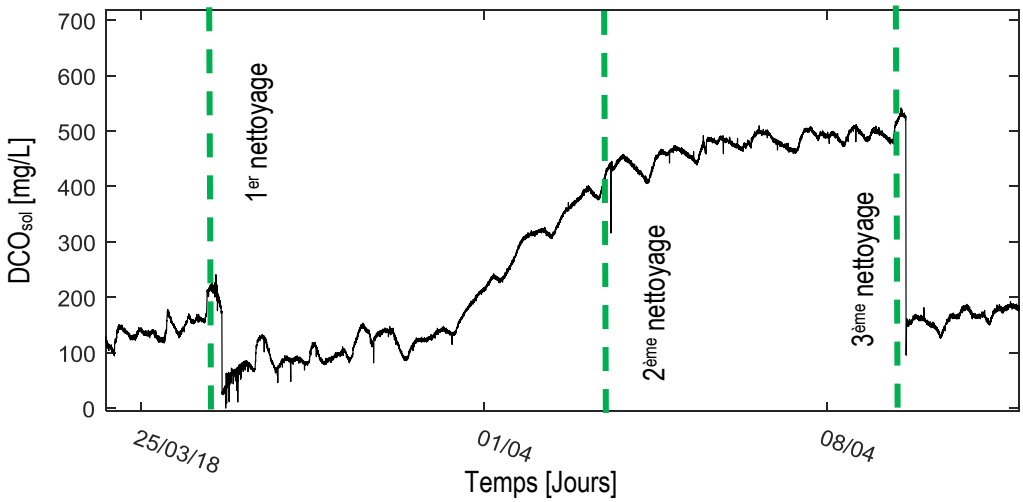


Figure 35. Dérive continuelle des données de DCO soluble (spectro::lyser, affluent du pilEAUte) après un mauvais nettoyage (deuxième trait pointillé)

Cependant, quand un capteur est nettoyé à temps et de la bonne façon, aucune conséquence n'est observable entre l'avant et l'après nettoyage. La Figure 37 est un exemple de nettoyage d'un capteur sans effet sur les données (cadre vert en pointillés). Il n'est pas observé de biais entre l'avant et l'après nettoyage (Figure 37). Ainsi, les données avant le nettoyage peuvent être considérées fiables.

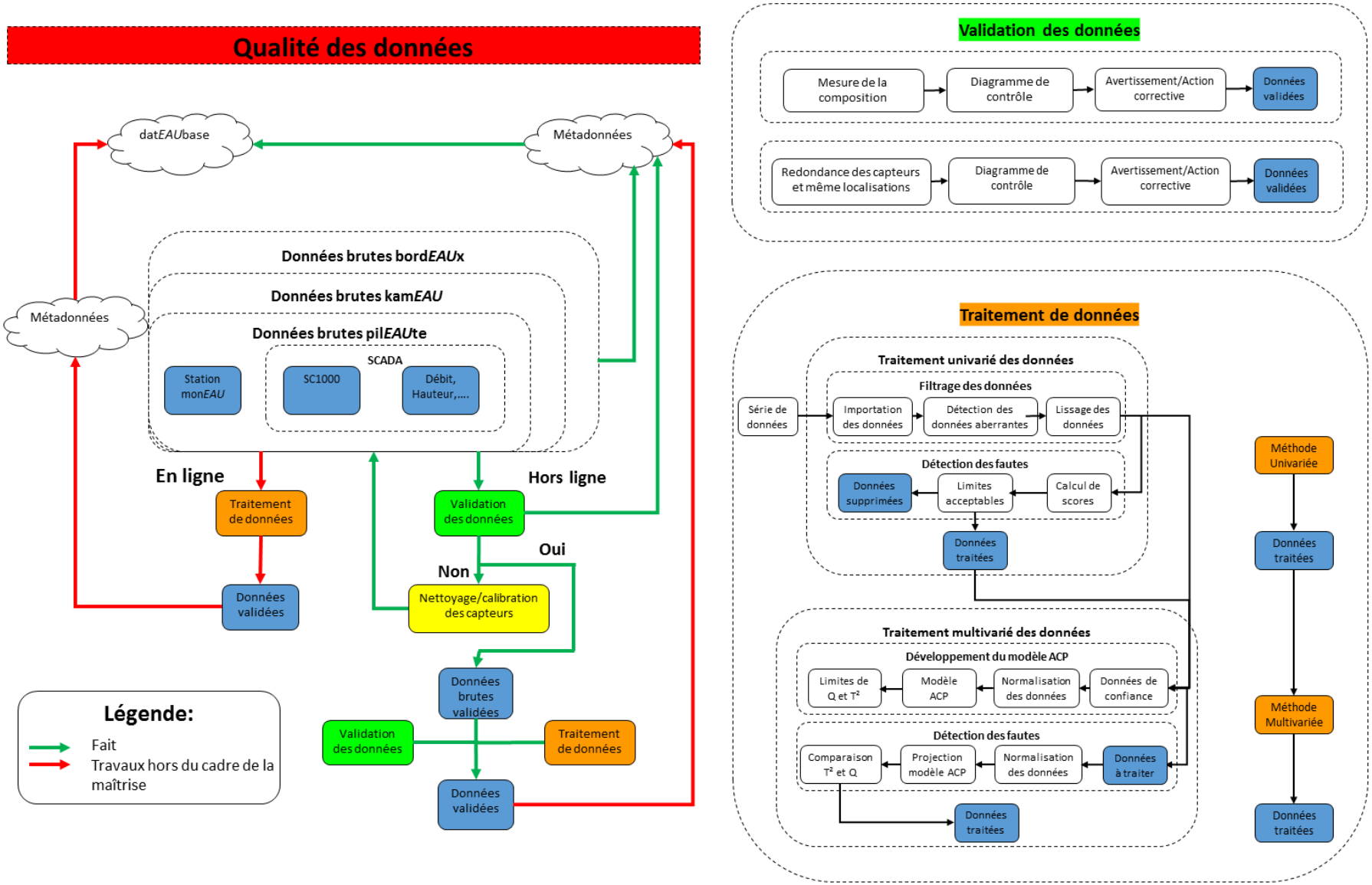


Figure 36. Diagramme général de la séquence de traitement de données pour en assurer la qualité.

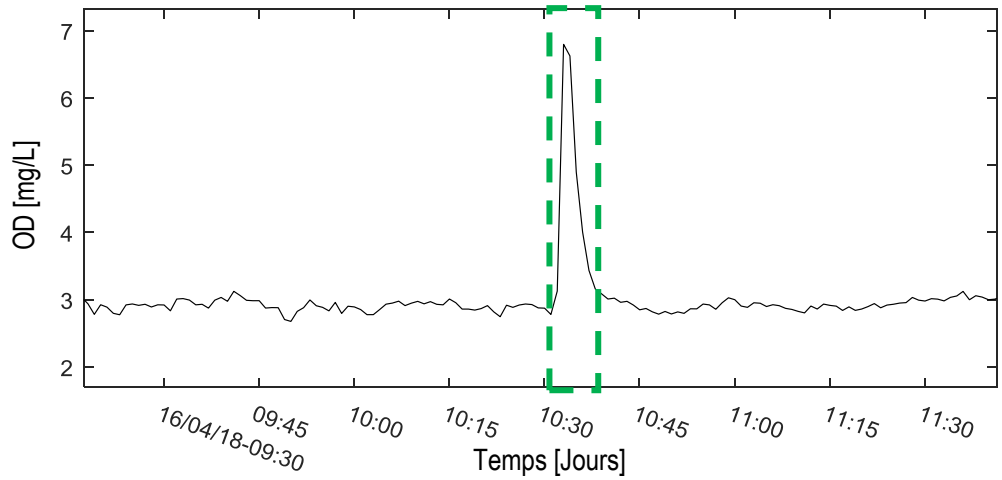


Figure 37. Nettoyage pro-actif d'un capteur d'oxygène dissous, du réacteur aéré pilEAUte (rectangle en pointillé vert)

4.1.1.2. kamEAU

Les capteurs du site kam*EAU* ont été présentés au chapitre 3 (section 3.1.2.1). Comme l'ensemble des capteurs mis en place dans le pil*EAU*te, ils peuvent se colmater et ainsi induire des problèmes de qualité des séries de données. La Figure 38 présente une période de données de MES avant et après le nettoyage (ligne verte en pointillés) du spectro::lyser installé à l'entrée de la station. Il est observé un bruit important dans la série avant ce nettoyage. Le 4 décembre (trait vert en pointillés), le nettoyage du capteur permet le retour à la normale de la série de données mais que durant quelques heures avec la réapparition du colmatage (de nombreux pics de bruit). Les hypothèses pouvant être amenées sur cet évènement sont une mauvaise procédure de nettoyage du capteur, un oubli de reprogrammation du nettoyage automatique à l'air ou une forte quantité de particules arrivant à l'entrée de la station durant cette courte période, car le capteur apparaît se rétablir à partir du 8 décembre.

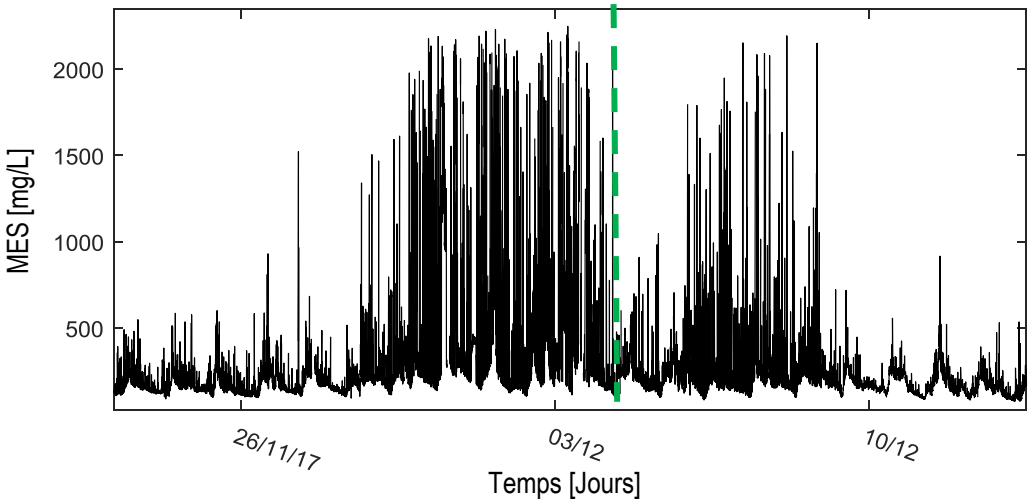


Figure 38. Colmatage rapide après le nettoyage du capteur (trait vert en pointillés) dans la série de données des MES (spectro::lyser, affluent KAMAK)

4.1.1.3. bordEAUx

Huit capteurs sont localisés à deux points de mesure : le réseau d'égouts et l'entrée de la StaRRE, Clos-de-Hild à Bordeaux, France. La doctorante Julia Ledergerber qui faisait la campagne de mesure, a été confrontée à un colmatage très important de ses capteurs. La Figure 39 présente des données de MES durant une partie de la campagne de quatre mois sur le site d'études ainsi que les instants des nettoyages (lignes en pointillées vertes). Il est remarqué, en premier lieu, que le nombre de nettoyage est plus important (deux fois par semaine) dû à un environnement (le réseau d'égout) plus difficile pour les capteurs. Cependant, même un nettoyage plus fréquent n'empêche pas le capteur de se colmater. La Figure 39 montre plusieurs dérives dans la série de données. Par exemple, le 25 mai, quelques jours après le nettoyage, une importante dérive est observée et arrêtée par le nettoyage avec l'apparition d'un biais. Ainsi, l'ensemble de la série de données de MES avant le troisième nettoyage (troisième trait en pointillés verts) sont des données non valides, car les nettoyages ont eu des effets sur la série de données.

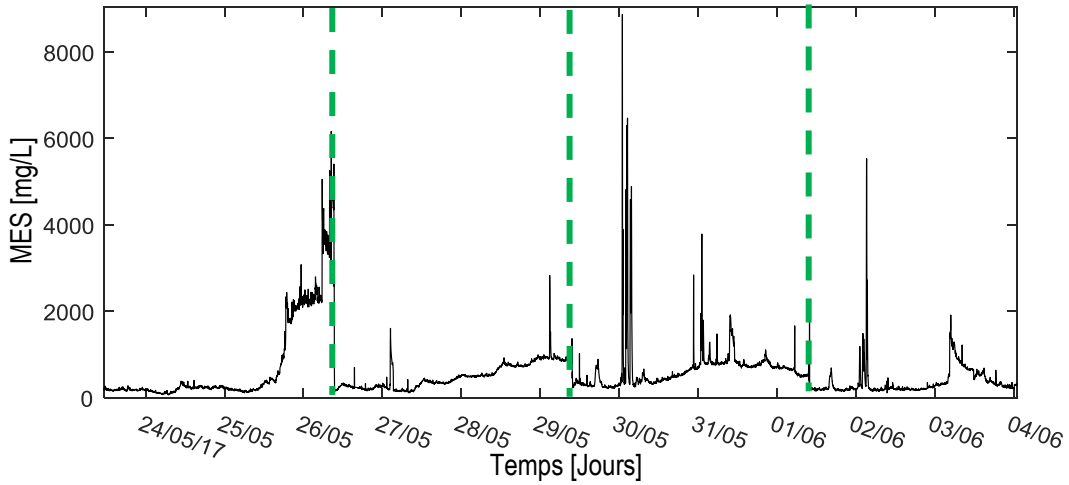


Figure 39. Dérive des données de MES (spectro::lyser, réseau d'égout Bordeaux) quelques temps après les nettoyages du capteur (lignes vertes pointillées)

4.1.2. Réaction des capteurs à une forte charge d'un composant

La réaction des capteurs à une forte charge d'un composant est exposée par le biais d'une expérience effectuée durant deux semaines pendant l'été 2018 dans le pil*EAU*te. Dans celle-ci, une multitude de capteurs présenté dans le Tableau 8 en redondance sont installés à une même localisation.

Tableau 8. Capteurs utilisés pour l'expérience et leurs variables mesurées dans un réacteur aéré du pilEAUte

Capteurs (Nombre de capteurs installés)	***Variables (Nombre de variables mesurées)***
spectro::lyser (x2)	NO_3-N (x2)
ammo::lyser	NH_4-N (x1) K (x1)
Varion	NH_4-N (x3) NO_3-N (x2) K (x3)

Les objectifs de cette expérience étaient :

- D'observer la réponse de chaque capteur pour un changement de concentration journalière;
- D'observer la réponse de chaque capteur à une forte concentration (test traceur; (Souidi, 2018)).

La Figure 40 et la Figure 41 montrent les données brutes de NH_4-N pour deux capteurs Varion (section 1.2.2) et les données brutes de NO_3-N pour deux capteurs spectro::lyser (section 1.2.2). Sur ces deux figures, il ressort les mêmes observations. Premièrement, un biais (flèches noires) important existe entre deux capteurs identiques mesurant une même variable, avec une même calibration au même moment par un seul chercheur. Deuxièmement, la réaction des capteurs est similaire dans le temps et le biais est toujours observable lors de l'injection. Ce biais est similaire pour la variable NH_4 dont le biais est équivalent à 70 % avant et pendant l'injection (Figure 40).

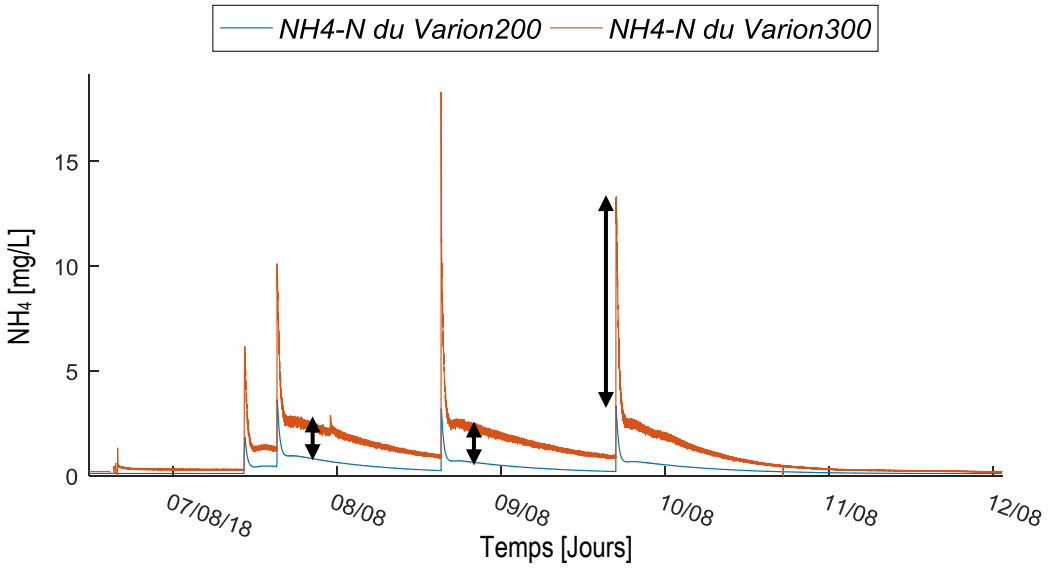


Figure 40. Suivi en continue de l'ammonium avec deux capteurs Varion à quatre fortes injections de NH_4 dans un réacteur aéré du pilEAUte

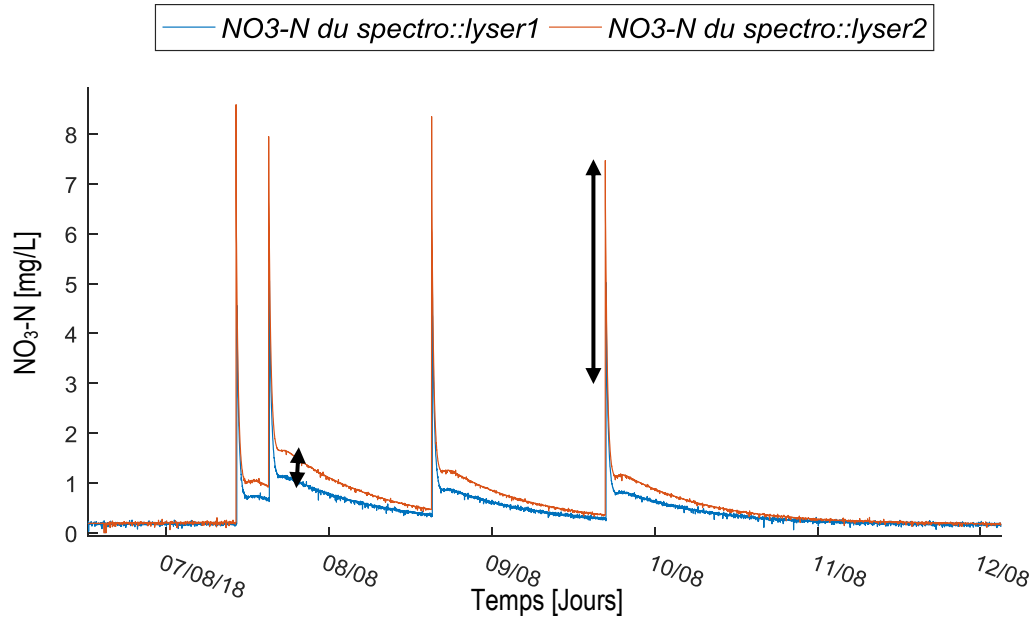


Figure 41. Suivi en continue des nitrates avec deux capteurs spectro::lyser à quatre fortes injections de NO_3 dans un réacteur aéré du pilEAUte

4.1.3. Conclusion

En conclusion de cette partie s'intitulant « La réaction des capteurs », il ressort que le nettoyage des capteurs nous informe sur la qualité des données. Le nettoyage permet de diminuer le colmatage du capteur causant une détérioration de la qualité des données. La solution consiste à mettre en place un nettoyage plus fréquent avec l'objectif de ne plus observer de biais apparaissant au moment du nettoyage. Pour ce faire, les outils automatiques développés dans la partie suivante pourraient être utiles. Ces derniers pourront détecter par exemple des dérives de données de leurs plages de valeurs normales dues au colmatage du capteur. C'est à ce moment que l'opérateur devra intervenir afin de rétablir les conditions normales.

L'expérience de l'injection d'une forte charge d'un composant, a montré que deux capteurs identiques mesurant un même paramètre (capteurs redondants) peuvent réagir similairement mais un biais peut être observable entre les séries de données. Une solution afin d'éliminer cet effet est la validation de données en ligne à partir de mesures de laboratoire régulières afin de calibrer les capteurs au besoin si une différence importante est observable entre les deux séries de données (Validation des capteurs : Redondance des capteurs).

4.2. Méthodes simples et modulaires de traitement des données

Cette partie expose les principales activités de ce projet de maîtrise, c'est-à-dire de rendre deux méthodes de traitement des données simples et modulaires dans leur application. Afin de montrer cette simplicité, les méthodes ont été documentées avec la rédaction de deux SOP en Annexes (SOP méthode univariée et SOP méthode multivariée) afin de faciliter et d'aider les futurs utilisateurs. Dans ce même ordre d'idée, cette partie développe l'élaboration des deux méthodes et leurs applications sur des séries de données de trois projets de recherche afin de donner une aide et des conseils supplémentaires aux utilisateurs. Ce dernier point amène aussi l'idée de la modularité des méthodes n'étant pas spécifiques à une série de données comme par le passé. Ainsi, les deux méthodes étudiées sont :

- La méthode univariée
- La méthode multivariée : L'Analyse par Composantes Principales (ACP)

La Figure 42 représente le diagramme général de ces deux méthodes (univariée et multivariée).

4.2.1. Méthode univariée

La méthode univariée expliquée dans le chapitre « Matériel et Méthodes » a pour but de détecter et retirer les données aberrantes et les fautes au sein des capteurs tels que le biais, la dérive, la défaillance du capteur et la dégradation de la précision (section 1.3). Cette méthode permet aussi l'amélioration du signal. Alferes et al. (2012) ont développé et utilisé leur méthode dans le but de traiter des séries de données provenant de capteurs installés dans des rivières.

Le traitement univarié des données a été simplifier par une implantation modulaire des différentes fonctions (Figure 42). Deuxièmement, ce point donne aussi l'idée de la modularité de la méthode. Par exemple, le bloc de la détection des données aberrantes peut être remplacé par un autre bloc permettant aussi cette détection telle que les réseaux de neurones. Les prochaines sous-parties développent chaque étape du traitement univarié ainsi que des séries de données traitées.

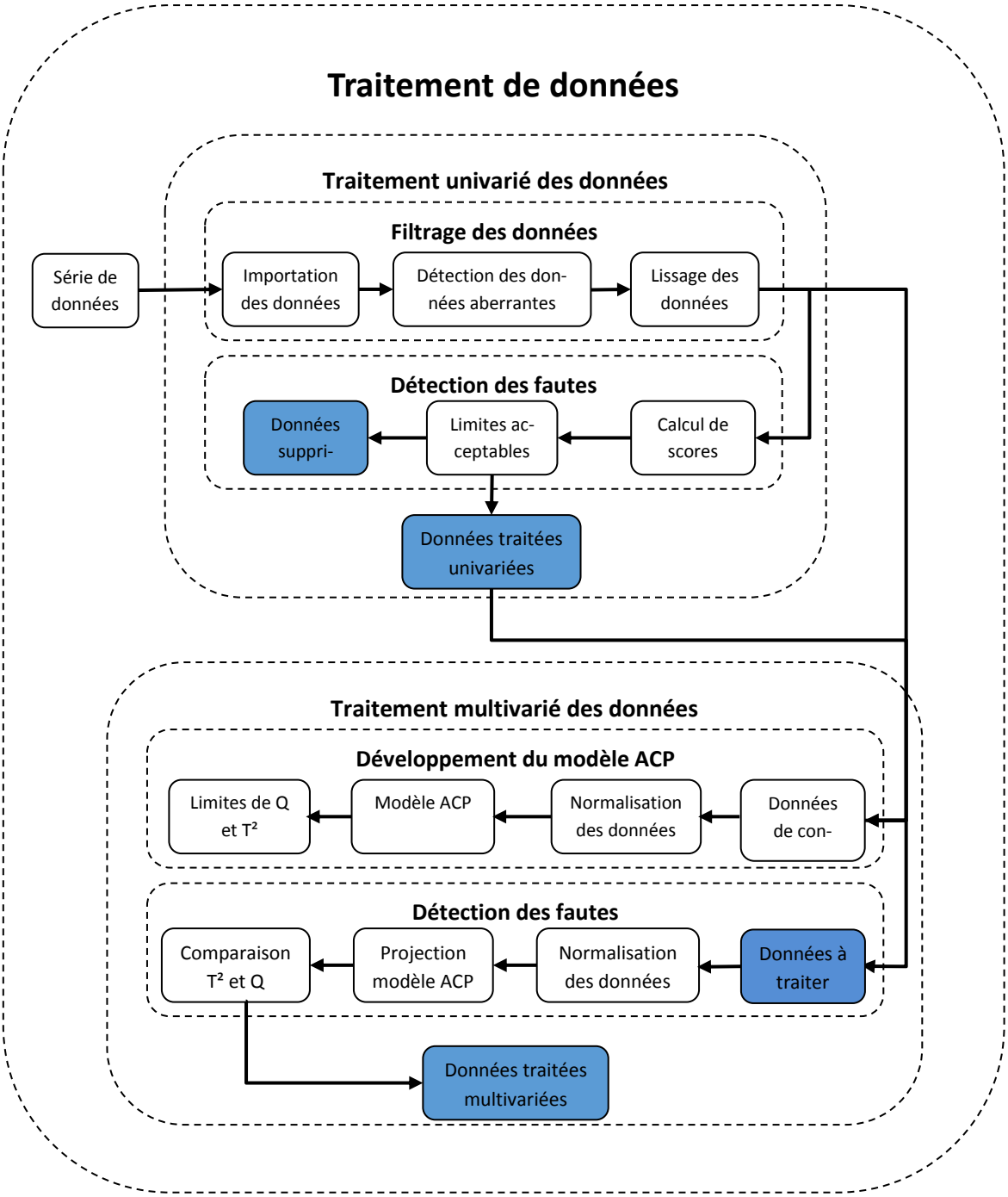


Figure 42. Diagramme général des deux phases de traitement des données (univariée et multivariée)

4.2.1.1. *Structure des données*

Avant de détailler le script de la méthode, une exposition de la structure des données doit être effectuée. L'importation de données dans MATLAB à partir d'un fichier « .csv » est réalisée par la fonction « **DataImport** » (Tableau 9) (SOP méthode univariée).

Tableau 9. Format du fichier «.csv»

Date et Temps	Points d'échantillonnage	Nom de la variables	Valeurs	Unités
Aaaa-mm-jj hh:mm				

Les séries de données dans MATLAB ont une structure comportant une colonne « channel » (Nom de la variable et son unité) et une colonne « values » (Temps en format MATLAB et valeurs) est obtenue (Figure 43). Il est important de respecter ce format de la structure lors de l'importation afin de ne pas rencontrer de problèmes à l'utilisation de la méthode.

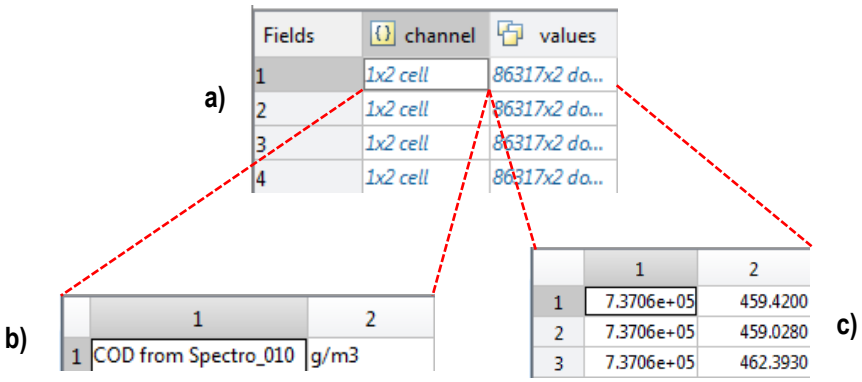


Figure 43. Exemple de structure de données importées dans MATLAB a) Structure générale b) Format de la colonne channel (nom de la variable, unité de la variable) c) Format de la colonne values (temps en format MATLAB, valeurs)

4.2.1.2. *Présentation du script général avec ces fonctions*

L'ensemble des parties du script se trouve dans le SOP en anglais pour une plus grande utilité auprès des utilisateurs potentiels. Pour chaque bloc montré en Figure 42, une fonction MATLAB permet d'effectuer l'étape automatiquement dans le but de faciliter l'utilisation de la méthode. En premier lieu, une génération de paramètres par défaut doit être mise en place à partir de la fonction « **DefaultParam** » (SOP méthode univariée). Ils sont documentés dans la fonction et dans le SOP (SOP méthode univariée).

Le Tableau 10 présente l'ensemble des paramètres utilisés et leur valeur par défaut pour la détection des données, le lissage et la détection des fautes. Les paramètres pour la détection de fautes sont initialisés à NaN (« not-a-number ») car ces derniers dépendent fortement de la variable traitée et donc ne devraient pas avoir de valeurs par défaut générales. Dans chaque projet de recherche étudié, des exemples de valeurs de ces paramètres seront donnés pour aider les futurs utilisateurs de l'outil.

Tableau 10. L'ensemble des paramètres utilisés dans la méthode univariée pour chaque étape

Détection des données aberrantes		
Paramètres	**Définition**	**Valeur par défaut**
param.nb_s	Facteur multiplicatif qui détermine le calcul de l'intervalle de prédiction. Une grande valeur de ce paramètre accepte la plupart des points et rejette seulement les valeurs aberrantes les plus évidentes.	3
param.nb_reject	Nombre de données rejetées consécutivement avant la réinitialisation de l'étape de détection des données aberrantes.	100
param.nb_backward	Nombre de données avant la dernière donnée rejetée.	15
param.MAD_ini	Écart absolu moyen utilisé pour démarrer et réinitialiser l'étape de détection des données aberrantes.	10
param.min_MAD	Écart absolu moyen minimum à utiliser.	0
param.ShowStats	Affiche des statistiques sur le processus de filtrage.	true
param.Verbose	Affiche des messages d'avertissement et d'erreur lorsqu'il est égal à « true ».	true
param.DT_RelRol	Le pas de temps (DT) doit être constant. Dans la série temporelle, la valeur maximale (maxDT) ou minimale (minDT) de DT est comparé à (1 + param.DT_RelRol) multiplié par la médiane de DT : maxDT > (1 + param.DT_RelRol) * medDT) **ou** minDT < (1 - param.DT_RelRol) * medDT) Sinon, le filtrage devra être effectuée avec prudence car la méthode assume un pas de temps constant.	0.01
param.restart	Redémarrage du filtrage à zéro.	true

Lissage des données		
param.h_smoother	Nombre de points pris en compte pour lisser une valeur spécifique par la moyenne mobile pondérée.	30
param.N_Reset	Si une série est de nouveau filtrée, la moyenne mobile exponentielle doit être appliquée à un certain nombre de points dits de préchauffage. Ce paramètre définit ce nombre de points. Aucune amélioration n'est habituellement visible pour une valeur supérieure à 4 ou 5.	2
Détection des fautes		
paramX.corr_max *paramX.corr_min*	Maximum et minimum du paramètre montrant si le bruit est distribué aléatoirement.	NaN
paramX.slope_max *paramX.slope_min*	Pente maximale et minimale attendue pour sur une bonne série de données.	NaN
paramX.std_max *paramX.std_minn*	Variation maximale et minimale du bruit des données.	NaN
paramX.range_max *paramX.range_min*	Valeur maximale et minimale attendue de la variable.	NaN

Ayant généré les paramètres par défaut, les différents « blocs de fonctions » peuvent être exécutés. Premièrement, le filtrage des données avec la détection des données aberrantes est effectué par la fonction **« Outlier-Detection »** suivie du lissage des données réalisé par la fonction « **kernel_smoother** » (SOP méthode univariée).

Deuxièmement, la détection des fautes est effectuée par la fonction « **D_score.** Précédent cette étape, l'utilisateur doit initialiser les paramètres « paramX.range_max » et « paramX.range_min ». Finalement, la dernière étape est l'obtention des données traitées (acceptées et rejetées). Ceci est opéré par la fonction « **TreatedD** ». En fin de processus de traitement, un certain de nombre de colonnes ont été ajoutées à la structure des données telles que les données acceptées « **AD** », les données lissées « **Smoothed_AD** », les indicateurs de défaillances « **D_scores** » et les données traitées « **Final_D** ». Cet ajout au fur et à mesure montre la grande modularité et simplicité de la méthode en effectuant étape par étape et en obtenant à chaque fois leur résultat. La Figure 44 présente un exemple de structure finale après le traitement de plusieurs variables séparément. En complément, pour chaque bloc de fonctions, des outils ont été implantés dans le but de visualiser chaque étape par un graphique (Figure 45). Enfin, l'ensemble des informations énoncées précédemment, le script, les fonctions (entrée, sorties des fonctions), les paramètres sont tous documentés plus en détails dans le SOP (SOP méthode univariée). Ceci permet une plus grande facilité aux futurs utilisateurs de la méthode.

Fields	channel	values	AD	Sec_Result	Smoothed_AD	Score	Final_D
1	1x2 cell	86317x2 do...	86317x1 do...	1x1 struct	86317x1 double	1x1 struct	1x1 struct
2	1x2 cell	86317x2 do...	86317x1 do...	1x1 struct	86317x1 double	1x1 struct	1x1 struct
3	1x2 cell	86317x2 do...	86317x1 do...	1x1 struct	86317x1 double	1x1 struct	1x1 struct
4	1x2 cell	86317x2 do...	86317x1 do...	1x1 struct	86317x1 double	1x1 struct	1x1 struct
5	1x2 cell	86317x2 do...	86317x1 do...	1x1 struct	86317x1 double	1x1 struct	1x1 struct
6	1x2 cell	86317x2 do...	86317x1 do...	1x1 struct	86317x1 double	1x1 struct	1x1 struct
7	1x2 cell	86317x2 do...	86317x1 do...	1x1 struct	86317x1 double	1x1 struct	1x1 struct

Figure 44. Structure finale après le traitement par la méthode univariée

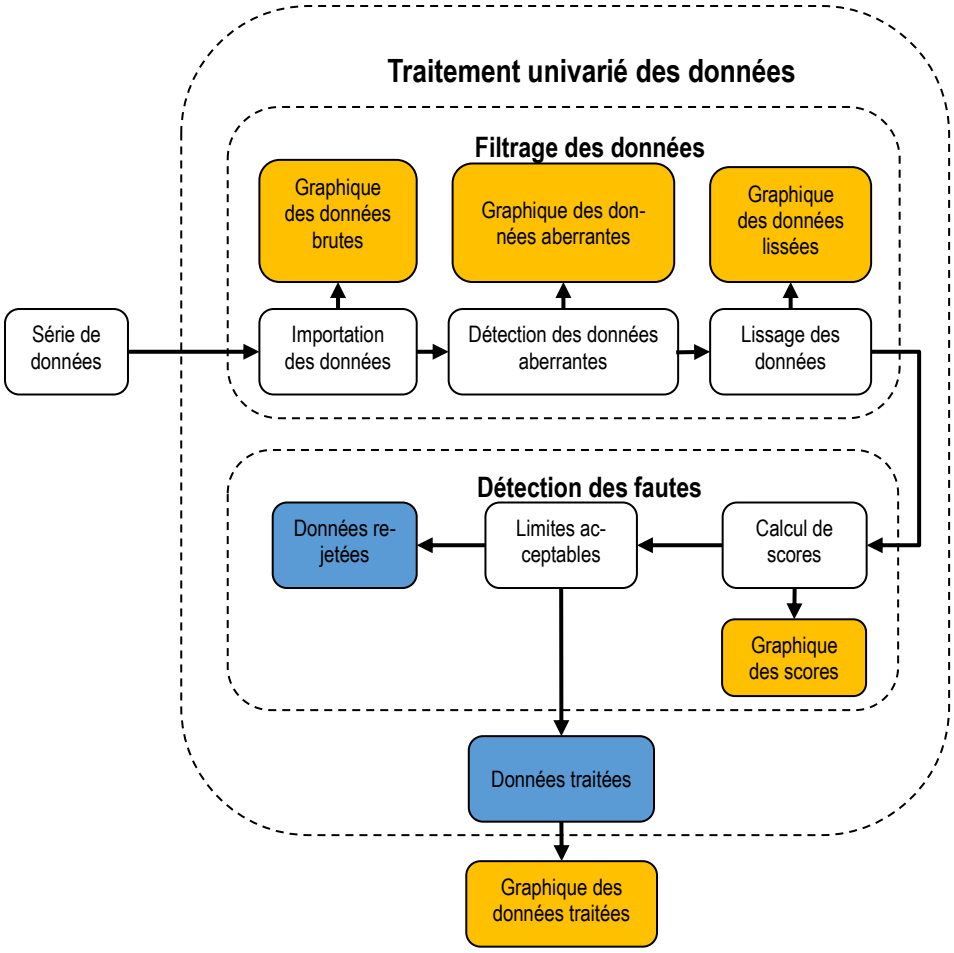


Figure 45. Diagramme général de la méthode univariée avec les outils graphiques

4.2.1.1. Applications

Avant l'utilisation de la méthode, des connaissances sur le procédé et sur le projet sont primordiales afin de connaître l'impact des nettoyages de capteurs, les attentes de la personne utilisant les données traitées et le

journal du suivi des données (« logbook » en anglais). Pour obtenir ces informations, des réunions avant, pendant et après le traitement ont été réalisées avec les personnes respectives de chaque projet : pil*EAU*te, kam*EAU* et bord*EAU*x.

a. pil*EAU*te

Pour le projet pil*EAU*te, les données ont permis de tester la méthode au cours de sa généralisation. Un exemple d'application sera montré sur des données d'oxygène dissous, mesuré au sein d'un réacteur aéré de boues activées (Figure 46). L'oxygène dissous est contrôlé par manipulation du débit d'air pour maintenir une valeur de 3 mg/L. L'application de chaque sous-étape de traitement est détaillée jusqu'à l'obtention des données traitées (acceptées et rejetées).

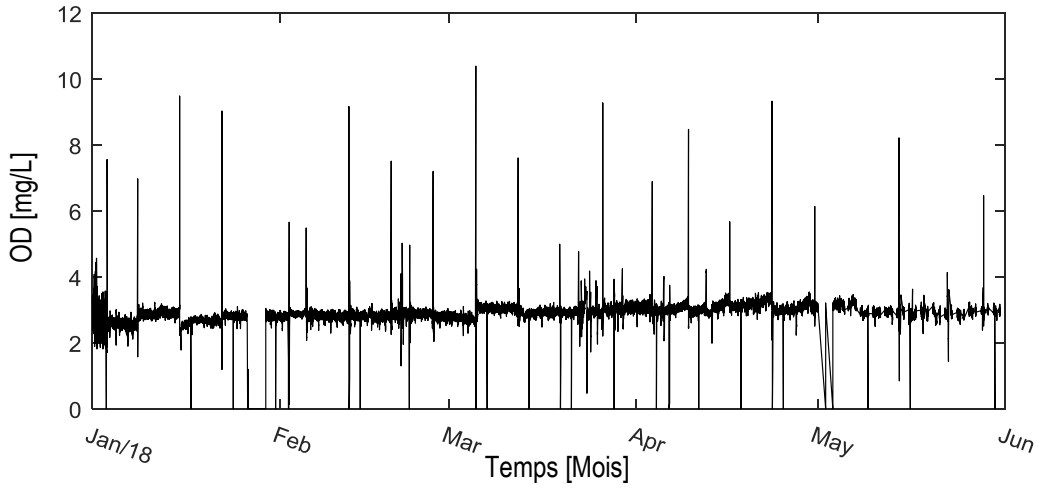


Figure 46. Données brutes d'oxygène dissous mesuré dans un réacteur aéré de boues activées du pilEAUte durant une période de 5 mois

o *Étape 1: Détection des données aberrantes*

Dans la première étape « **filtrage des données** », une détection des données aberrantes est opérée. Plusieurs paramètres doivent être choisis afin de mettre en place le modèle autorégressif (section a). Comme expliqué dans la partie 4.2.1.2, une valeur par défaut est proposée pour chaque paramètre à l'utilisateur au moment du premier lancement de la méthode. La détection des données aberrantes a été réalisée à l'aide des paramètres par défaut montrés dans le Tableau 10. La Figure 47 montre les données aberrantes détectées sur cette série de données, soit 1,24 % de données aberrantes identifiées. La Figure 48 montre un agrandissement d'une période où la méthode a permis la détection d'un nettoyage du capteur n'ayant pas d'effet sur la série de données.

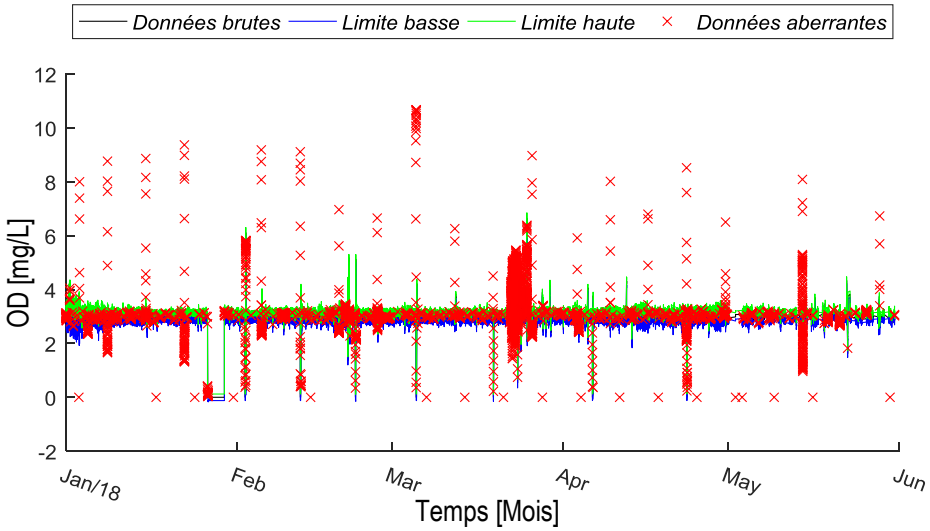


Figure 47. Détection des données aberrantes pour la variable oxygène dissous mesurée au sein d'un bioréacteur du pilEAUte durant une période de cinq mois

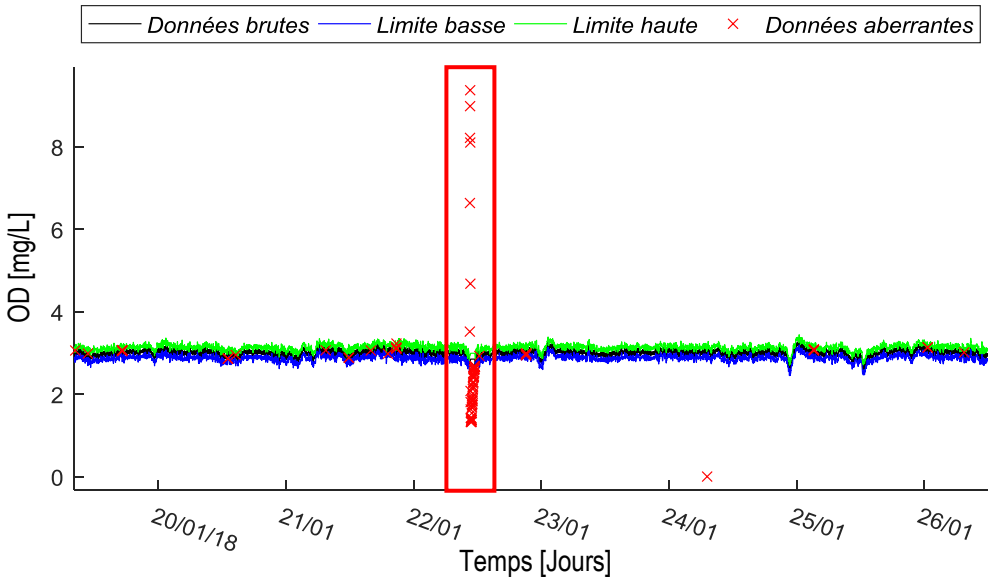


Figure 48. Détection de données aberrantes pour la variable d'oxygène dissous mesurée dans un bioréacteur du pilEAUte correspondant à un nettoyage hebdomadaire du capteur (détail de la Figure 47)

Durant la généralisation, une analyse de sensibilité de la méthode a été réalisé en modifiant les paramètres par défaut dans le but d'observer les effets sur la détection des données aberrantes. Par exemple, le paramètre « param.nb_s » a été diminué de la valeur 3 à 2. Ce changement a provoqué un pourcentage de données aberrantes plus élevé, égal à 7,9 % (Figure 49). La Figure 50 montre la même période de données que la Figure 48. Cependant, la détection est trop restrictive. Des bonnes données ou « faux positifs » se retrouvent en données aberrantes alors qu'elles ne devraient pas y être.

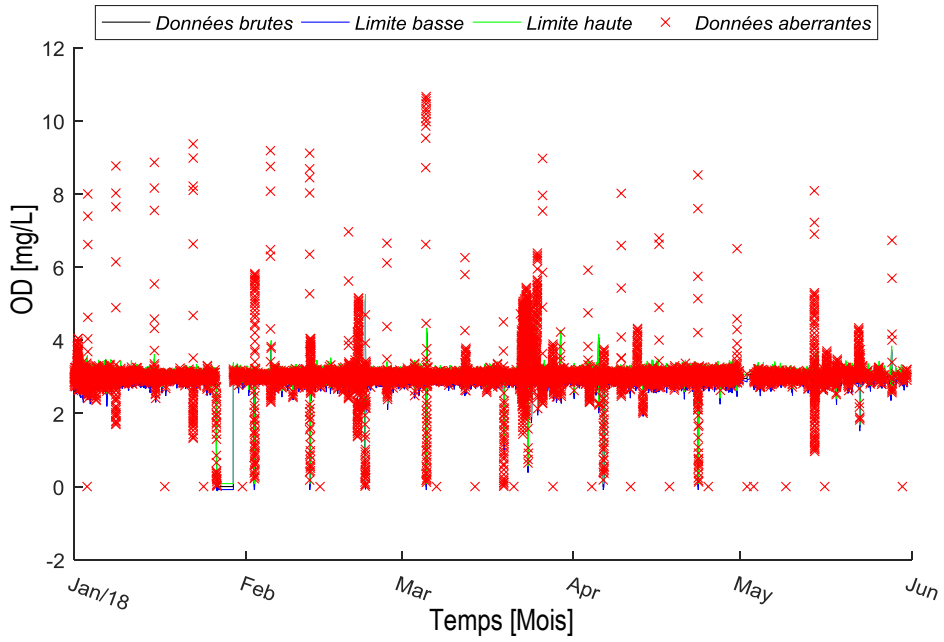


Figure 49. Détection des données aberrantes pour la variable oxygène dissous mesurée dans un bioréacteur du pilEAUte durant une période de cinq mois avec le paramètre « param.nb_s » égale à 2

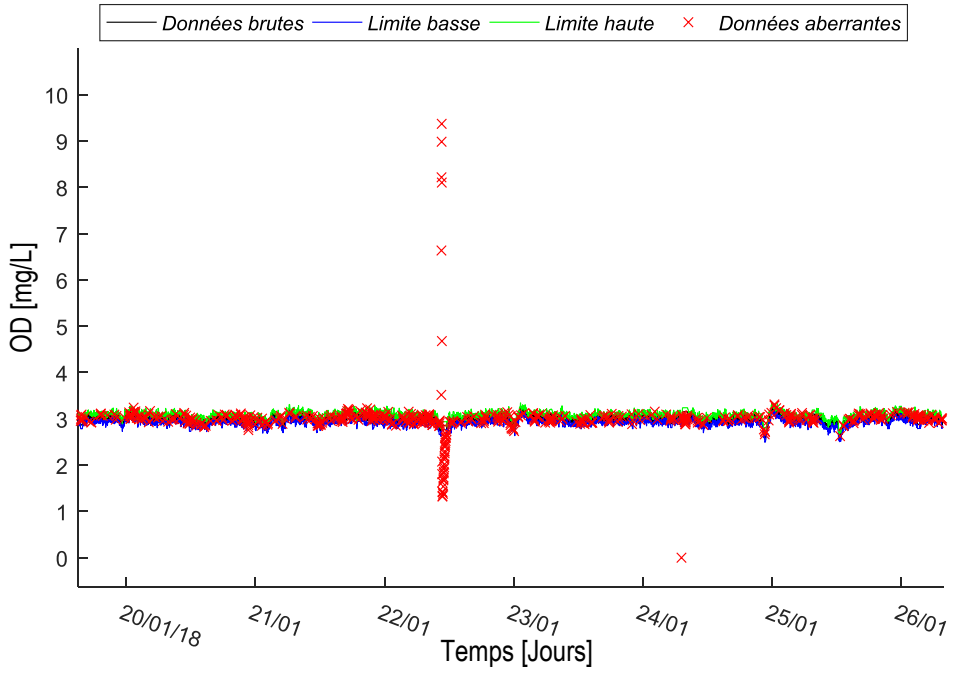


Figure 50. Détection de données aberrantes de la variable d'oxygène dissous mesurée dans un bioréacteur du pilEAUte correspondant à un nettoyage hebdomadaire du capteur avec le paramètre « param.nb_s » égale à 2 (détail de la Figure 49)

- *Étape 2: Lissage des données*

Le lissage des données est la seconde étape de la méthode. Ce lissage est réalisé à l'aide d'une moyenne mobile (section b) sur *n* points. La Figure 51 montre les données filtrées de la variable d'oxygène dissous en utilisant les paramètres par défaut exposés dans le Tableau 10. La Figure 52 illustre un agrandissement de la série de données d'oxygène dissous avec le lissage des données et la réduction du bruit.

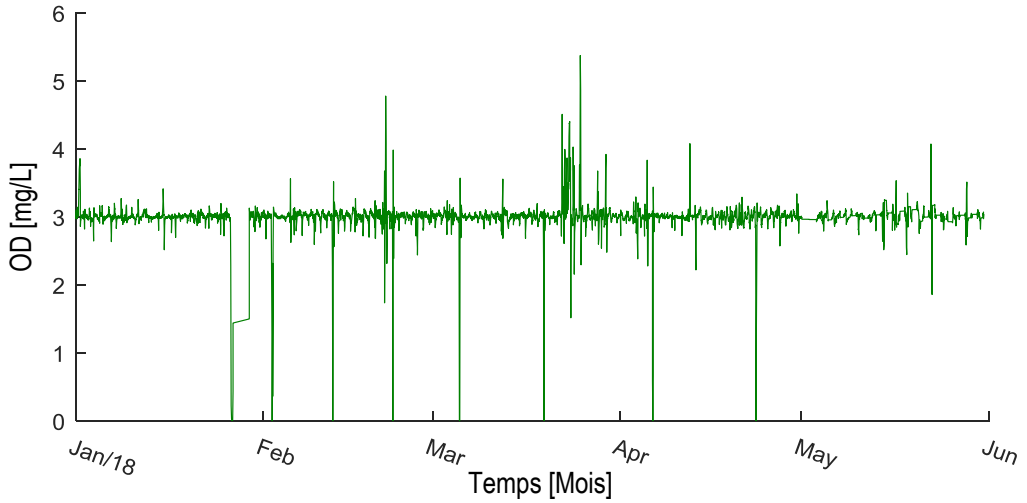


Figure 51. Données lissées d'oxygène dissous mesuré dans un bioréacteur du pilEAUte sur une période de cinq mois

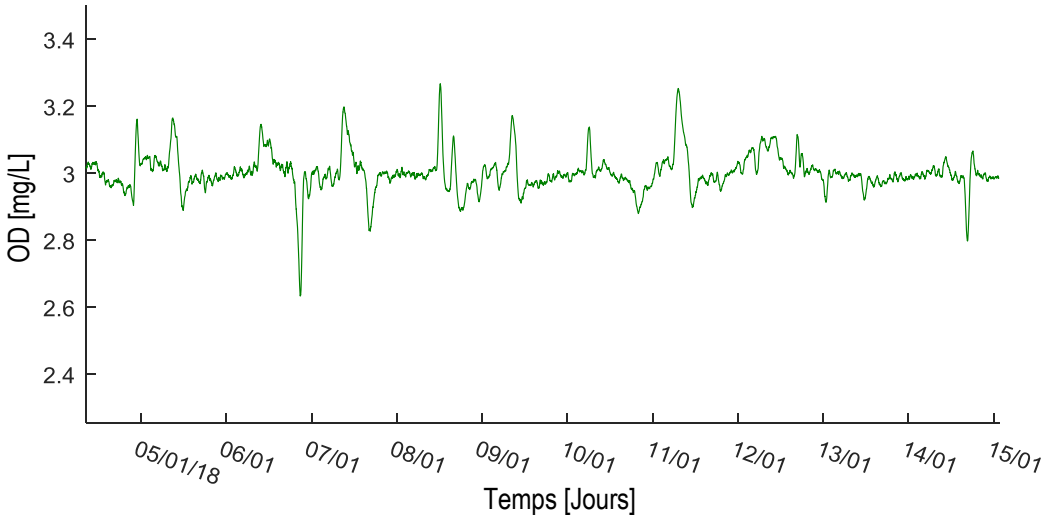


Figure 52. Agrandissement sur une période de dix jours des données filtrées d'oxygène dissous mesuré dans un bioréacteur du pilEAUte (détail de la Figure 51)

Pendant la généralisation de la méthode, il a été aussi effectué une analyse de sensibilité de la méthode en augmentant le « param.h_smoother » par exemple, de 30 à 200. La Figure 53 montre la même période que sur

la Figure 52 mais cette augmentation a eu pour effet un lissage plus prononcé et une perte de certaines variations importantes et valides.

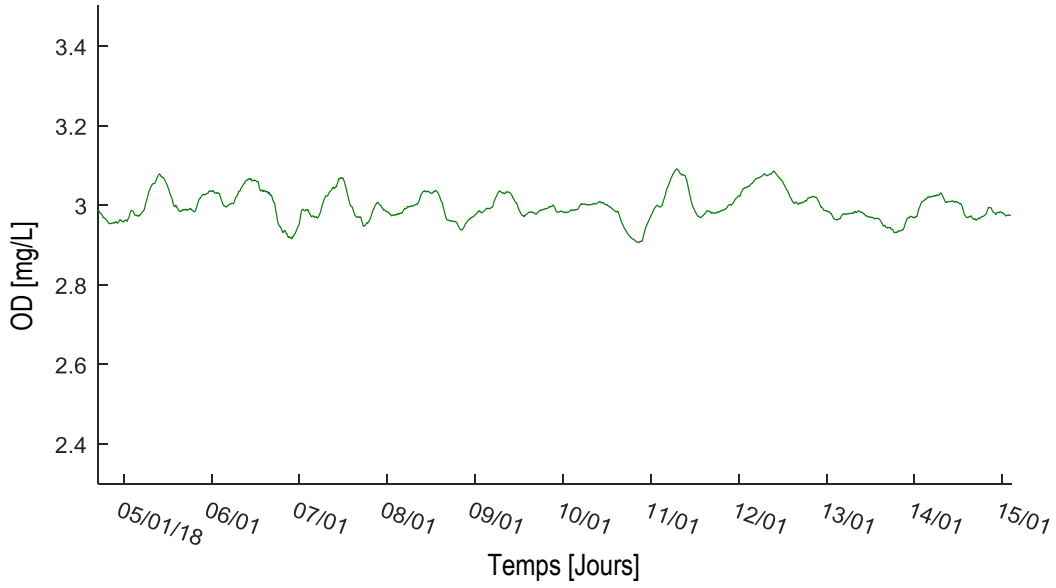


Figure 53. Agrandissement sur une période de dix jours des données filtrées d'oxygène dissous mesuré dans un réacteur aéré du pilEAUte avec l'augmentation du paramètre « param.h_smoother » à 200

○ *Étape 3: Détection des fautes*

De ces données filtrées, la troisième étape de la méthode, la détection des fautes peut être exécutée. Au sein de cette étape, quatre indicateurs de défaillances et leurs limites sont déterminés. Les limites de ces indicateurs de défaillances sont initialisées à une valeur NaN lors de la génération des paramètres au début de la méthode. Ce choix des valeurs est basé sur la connaissance du système telle que les valeurs normales de l'oxygène au sein du réacteur pour le paramètre « plage des données (« range » en anglais) », des nettoyages hebdomadaires du capteur, la dynamique normale, le bruit normal, des pannes ou autres problèmes survenus sur le capteur pour les autres indicateurs de défaillances. La Figure 54 illustre les données lissées d'oxygène dissous de la Figure 51 les indicateurs de défaillances calculés et leurs limites hautes et basses. Il est observé la détection de plusieurs fautes où certains indicateurs de défaillances sont en-dessous ou au-dessus de leurs limites. Par exemple, la faute détectée (rectangle rouge) par les indicateurs de défaillances « pente » et « plage des valeurs » est due à une défaillance complète du capteur d'après le fichier du suivi.

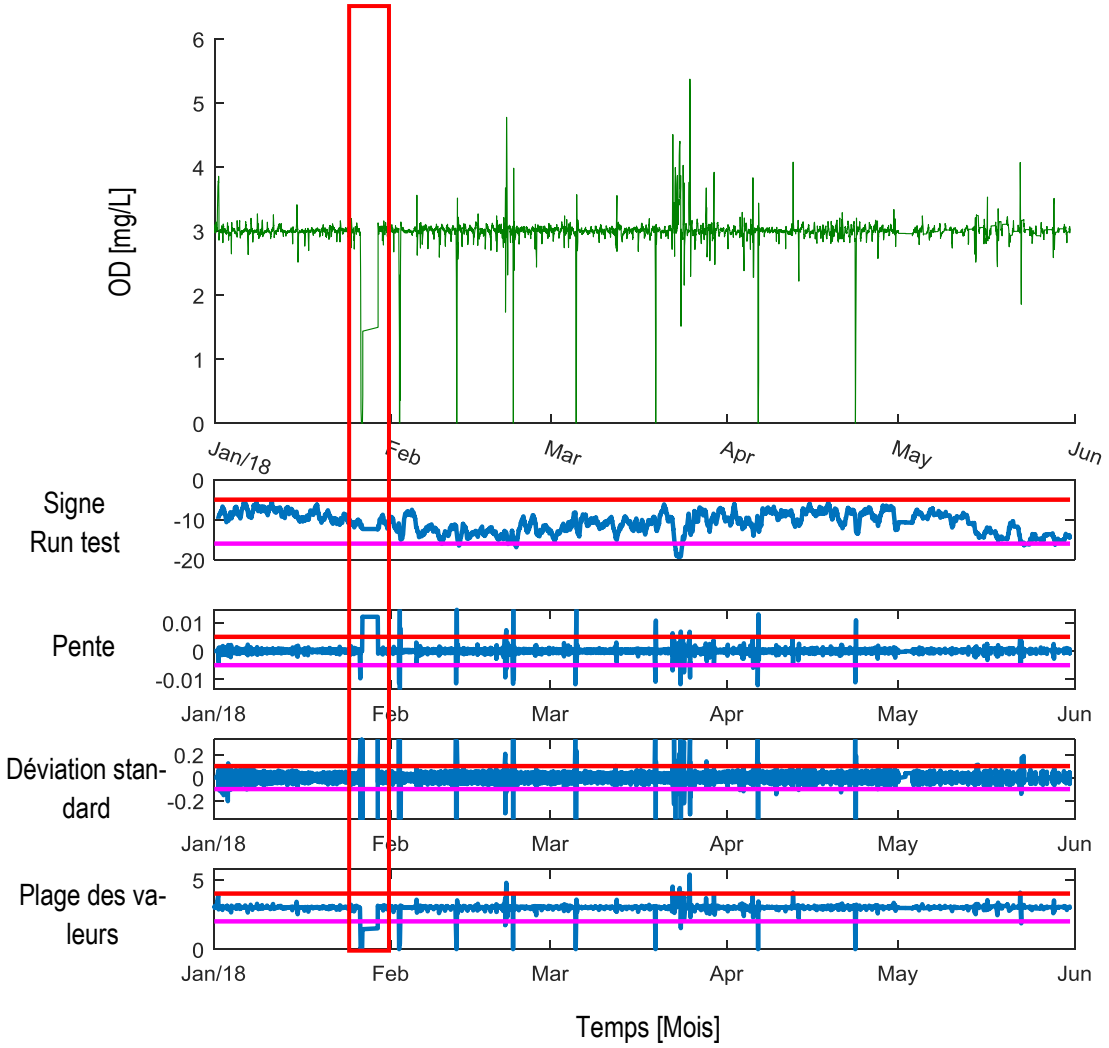


Figure 54. Détermination des indicateurs de défaillances en comparaison avec leurs limites pour la détection des fautes sur la période de cinq mois de l'oxygène dissous mesuré dans un bioréacteur du pilEAUte

○ *Étape 4: Données traitées*

Enfin, les données traitées (acceptées et rejetées) sont obtenues lors de la dernière étape. La Figure 55 illustre les données acceptées finales de l'ensemble de la série de données traitées. Sur cette période de six mois (190 000 données), environ 3 % des données ont été rejetées. Ce pourcentage est inférieur à l'intervalle exposé par Alferes et al., (2013a) qui est de 5 à 60 %. En réalisant un agrandissement de sur la Figure 55, la Figure 56a montre une période illustrant deux colmatages et une défaillance du capteur. Sur la Figure 56b, il est observé que les trois fautes ont été détectées et éliminées de la série de données finales.

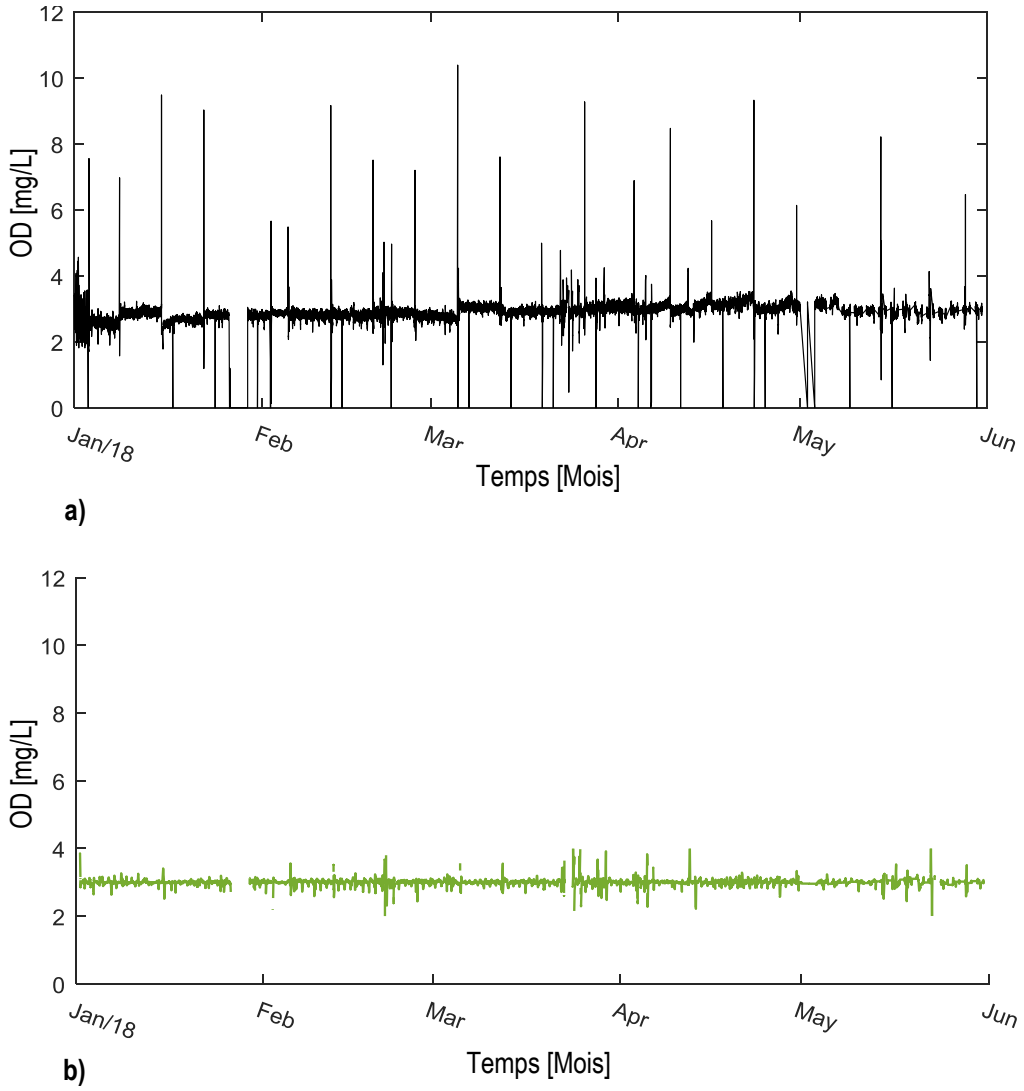


Figure 55. Données brutes et traitées de la variable d'oxygène dissous mesuré dans un réacteur aéré du pilEAUte a) Données brutes b) Données traitées

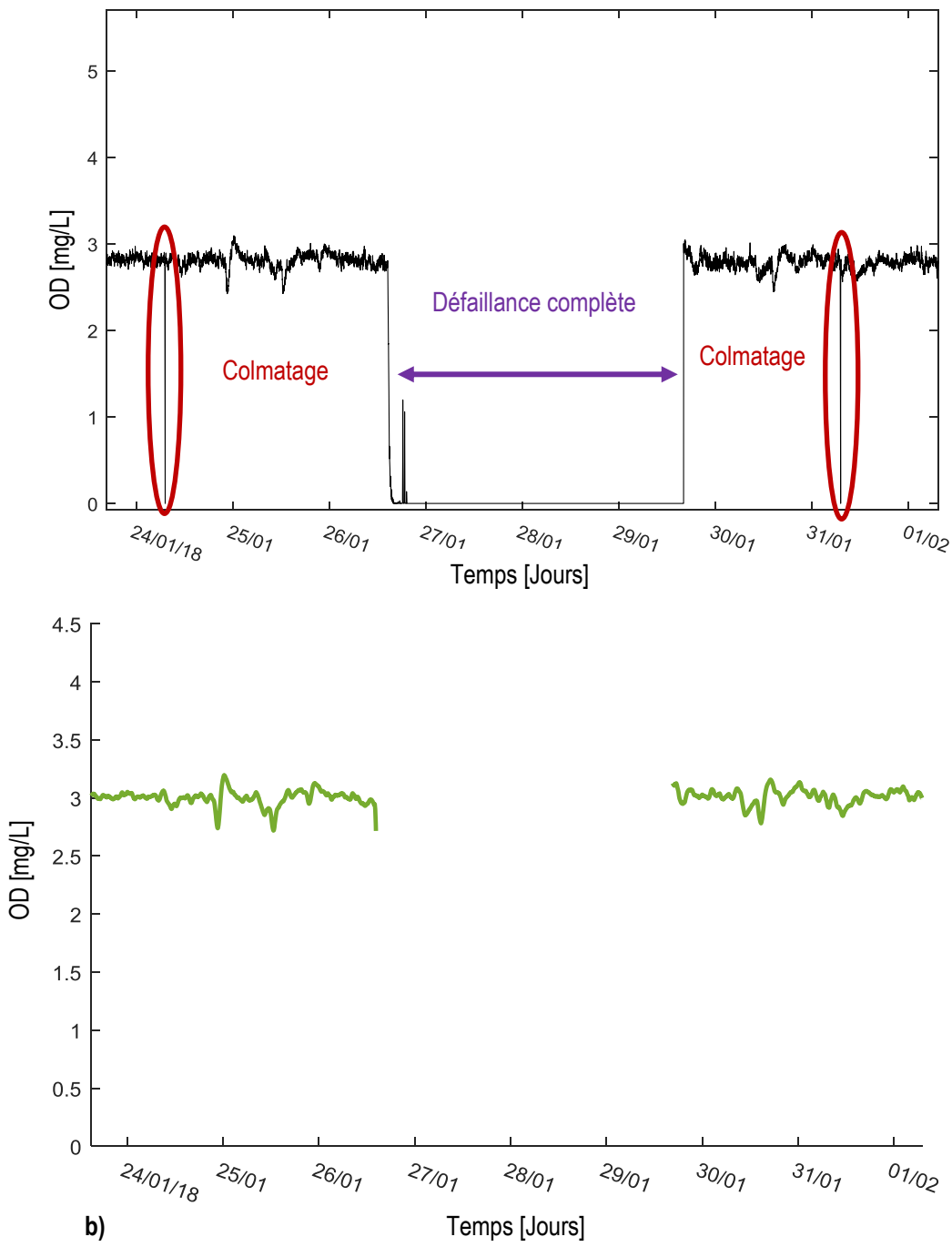


Figure 56. Agrandissement sur la détection d'une défaillance complète et deux colmatages du capteur d'oxygène dissous mesuré dans un bioréacteur du pilEAUte (détail de la Figure 55) a) Données brutes b) Données traitées

o *Compléments*

Pour plusieurs capteurs et leurs variables respectives dans le projet pil*EAU*te, une quantification des divers paramètres utilisés dans la méthode a été résumée dans le Tableau 11. Lorsque que certains paramètres n'apparaissent pas dans le tableau, cela signifie que le paramètre par défaut a été pris en compte. Ce tableau donne des aides et une plus simple utilisation de l'outil pour les futurs utilisateurs de la méthode.

Tableau 11. Quelques exemples de valeurs de paramètres pour la méthode univariée pour les capteurs du pilEAUte

Capteur	***Variables***	***Paramètres de la méthode univariée***	
		Filtrage des données	***Détection des fautes***
spectro::lyser	*DCO totale mesurée à l'effluent d'un décanteur primaire*		**Signe run-test :** - Limite haute = 20 - Limite basse = - 20 **Pente [mg/L.min] :** - Limite haute = 0,25 - Limite basse = - 0,25 **Déviation standard [mg/L] :** - Limite haute = 0,07 - Limite basse = - 0,07 **Plage des valeurs [mg/L] :** - Limite haute = 600 - Limite basse = 50
ammo::lyser	*NH_4 mesuré à l'effluent d'un décanteur primaire*	param.h_smoother = 5	**Signe run-test :** - Limite haute = 20 - Limite basse = - 20 **Pente *[mgN/L.min]* :** - Limite haute = 0,1 - Limite basse = - 0,1 **Déviation standard [mgN/L] :** - Limite haute = 0,1 - Limite basse = - 0,1 **Plage des valeurs [mgN/L] :** - Limite haute = 80 - Limite basse = 5

Varion	*NO_3 mesuré à l'effluent d'un décanteur secondaire*	*param.h_smoother = 40*	**Signe run-test :** - Limite haute = 20 - Limite basse = -10 **Pente [mgN/L.min] :** - Limite haute = 0,002 - Limite basse = - 0,002 **Déviation standard [mgN/L] :** - Limite haute = 0,03 - Limite basse = - 0,03 **Plage des valeurs[mgN/L] :** - Limite haute = 15 - Limite basse = 2
Conductimètre	*Conductivité mesurée à l'effluent d'un décanteur primaire*		**Signe run-test :** - Limite haute = 20 - Limite basse = - 20 **Pente [µS/cm.min] :** - Limite haute = 1 - Limite basse = - 0,5 **Déviation standard [µS/cm] :** - Limite haute = 0,1 - Limite basse = - 0,1 **Plage des valeurs [µS/cm] :** - Limite haute = 3000 - Limite basse = 500
Solitax	*MES mesurées dans un réacteur à boues activées*	param.h_smoother = 5	**Signe run-test :** - Limite haute = 20 - Limite basse = -20 **Pente [mg/L.min] :** - Limite haute = 5 - Limite basse = - 5 **Déviation standard [mg/L] :** - Limite haute = 0,1 - Limite basse = - 0,1 **Plage des valeurs [mg/L] :** - Limite haute = 4000 - Limite basse = 500

LDO	*OD mesuré dans un réacteur à boues activées*		**Signe run-test :** - Limite haute = - 5 - Limite basse = - 16 **Pente [mg/L.min] :** - Limite haute = 0,005 - Limite basse = - 0,005 **Déviation standard [mg/L] :** - Limite haute = 0,1 - Limite basse = - 0,1 **Plage des valeurs[mg/L] :** - Limite haute = 4 - Limite basse = 2

b. kam*EAU*

Pour le projet kam*EAU*, des illustrations finales du traitement des données pour la méthode univariée seront exposées pour deux des trois points de mesure sur le site d'étude. Pour l'ensemble des illustrations, une quantification des paramètres a été effectuée afin de donner une aide supplémentaire aux futurs utilisateurs. Les données collectées sur une période d'un an serviront à la calibration et la validation d'un modèle représentant le système KAMAK permettant l'optimisation du système (section 3.1.2). Les diverses séries de données ont eu un prétraitement afin d'éliminer les données insensées. Ces dernières correspondent aux périodes où le capteur était en faute d'après un journal du suivi des capteurs. Ainsi, les résultats finaux devaient améliorer le signal brut et rejeter un minimum de données étant donné la précédente perte de données par le prétraitement manuel.

o *Entrée de la station*

Pour ce premier point de mesure, deux exemples de résultats de données traitées sont illustrés par les variables DCO et température. Premièrement, pour la DCO, les paramètres par défaut de la méthode ont été adaptés à la série de données en essayant de rejeter un minimum de données. Le Tableau 12 présente les deux paramètres modifiés dont le paramètre « param.nb_s » ayant pour but d'accepter plus ou moins de données et de rejeter les valeurs aberrantes les plus évidentes. Ici, celui-ci a été doublé afin de conserver le maximum de données étant donné le prétraitement appliqué. Le paramètre « param.nb_reject » a été lui diminué à une valeur de 30 afin de réinitialiser la méthode de détection des données aberrantes toutes les 30 minutes. Cette durée est équivalente à un nettoyage habituel du capteur. Ainsi, la majorité des nettoyages a été détectée et définie comme des données aberrantes. Avant de passer au résultat final, le Tableau 13 présente les limites des quatre indicateurs de défaillances qui ont permis de détecter les fautes au sein de la série de données.

Tableau 12. Valeurs des paramètres modifiés pour la méthode de filtrage des données pour la variable DCO mesurée à l'entrée du KAMAK

Paramètres	***Valeurs par défaut***	***Nouvelles valeurs***
param.nb_s	3	6
param.nb_reject	100	30

Tableau 13. Limites des indicateurs de défaillances pour la détection de fautes dans la série de données de la variable DCO mesurée à l'entrée du KAMAK

Indicateurs de défaillances	**Limites**
Signe run-test	Limite haute = 9 Limite basse = - 22,3
Pente [mg/l.min]	Limite haute = 3 Limite basse = - 3
Déviation standard [mg/l]	Limite haute = 1 Limite basse = - 0,75
Plage des valeurs [mg/l]	Limite haute = 1500 Limite basse = 50

Ayant accompli la méthode au complet, la Figure 57 représente les données brutes et traitées de DCO. Il est observé que les données brutes et traitées sont très bruitées. Cependant, en zoomant sur une période (Figure 58), il est remarqué que le signal est amélioré avec une élimination des données aberrantes et une conservation du patron journalier des eaux usées en entrée de station.

De ces résultats, il peut être ressorti quelques informations telles que le pourcentage de données aberrantes et de données rejetées (Tableau 14). Le pourcentage des données aberrantes est de 6 et 1 % pour les données rejetées en coordination avec les attentes de l'utilisateur final. Lors du prétraitement, le doctorant avait rejeté 8 % de données. Ainsi, sur cette période d'un an comportant 542 881 données, au total 9 % de données ont été rejetées par le prétraitement et le traitement univarié.

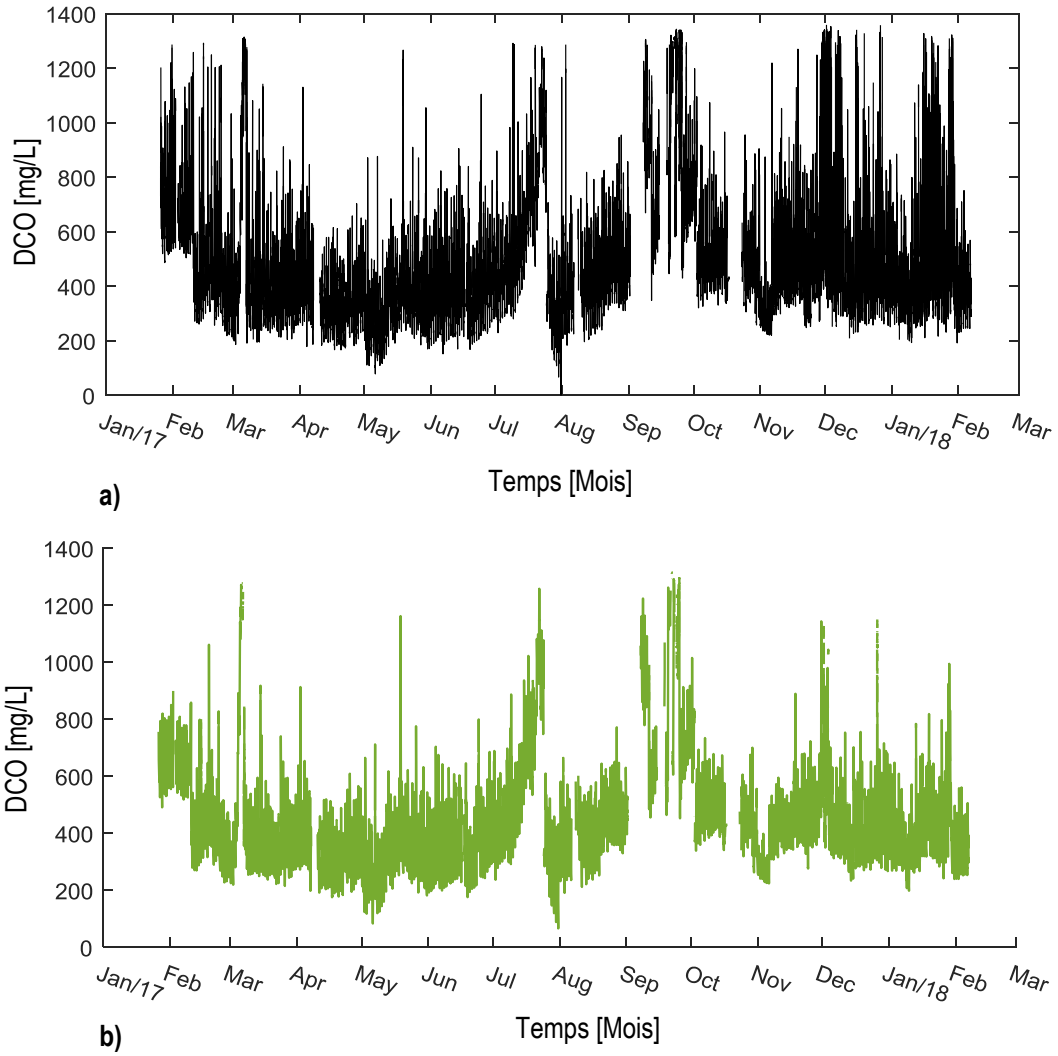
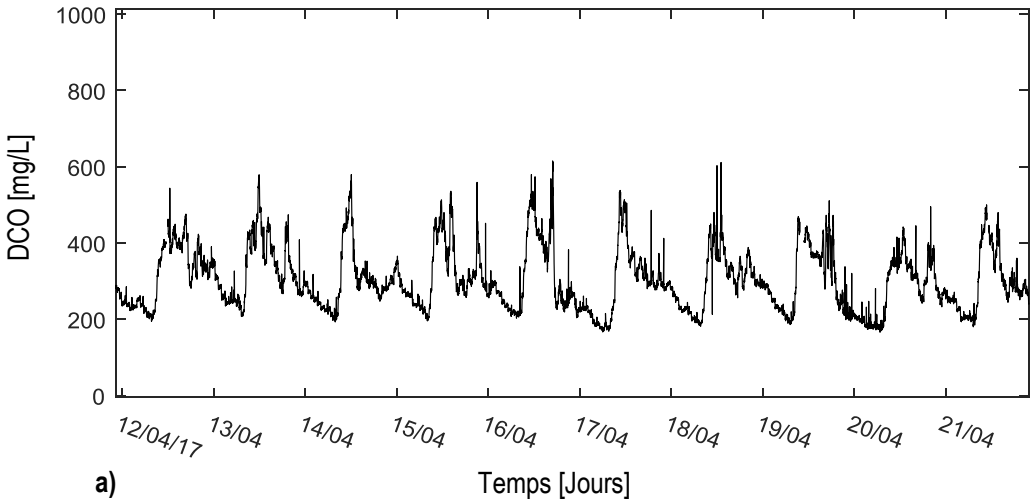


Figure 57.Données brutes et traitées de la variable DCO mesurée à l'entrée d'un étang aéré KAMAK pour une période d'un an a) Données brutes b) Données traitées



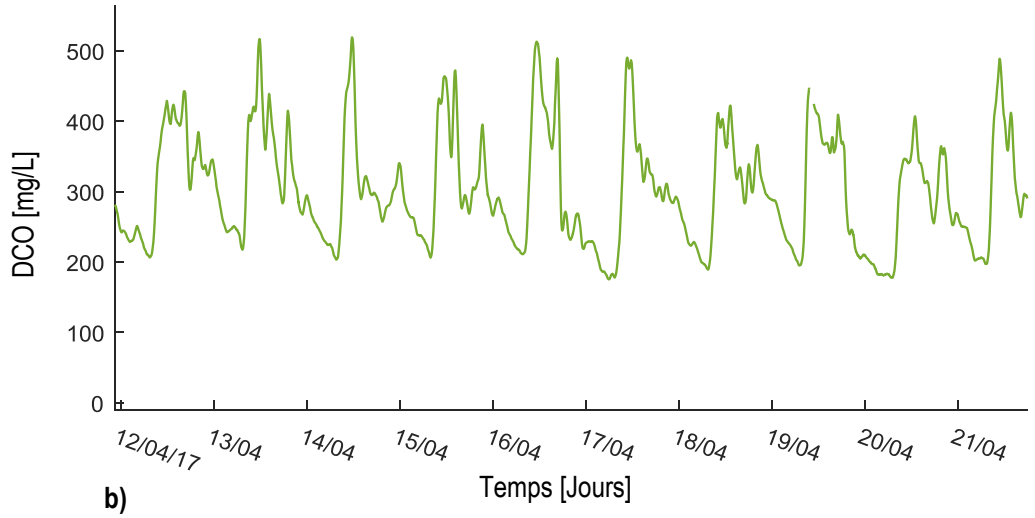


Figure 58. Agrandissement sur les données brutes et traitées de la variable DCO mesurée à l'entrée d'un étang aéré KAMAK sur une période de neuf jours (détail de la Figure 57) a) Données brutes b) Données traitées

Tableau 14. Pourcentage des données aberrantes et rejetées pour la variable DCO mesurée à l'entrée d'un étang aéré KAMAK

Paramètres	*DCO*
Nombre de données	542 881
% de données aberrantes	6 %
% de données rejetées	1 %

Pour la variable température, certains paramètres par défaut ont dû être modifiés afin de rejeter un minimum de données tout en gardant le patron annuel de la température (Figure 59). La valeur du paramètre « param.nb_reject » est aussi basée sur le temps de nettoyage du capteur qui était plus long que pour le capteur de DCO. Le paramètre de lissage des données « param.h_smoother » est plus grand pour cette variable afin de garder les patrons journaliers mais en éliminant les faibles variations de quelques dixièmes de degré à court terme. Enfin, le Tableau 16 montre les diverses limites des indicateurs de défaillances pour la détection des fautes. Cette quantification de paramètres et de limites pourra aider les utilisateurs futurs de la méthode.

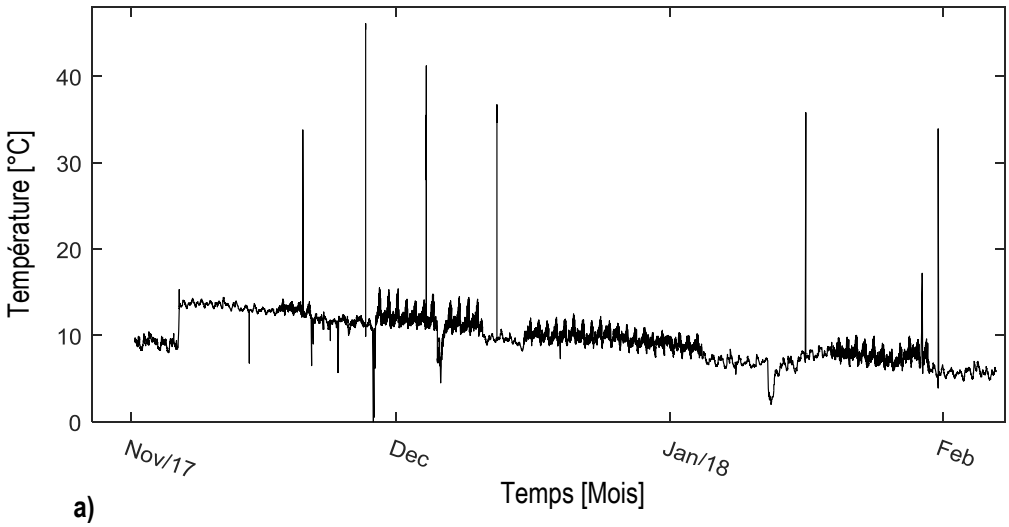
Tableau 15. Valeurs des paramètres modifiées pour la méthode de filtrage des données pour la variable température mesurée à l'entrée d'un étang aéré KAMAK

Paramètres	***Valeurs par défaut***	***Nouvelles valeurs***
param.nb_reject	100	40
param.h_smoother	30	60

Tableau 16. Limites des indicateurs de défaillances pour la détection de fautes dans la série de données de la variable température mesurée à l'entrée du KAMAK

Indicateurs de défaillances	**Limites**
Signe run-test	Limite haute = - 18 Limite basse = - 25
Pente [°C.min]	Limite haute = 0,15 Limite basse = - 0,25
Déviation standard [°C]	Limite haute = 0,5 Limite basse = - 2
Plage des valeurs [°C]	Limite haute = 20 Limite basse = 0,05

La Figure 59 expose les données brutes et traitées du suivi de la température, en entrée de station durant une courte période de trois mois. Un agrandissement est montré en Figure 60. Les variations de la température sont gardées et une élimination des données aberrantes est effectuée. La méthode a aussi conservé par exemple, les fortes variations soudaines de la température, le 6 novembre dues à un nettoyage du capteur et la fonte des neiges en mi-janvier. Les pourcentages de données aberrantes et de données rejetées sont résumés dans le Tableau 17. Concernant le pourcentage des données rejetées, la valeur est très faible, égale à 0,1 % sur cette période de trois mois.



a)

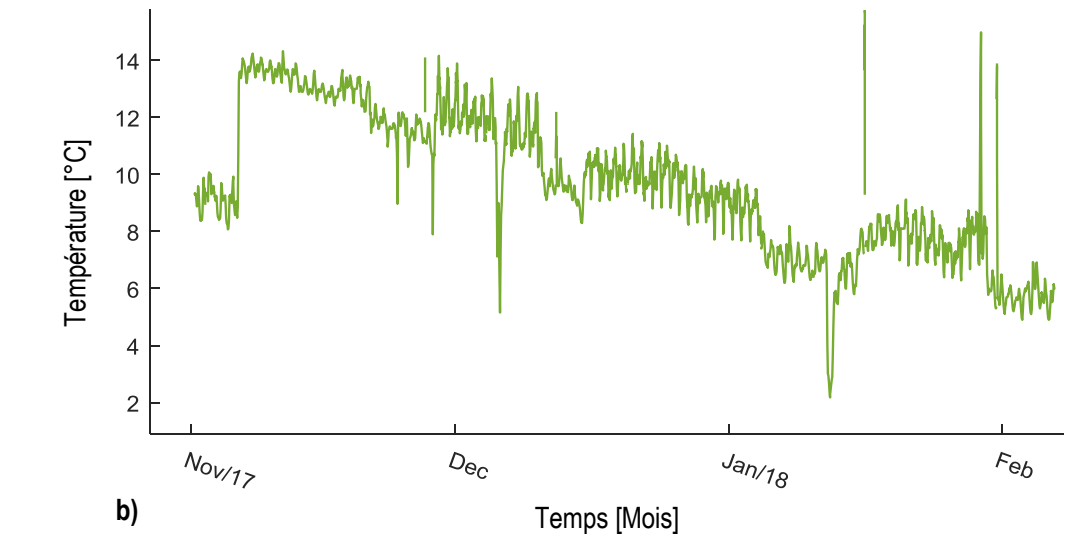
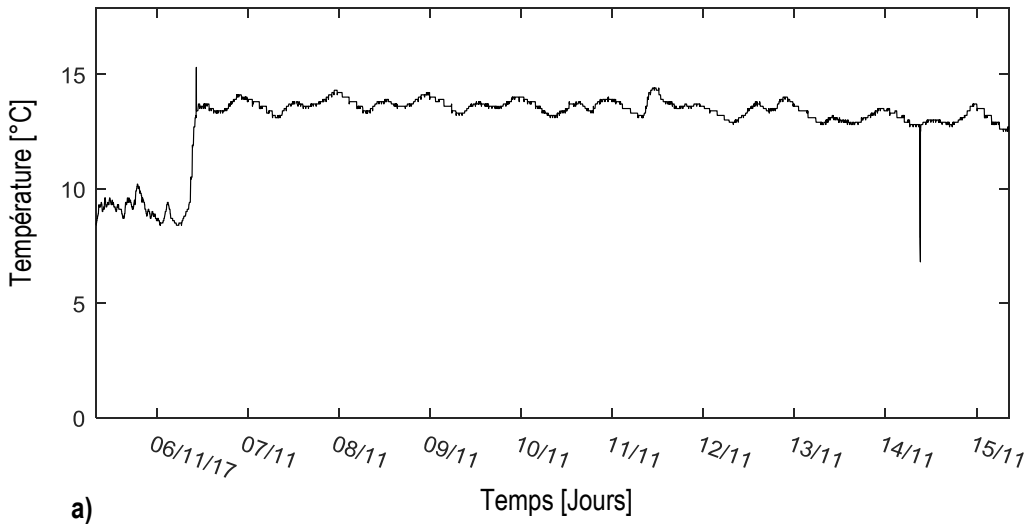


Figure 59. Données brutes et traitées de température mesurée à l'entrée d'un étang aéré KAMAK durant la période de trois mois a) Données brutes b) Données traitées



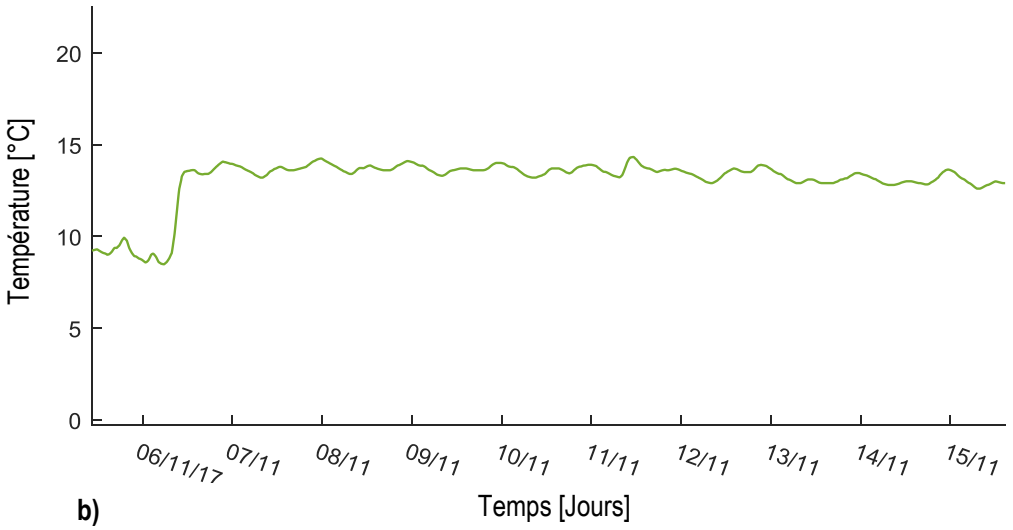


Figure 60. Agrandissement sur les données de température mesurée à l'entrée d'un étang aéré KAMAK sur une période de douze jours (détail de la Figure 59) a) Données brutes b) Données traitées

Tableau 17. Pourcentage des données aberrantes et rejetées pour la variable température mesurée à l'entrée d'un étang aéré KAMAK

Paramètres	*Température*
Nombre de données	139 679
% de données aberrantes	17 %
% de données rejetées	0,1 %

○ *Sortie de la station*

En sortie de station, les données traitées seront exposées pour les variables nitrate et oxygène dissous. Tout d'abord, comme pour l'affluent, les paramètres par défaut ont dû être modifiés et quantifiés dans des tableaux. Ceci aidera les futurs utilisateurs de la méthode. Comme pour les séries de données à l'entrée, le but du traitement était de garder le maximum de données tout en rejetant les données aberrantes les plus évidentes telles que celles créées par les nettoyages des capteurs. Le Tableau 18 présente le seul paramètre modifié dans la méthode de traitement de données. Cette modification est aussi basée sur les cycles de nettoyage manuels de l'opérateur. Pour ce capteur, la durée du nettoyage était généralement d'une vingtaine de minutes, d'où la valeur 20 pour le paramètre « param.nb_reject ». Le Tableau 19 présente les limites des indicateurs de défaillances pour la détection des fautes pour la variable nitrate.

Tableau 18. Paramètre modifié pour la méthode de filtrage des données pour la variable nitrate mesurée à la sortie d'un étang aéré KAMAK

Paramètre	***Valeur par défaut***	***Nouvelle valeur***
param.nb_reject	100	20

Tableau 19. Limites des indicateurs de défaillances pour la détection de fautes pour la série de données de la variable nitrate mesurée à la sortie d'un étang aéré KAMAK

Indicateurs de défaillances	**Limites**
Signe run-test	Limite haute = 3 Limite basse = - 18
Pente [mg/L.min]	Limite haute = 0,04 Limite basse = - 0,04
Déviation standard [mg/L]	Limite haute = 0,4 Limite basse = - 0,4
Plage des valeurs [mg/L]	Limite haute = 25 Limite basse = 0,5

En termes de résultats, la Figure 61 représente les données brutes et traitées des nitrates. Il est observé une détection et un remplacement des données aberrantes causées majoritairement par les nettoyages du capteur. En agrandissant sur une période (Figure 62), les mêmes idées sont apportées telles que la conservation et le lissage de la variation des nitrates et la suppression des données aberrantes les plus flagrantes. Finalement, le Tableau 20 expose les pourcentages de données aberrantes et rejetées pour l'ensemble de la série traitée. Dans ce cas, 1,6 % des données ont été rejetées. Lors du prétraitement de cette variable, 10 % des données avaient été rejetées manuellement sur la période d'après le « logbook » indiquant par exemple des défaillances du capteur. Ainsi, un total d'environ 12 % de données a été enlevé de la série brute comportant 542 881 de données.

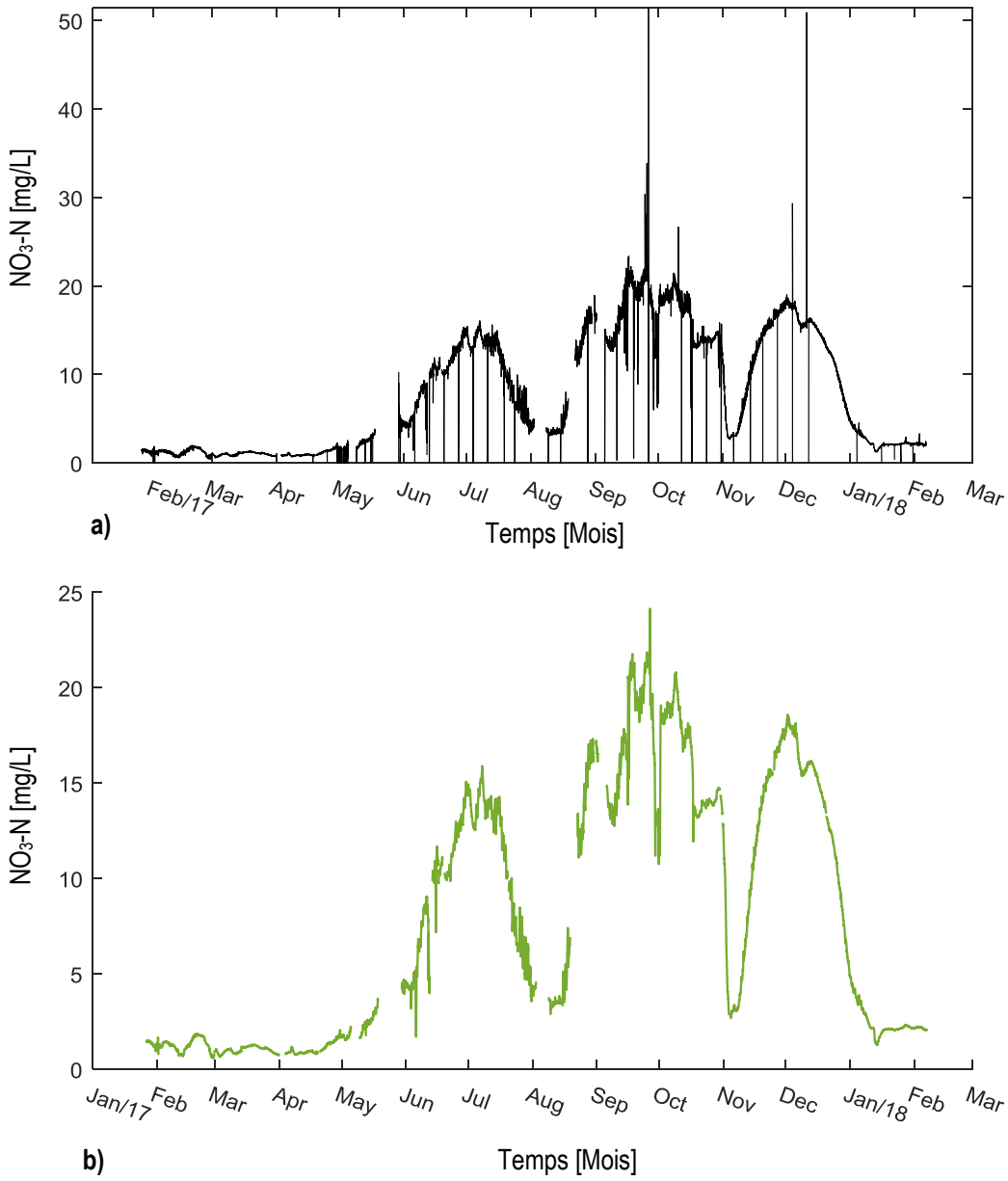


Figure 61. Données brutes et traitées de la variable nitrate mesurée à l'effluent d'un étang aéré KAMAK sur une période d'un an a) Données brutes b) Données traitées

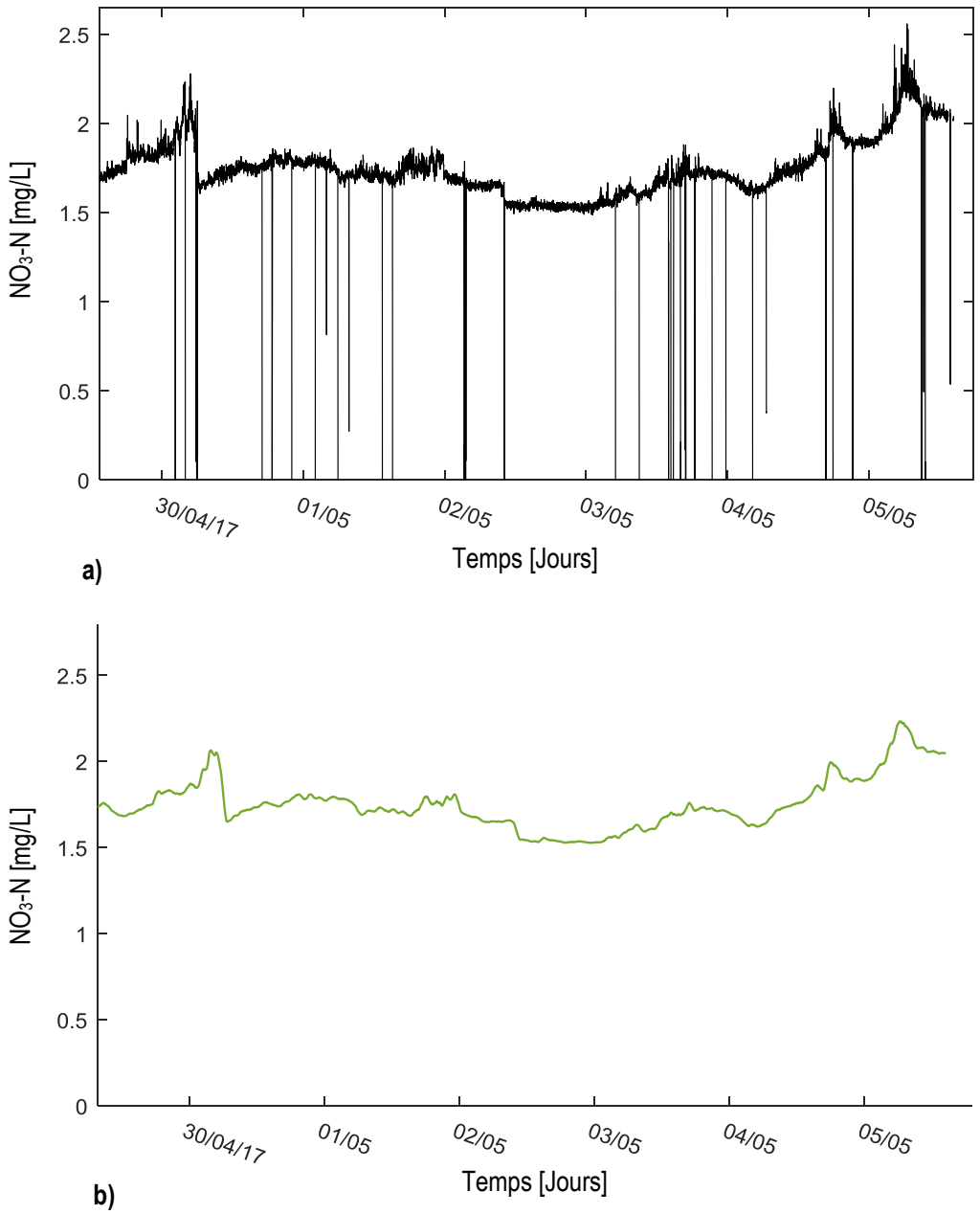


Figure 62. Agrandissement sur les données brutes et traitées de la variable nitrate mesurées à l'effluent d'un étang aéré KAMAK sur une période de sept jours (détail de la Figure 61) a) Données brutes b) Données traitées

Tableau 20. Pourcentage des données aberrantes et rejetées pour la variable nitrate mesurée à l'effluent d'un étang aéré KAMAK

Paramètres	*NO_3-N*
Nombre de données	542 879
% de données aberrantes	3,5 %
% de données rejetées	1,6 %

Ensuite, pour la variable oxygène dissous, la détection des données aberrantes a été réalisée en modifiant la valeur du paramètre « param.nb_reject ». Elle est égale à celle utilisée pour la variable précédente (Tableau 21). De nouveau sa valeur est fonction des durées de nettoyages manuels effectués par l'opérateur.
Concernant le lissage des données, la valeur du paramètre « param.h_smoother » est plus importante et égale à 240. Ceci a permis de lisser les données toutes les quatre heures et ainsi de prendre en compte les grandes variations et non les petites. Il a aussi eu pour but de réduire le bruit important dans la série de données brutes. Enfin, pour l'étape de détection des fautes, le Tableau 22 quantifie les limites des indicateurs de défaillances employés. La valeur limite basse du paramètre « plage des valeurs » est initialisée à - 0.1 mg/L afin de ne pas rejeter les concentrations d'oxygène dissous égale à 0 mg/L comme étant des données valides.

Tableau 21. Valeurs modifiées des paramètres modifiés pour la variable d'oxygène dissous à l'effluent d'un étang aéré KAMAK

Paramètres	***Valeurs par défaut***	***Nouvelles valeurs***
param.nb_reject	90	20
param.h_smoother	30	240

Tableau 22. Limites des indicateurs de défaillances pour la détection de fautes dans la série de données de la variable d'oxygène dissous mesurée à la sortie d'un étang aéré KAMAK

Indicateurs de défaillances	**Limites**
Signe run-test	Limite haute = 4 Limite basse = - 23
Pente [mg/L.min]	Limite haute = 0,0015 Limite basse = - 0,0014
Déviation standard [mg/L]	Limite haute = 8 Limite basse = - 10
Plage des valeurs [mg/L]	Limite haute = 13 Limite basse = - 0,1

La Figure 63 montre les données traitées d'oxygène dissous collectées à l'effluent d'un étang aéré KAMAK sur une année. En agrandissant sur une période, le signal brut est aussi conservé et lissé afin de réduire le bruit. Les données aberrantes représentées par les pics élevés sont aussi écartées du signal final (Figure 63). Le Tableau 23 montre le pourcentage des données aberrantes et rejetées. De nouveau, relativement peu de valeurs ont été rejetées, avec un pourcentage équivalent à 1.1 %. Lors du prétraitement, le pourcentage de données rejetées avait été de 16 %. Au vu d'un pourcentage déjà important lors de ce prétraitement, le traitement univarié a permis de minimiser la perte de données additionnelles tout en donnant le résultat voulu.

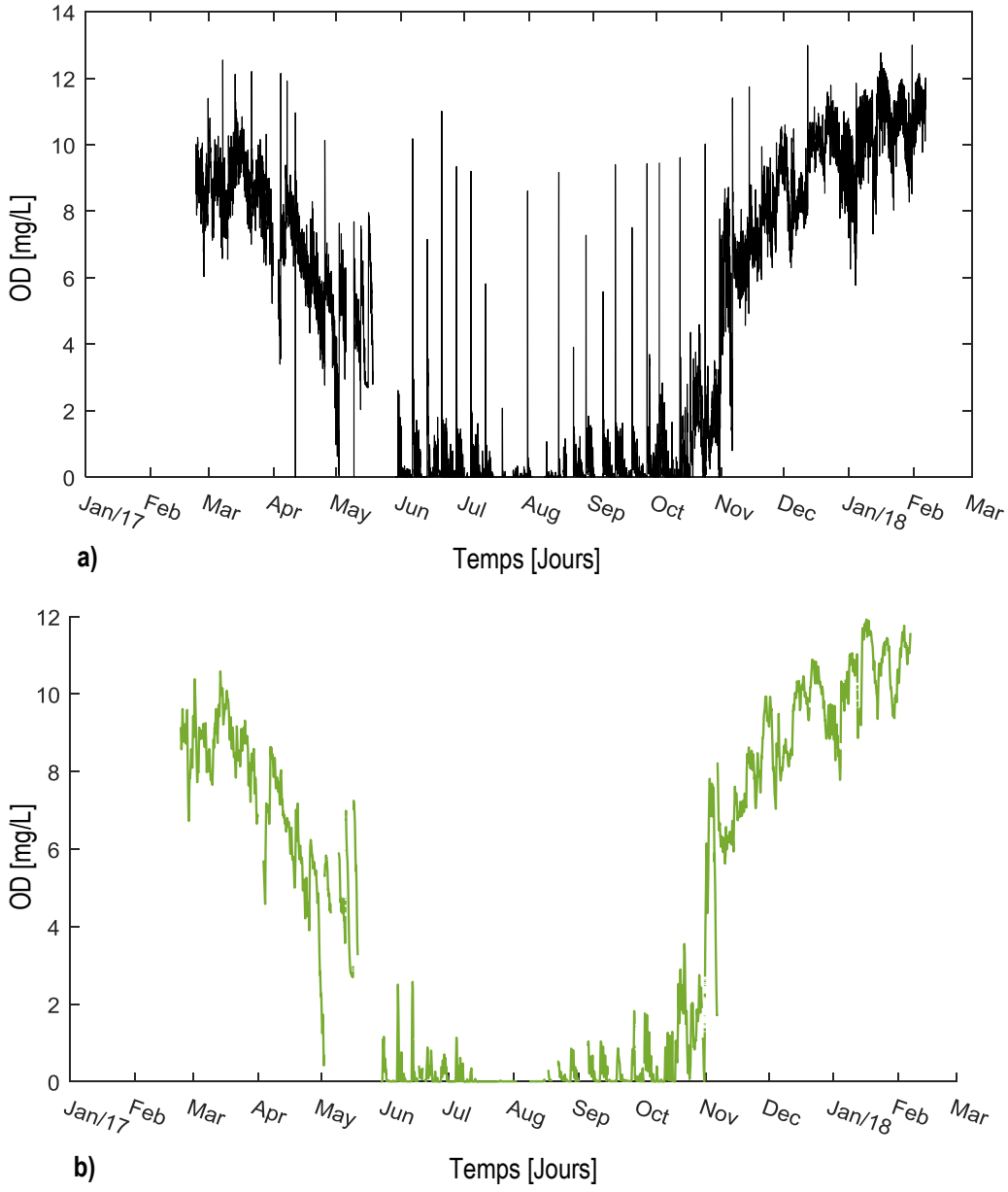


Figure 63. Données brutes et traitées de l'oxygène dissous à l'effluent d'un étang aéré KA-MAK sur une période d'un an a) Données brutes b) Données traitées

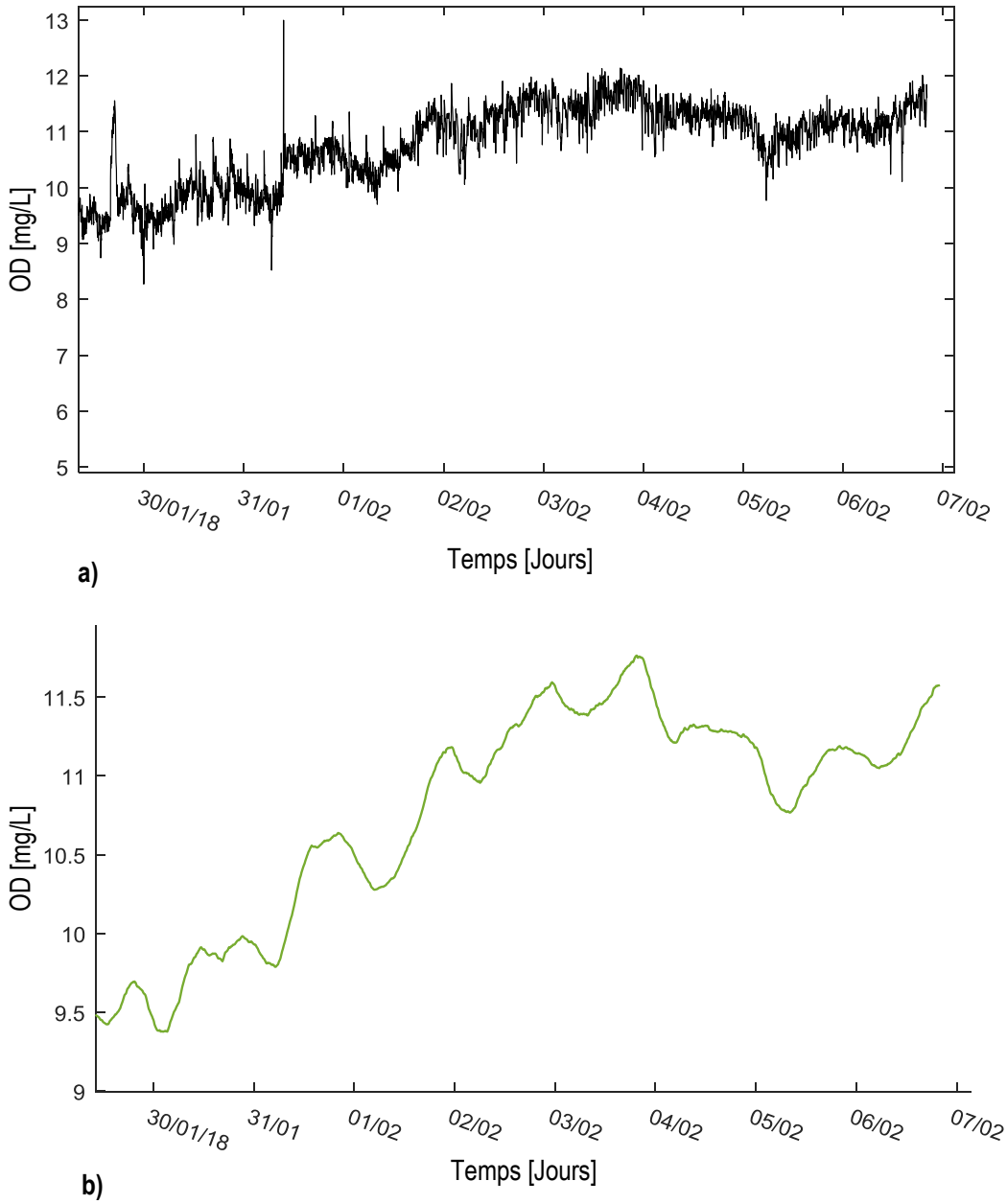


Figure 64. Agrandissement sur les données brutes et traitées de l'oxygène dissous à l'effluent d'un étang aéré KAMAK sur une période de neuf jours (détail de la Figure 63) a) Données brutes b) Données traitées

Tableau 23. Pourcentage des données aberrantes et rejetées pour l'oxygène dissous à l'effluent d'un étang aéré KAMAK

Paramètres	*Oxygène dissous*
Nombre de données	542 879
% de données aberrantes	2.0 %
% de données rejetées	1,1 %

c. Bord*EAU*x

La méthode a aussi été appliquée sur les données d'une campagne d'échantillonnage menée durant l'été 2017 pour le projet bord*EAU*x. Ces données collectées serviront à la calibration d'un modèle intégré afin d'optimiser simultanément la gestion du réseau d'égout et de la StaRRE (section 3.1.3). Afin d'obtenir des données de bonne qualité, les données brutes devaient être traitées par la méthode univariée afin d'éliminer les données aberrantes et les fautes de capteurs. Le site d'étude comporte deux points de mesures, le réseau d'égout et l'entrée de la StaRRE. Les points importants pris en compte dans le traitement ont été la conservation du patron journalier des eaux usées ainsi que l'élimination d'un minimum de données. Pour chaque illustration montrée, une quantification des paramètres et des limites des indicateurs est exposée afin de donner des pistes d'aide aux futurs utilisateurs.

o *Réseau d'égout*

Deux variables mesurées dans le réseau d'égout sont données par les Figure 65 et Figure 67 : les MES et la température. Tout d'abord, pour les MES, une modification des paramètres, par défaut de la méthode de détection des données aberrantes, a été opérée dans le but de rejeter les données aberrantes les plus évidentes tout en conservant le patron journalier (Tableau 24.). Le paramètre « param.nb_reject » a été ajusté afin de réinitialiser la méthode de détection des données aberrantes après 60 données, correspondant à un nettoyage de capteur d'une durée de 60 minutes. Pour la deuxième étape de la méthode, la détection des fautes, le Tableau 25 montre les limites des indicateurs de défaillances qui permettent de détecter les fautes au sein de la série de données.

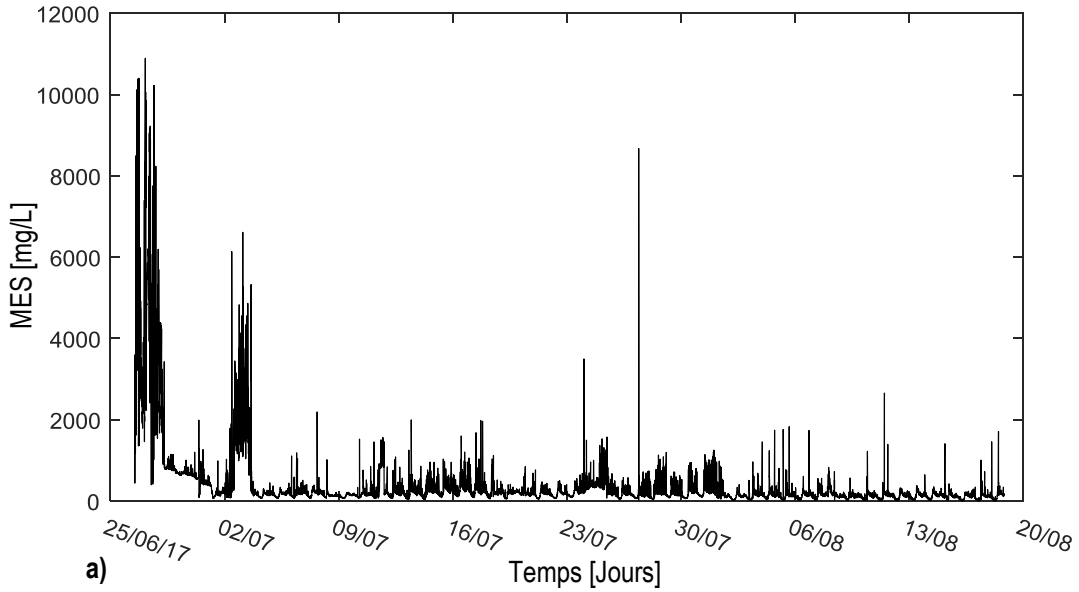
Tableau 24. Nouvelles valeurs des paramètres modifiés pour la méthode de filtrage des données pour la variable MES dans le réseau d'égout à Bordeaux

Paramètres	***Valeurs par défaut***	***Nouvelles valeurs***
param.nb_reject	100	60

Tableau 25. Limites des indicateurs de défaillances pour la détection de fautes dans la série de données de la variable MES mesurée dans le réseau d'égout à Bordeaux

Indicateurs de défaillances	**Limites**
Signe run-test	Limite haute = - 2 Limite basse = - 20
Pente [mg/L.min]	Limite haute = 2 Limite basse = - 2
Déviation standard [mg/L]	Limite haute = 0,5 Limite basse = - 0,5
Plage des valeurs [mg/L]	Limite haute = 1000 Limite basse = 0

La Figure 65 montre un exemple de résultat du signal traité par la méthode univariée. Premièrement, une amélioration de la qualité du signal peut être observée avec la suppression des données aberrantes comme étant des colmatages du capteur (Figure 65). Mais, sur cette même figure, trois périodes (cercles en pointillés rouges) montrent la limitation de la méthode avec des données qui n'ont pas été rejetées. Dans le futur, ces dernières pourraient être éliminées en effectuant des travaux additionnels sur l'étape de la détection des fautes en y ajoutant de nouveaux indicateurs de défaillances. La Figure 66 illustre un zoom d'une période de la campagne de mesure. La même conclusion est montrée avec une amélioration de la qualité du signal dont la suppression des données aberrantes représentées par les pics élevés. Le point primordial pour l'utilisatrice finale, Julia Ledergerber, a été la conservation du cycle journalier de l'eau usée comme étant important pour la calibration de son modèle.



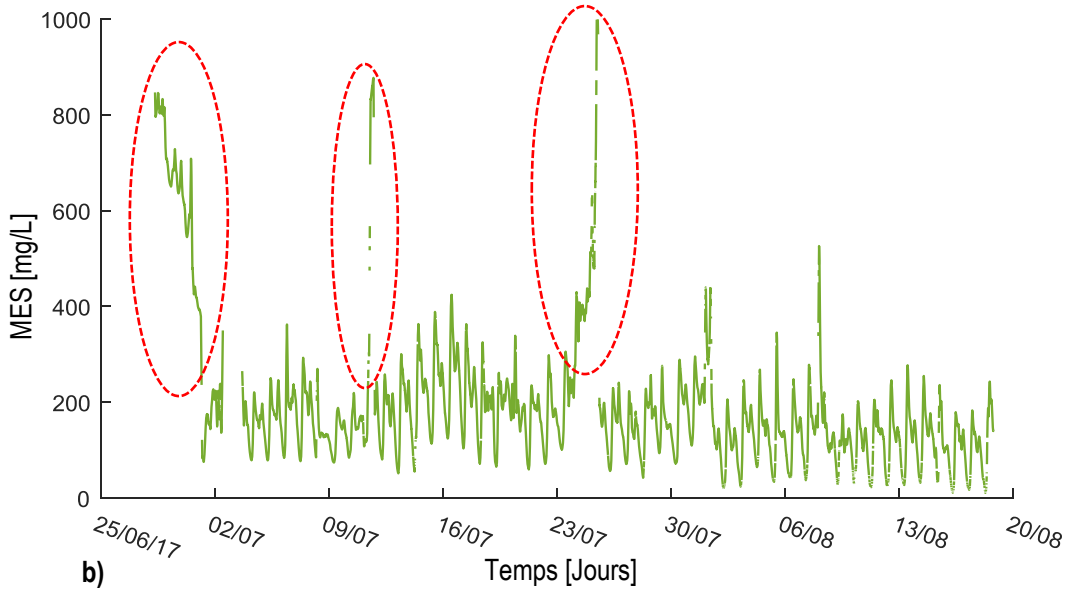
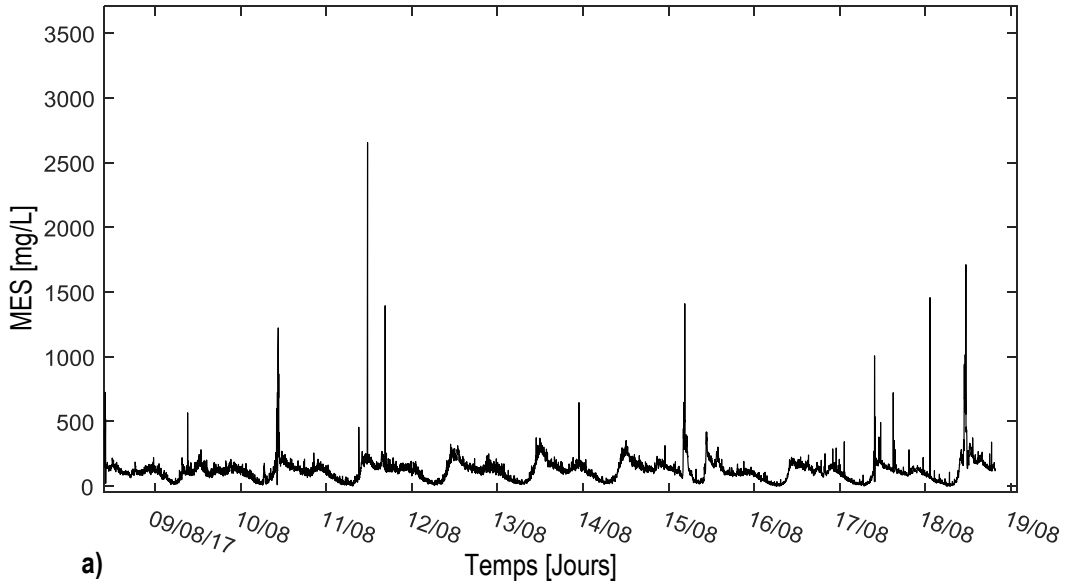


Figure 65. Données brutes et traitées de la variable MES mesurée dans le réseau d'égout sur une période de deux mois à Bordeaux a) Données brutes b) Données traitées



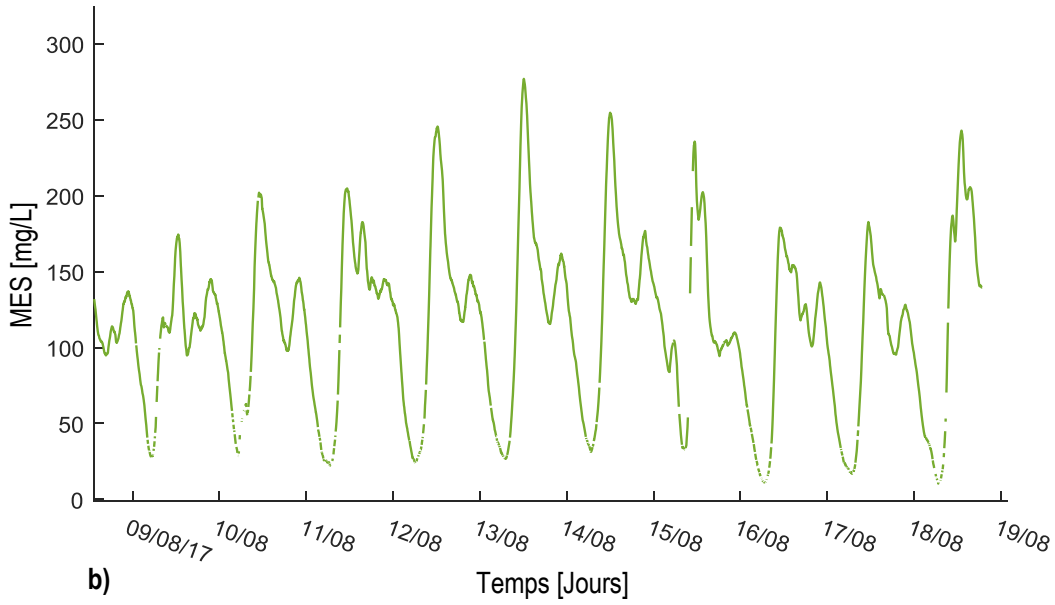


Figure 66. Agrandissement sur les données brutes et traitées des MES mesurées dans le réseau d'égout sur une période d'onze jours à Bordeaux (détail de la Figure 65) a) Données brutes b) Données traitées

Le Tableau 26. quantifie les pourcentages des données aberrantes et rejetées. Pour ces périodes, 18 % de données aberrantes ont été détectées et 12 % de données finales ont été enlevées. La majorité des données aberrantes correspondent au colmatage du capteur et aux périodes de nettoyage du capteur (deux fois par semaine). Les données rejetées sont retrouvées en grande partie au début de la campagne avec des difficultés de calibration du capteur et un fort colmatage de ce dernier d'après le cahier de bord du suivi des capteurs. Ces diverses informations avaient été discutées avec la doctorante lors d'une première réunion avant l'utilisation de la méthode.

Tableau 26. Pourcentage des données aberrantes et rejetées pour les MES mesurées dans le réseau d'égout à Bordeaux

Paramètres	*MES*
Nombre de données	38 347
% de données aberrantes	18 %
% de données rejetées	12 %

Ensuite, la Figure 67 expose des données de température traitées par la méthode. Pour cette variable, les paramètres de la méthode univariée ont été modifiés et quantifiés dans le Tableau 27. Le paramètre « param.nb_s » a été augmenté dans le but de détecter les données aberrantes les plus évidentes et de conserver

le patron journalier. La valeur du paramètre « param.nb_reject » a aussi été implanté à 250 dans le but de détecter les nettoyages de capteur d'une durée d'environ de 30 minutes. Ceci se base sur une fréquence d'enregistrement des données toutes les 5 secondes. Le paramètre « param.h_smoother » a été rectifié à 120 points afin de lisser les données toutes les 10 minutes. Pour la détection des fautes, le
Tableau 28. expose les limites de chaque indicateur de défaillances afin de détecter les fautes au sein de la série de données. Les mêmes conclusions que pour les cas précédents peuvent être ressorties tels qu'une amélioration de la qualité du signal, une élimination des données aberrantes les plus visibles et des pourcentages de données rejetées relativement faibles (Tableau 29). Les pics élevés éliminés représentent une mauvaise connexion entre le capteur et la station mon*EAU*.

Tableau 27. Nouvelles valeurs des paramètres modifiés pour la méthode de filtrage des données pour la variable température mesurée en réseau d'égout à Bordeaux

Paramètres	***Valeurs par défaut***	***Nouvelles valeurs***
param.nb_s	3	7
param.nb_reject	100	250
Param.h_smoother	30	120

Tableau 28. Limites des indicateurs de défaillances pour la détection de fautes dans la série de données de la variable température mesurée dans le réseau d'égout à Bordeaux

Indicateurs de défaillances	**Limites**
Signe run-test	Limite haute = - 12 Limite basse = - 23
Pente [°C/min]	Limite haute = 0,002 Limite basse = - 0,002
Déviation standard [°C]	Limite haute = 0,02 Limite basse = - 0,02
Plage des valeurs [°C]	Limite haute = 22,5 Limite basse = 19

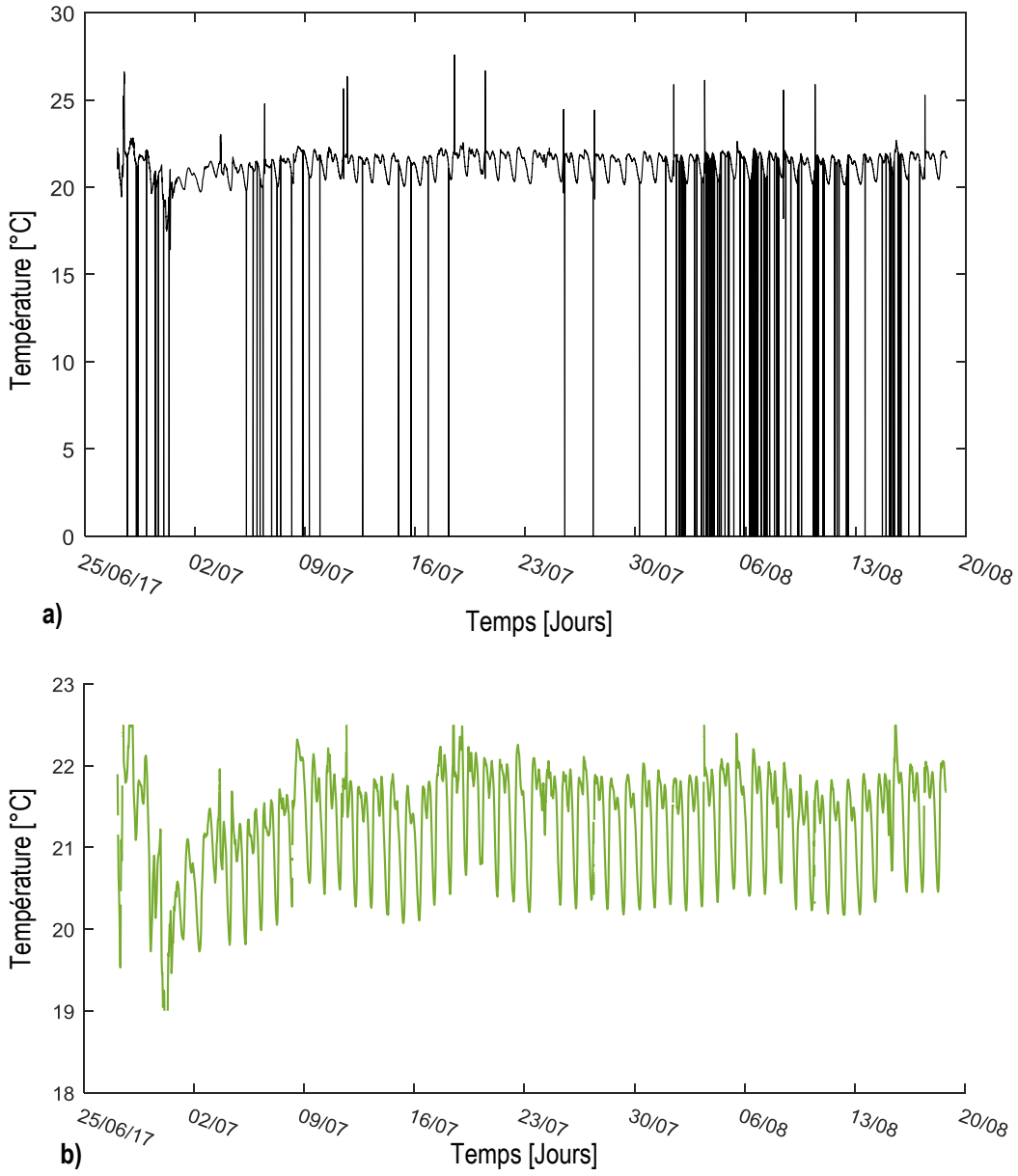


Figure 67. Données brutes et traitées de température mesurée dans le réseau d'égout sur une période de deux mois à Bordeaux a) Données brutes b) Données traitées

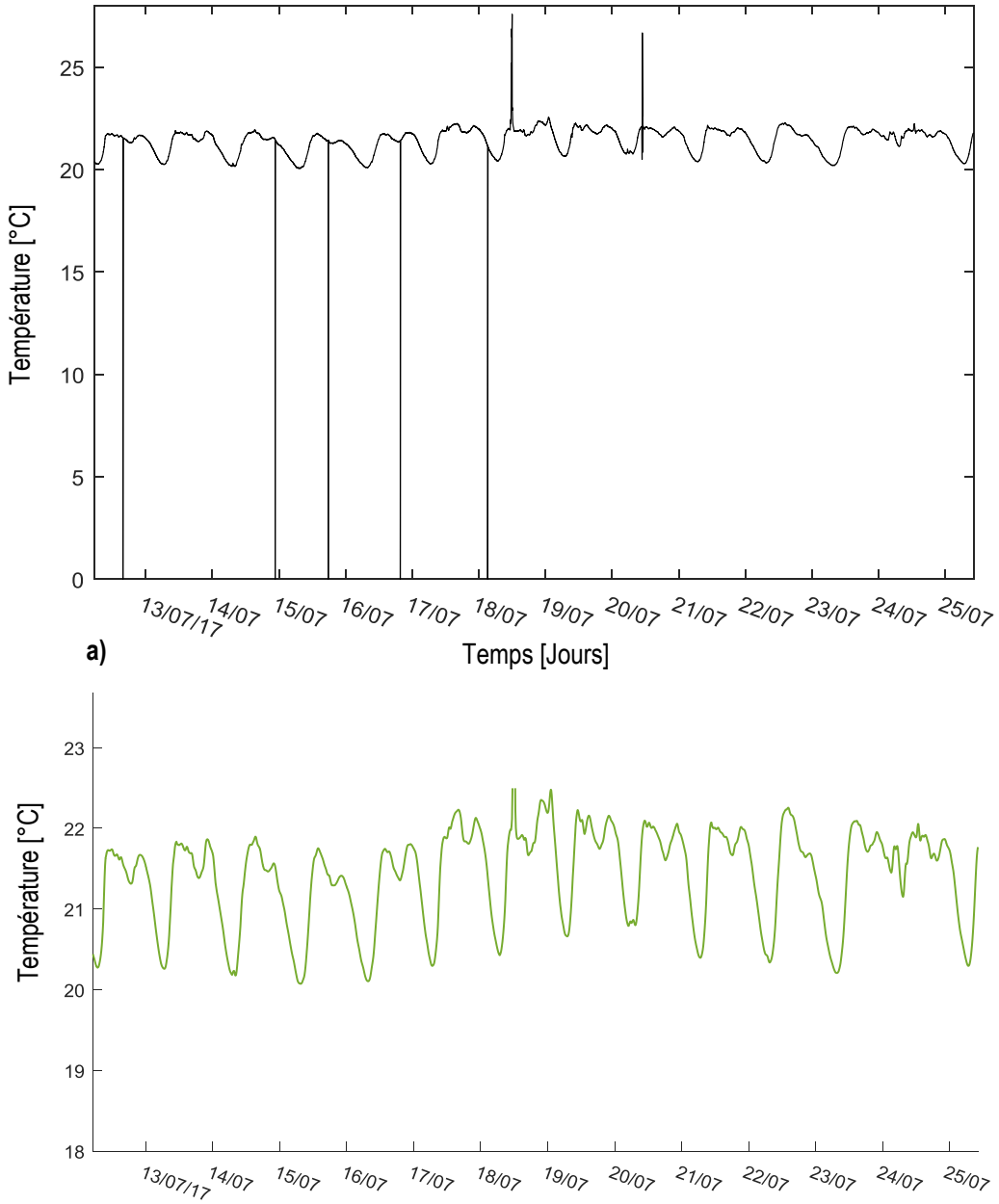


Figure 68. Agrandissement des données brutes et traitées de température mesurée dans le réseau d'égout à Bordeaux (détail de la Figure 67) a) Données brutes b) Données traitées

Tableau 29. Pourcentage des données aberrantes et rejetées pour la température mesurée dans le réseau d'égout à Bordeaux

Paramètres	*Température*
Nombre de données	910 785
% de données aberrantes	5 %
% de données rejetées	1,5 %

o *Entrée de la station*

Comme pour le point précédent, deux exemples d'illustrations sont donnés pour les variables DCO et pH mesurées à l'entrée de la StaRRE de Bordeaux. Comme pour les cas précédents, les paramètres par défaut ont été ajustés. Les paramètres « param.h_smoother » et « param.nb_reject » ont été diminués afin de conserver le patron journalier tout en même temps, de rejeter les données aberrantes les plus évidentes (colmatage du capteur) (Tableau 30.). En effet, une valeur de 60 correspond à un nettoyage de capteur d'une durée de 30 minutes pour une fréquence d'enregistrement égale à 2 minutes. Pour l'étape de la détection des fautes, le Tableau 31. expose les limites de chaque indicateur de défaillances.

Tableau 30. Valeurs des paramètres modifiés pour la méthode de filtrage des données pour la variable DCO mesurée à l'entrée de la StaRRE à Bordeaux

Paramètres	***Valeurs par défaut***	***Nouvelles valeurs***
param.nb_reject	100	60
param.h_smoother	30	5

Tableau 31. Limites des indicateurs de défaillances pour la détection de fautes dans la série de données de la variable DCO mesurée à l'entrée de la StaRRE à Bordeaux

Indicateurs de défaillances	**Limites**
Signe run-test	Limite haute = 15 Limite basse = - 15
Pente [mg/L.min]	Limite haute = 10 Limite basse = - 10
Déviation standard [mg/L]	Limite haute = 1 Limite basse = - 1
Plage des valeurs [mg/L]	Limite haute = 1500 Limite basse = 0

En outre, la Figure 69b montre le résultat final des données traitées. Sur cette figure, les données aberrantes ont été enlevées et une amélioration du signal comparée au signal brut a été obtenue sur cette période. Cependant, certaines données (cercles rouges) n'ont pas été rejetées par la méthode. Elles montrent la limitation de la méthode déjà évoquée précédemment. Dans le but de détecter l'ensemble des fautes, des travaux additionnels avec l'ajout de nouveaux indicateurs de défaillances pourront être effectués. Enfin, quelques informations peuvent découler de ces résultats, tels que les pourcentages de données aberrantes et des données rejetées exposés dans le Tableau 32.. Pour cette période de 4 mois, le pourcentage de données aberrantes est équivalent à 10 % et 7 % pour les données rejetées.

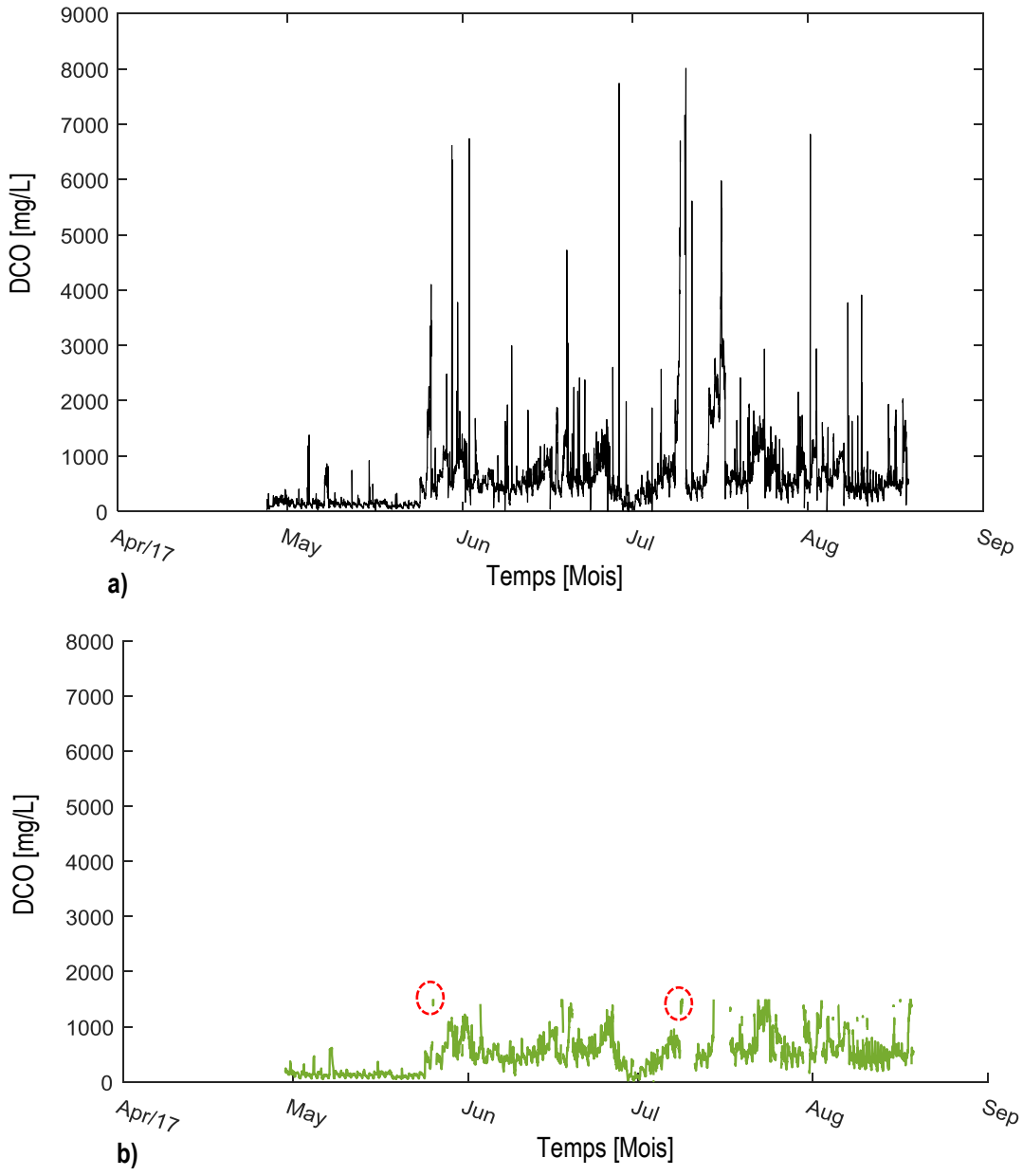


Figure 69. Données brutes et traitées de la DCO mesurée à l'entrée de la StaRRE sur une période de trois mois à Bordeaux a) Données brutes b) Données traitées

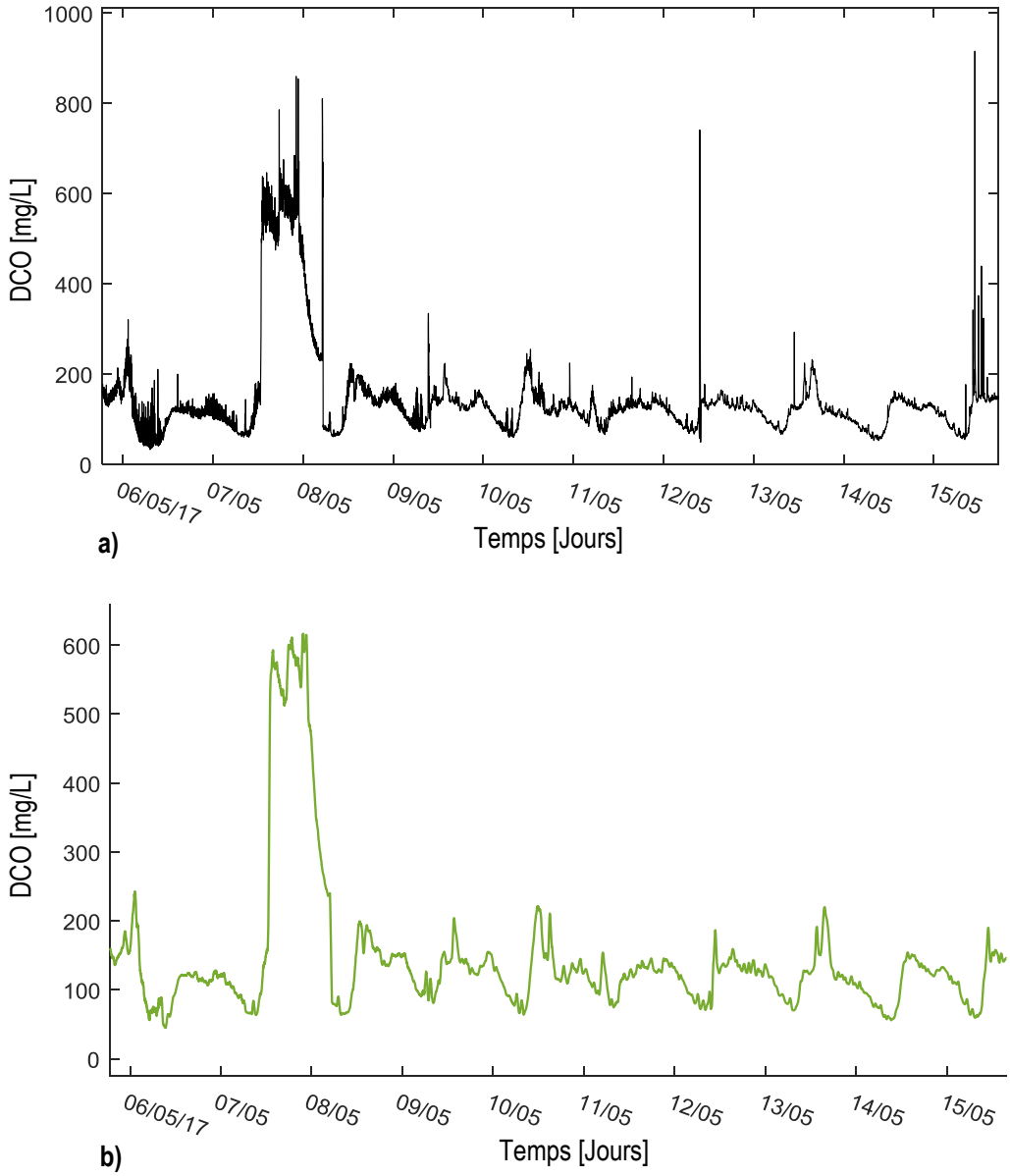


Figure 70. Agrandissement sur les données brutes et traitées de la DCO mesurée à l'entrée de la StaRRE de Bordeaux sur une période de neuf jours (détail de la Figure 69) a) Données brutes b) Données traitées

Tableau 32. Pourcentage des données aberrantes et rejetées pour la DCO mesurée à l'entrée de la StaRRE de Bordeaux

Paramètres	*DCO*
Nombre de données	73 421
% de données aberrantes	10 %
% de données rejetées	7 %

Un deuxième exemple illustré par la Figure 71 expose les mêmes conclusions que pour le cas ci-dessus tels qu'une conservation du signal brut avec en même temps une élimination des données aberrantes pour la variable pH.

Tout comme les variables précédentes, les paramètres par défaut de la méthode univariée ont été ajustés (Tableau 33). Les valeurs des paramètres « param.nb_reject » et « param.h_smoother » sont importantes afin de rejeter les nettoyages des capteurs et de minimiser le bruit dans la série de données. En effet, un « param.nb_reject » élevé permet au modèle de ne pas se réinitialiser durant les nettoyages. Ici, puisque le pas de temps est de 5 secondes, le « param.nb_reject » a été implanté à une valeur de 720 correspondante à la durée d'un nettoyage (3600 secondes). Le même ordre idée est appliquée pour le paramètre « param.h_smoother ». Le lissage prend donc en compte une période de 10 minutes équivalente à 120 données. Enfin, la perte de données est relativement faible avec 3,8 % de données rejetées, comparée à un total de données d'environ 2 millions (Tableau 35).

Tableau 33. Nouvelles valeurs des paramètres modifiés pour la méthode de filtrage de données pour la variable pH mesurée à l'entrée de la StaRRE à Bordeaux

Paramètres	***Valeurs par défaut***	***Nouvelles valeurs***
param.nb_reject	100	720
Param.h_smoother	30	120

Tableau 34. Limites des indicateurs de défaillances pour la détection de fautes dans la série de données de la variable pH mesurée à l'entrée de la StaRRE à Bordeaux

Indicateurs de défaillances	**Limites**
Signe run-test	Limite haute = 20 Limite basse = - 20
Pente [1/min]	Limite haute = 0,001 Limite basse = - 0,001
Déviation standard [-]	Limite haute = 0,02 Limite basse = - 0,02
Plage des valeurs [-]	Limite haute = 8 Limite basse = 7

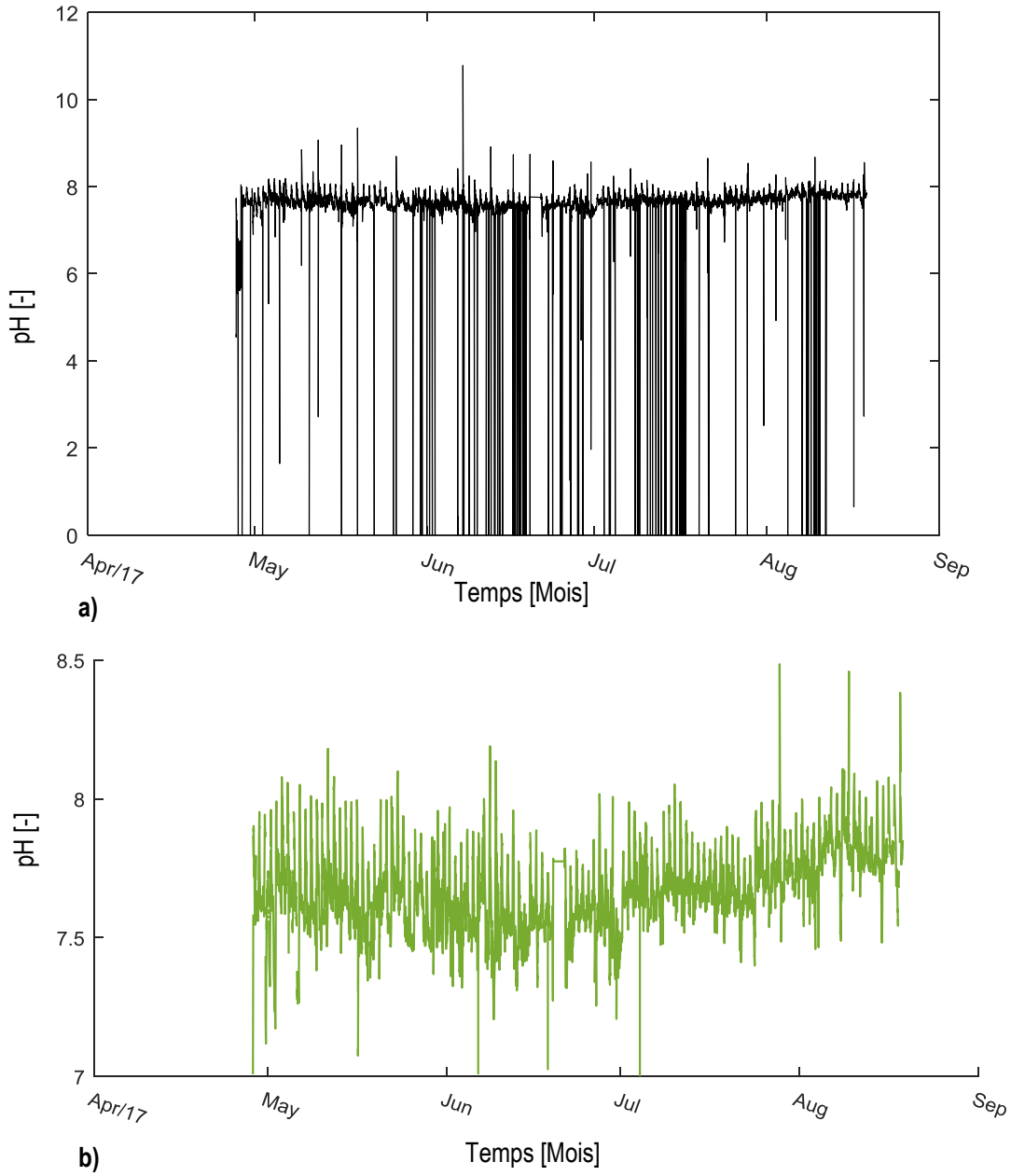


Figure 71. Données brutes et traitées de la variable pH mesurée à l'entrée de la StaRRE sur une période de trois mois a) Données brutes b) Données traitées

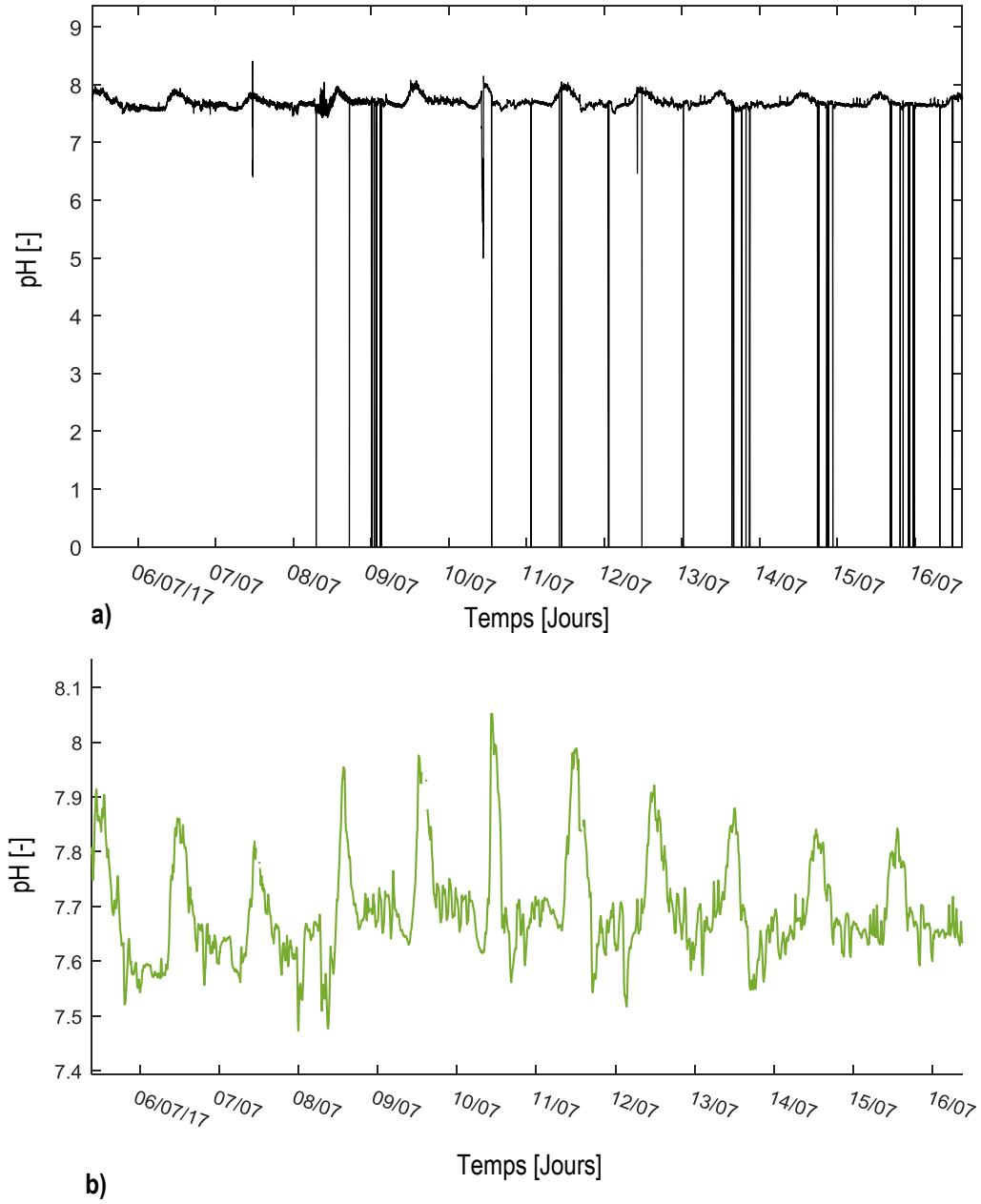


Figure 72. Agrandissement sur les données brutes et traitées de pH mesuré à l'entrée de la StaRRE à Bordeaux (Détail de la Figure 71) a) Données brutes b) Données traitées

Tableau 35. Pourcentage des données aberrantes et rejetées pour la variable pH mesurée à l'entrée de la StaRRE de Bordeaux

Paramètres	*pH*
Nombre de données	1 923 054
% de données aberrantes	10 %
% de données rejetées	3,8 %

4.2.1.2. Conclusion

Trois cas d'étude ont illustré l'applicabilité de la méthode univariée : la conservation de l'information initiale et l'élimination des données aberrantes tout en conservant le maximum de données. Les pourcentages de données rejetées par la méthode univariée (0,1 à 12%) étaient inférieurs à ceux retrouvés dans la littérature : 5 à 60 % (Alferes et al., 2013a). Ce dernier point est très essentiel pour les utilisateurs finaux. Pour chaque illustration, une quantification des paramètres modifiés pour le filtrage et des limites des indicateurs de défaillances, ont été exposée. Cette quantification permettra de donner des pistes d'aide aux futurs utilisateurs de l'outil et de simplifier son utilisation. Enfin, pour certaines illustrations, la limitation de la méthode a été observée. L'ensemble des données aberrantes et des fautes de capteurs n'ont pas toutes été supprimées. Des travaux futurs pourront être effectués afin d'ajouter des indicateurs de défaillances dans la détection des fautes.

4.2.2. Méthode multivariée

La méthode multivariée par analyse des composantes principales a pour but de détecter des fautes au sein de série de données de plusieurs capteurs, par exemple les dérives (section 1.3). Cette méthode peut aussi être un complément à la méthode univariée car elle s'applique typiquement après le filtrage des données (détection des données aberrantes et lissage des données) (Figure 42). Alferes et al. (2013b) avaient développé et utilisé la méthode multivariée pour des cas spécifiques. Dans ce projet, cette dernière a été retravaillée dans le but de simplifier et faciliter son utilisation avec la réaction d'un SOP et d'un script général. Des blocs de fonctions ont été aussi créés dans le but de rendre la méthode modulaire (Figure 45). Enfin, dans le même ordre, le format des données est similaire à la méthode univariée sans complexifier le passage d'une méthode à l'autre (section 4.2.1.1).

4.2.2.1. Présentation du script et de ses fonctions

Cette partie présente succinctement le script général de la méthode car ce dernier est développé plus en détails dans le SOP (SOP méthode multivariée). Au début de la méthode, deux étapes préliminaires doivent être effectuées telles que la génération de paramètres et la vérification du même nombre de données. L'étape de la généralisation est réalisée par la fonction « **DefaultsParamPCA** ». Au sein de cette fonction, tous les paramètres sont initialisés à une valeur par défaut. Le Tableau 36 expose l'ensemble des paramètres avec une définition et une valeur par défaut. Pour la vérification du nombre de données, la fonction « **InitializationD** » permet d'accomplir l'étape.

Tableau 36. Paramètres utilisés dans la méthode multivariée et leur valeur par défaut

Paramètres	***Définition***	***Valeur par défaut***
Param.Time	Sélection de l'intervalle de temps pour l'interpolation des données avec la fonction « **InitializationD** »	1/24/60
Param.Xstdmin	Le minimum de la déviation standard	0.01
Param.Normalisation	Choix de normaliser les données	true
Param.p	Un niveau de signification approprié pour effectuer le test. La valeur est généralement de 0,95 ou 0,99	0.95
Param.calfa	La norme normale déviée	2.68
Param.alfa	Ce paramètre « alfa » correspond au niveau de confiance	0.99
Param.n	Nombre de variables dans l'analyse PCA	[1 : 14]
Param.a	Composantes principales choisies	[1 2 3]

Ayant effectué les deux étapes préliminaires, « la construction du modèle ACP » peut être réalisée en choisissant tout d'abord des données ayant un comportement normal. Ceci est fait à l'aide de la fonction « **SelectTime** ». Par la suite, un choix de normalisation des données sélectionnées est alors offert à l'utilisateur (section b). Enfin, en ayant choisi le nombre de composantes principales à l'aide de la méthode graphique et de la fonction « **plotpar** », la fonction « **ModelPCA** » permet bien entendu la construction du modèle ACP. La dernière sous-étape est la détermination automatique des deux tests T^2 et Q et leurs limites par les deux fonctions « **StatTest** » et « **LimitTest** » (section d). Pour chaque fonction explicitée précédemment, des explications sont apportées dans le SOP en Annexe (SOP méthode multivariée). Ceci a pour but d'aider et simplifier l'utilisation de la méthode par des futurs utilisateurs.

Ensuite, la deuxième principale étape est la détection de fautes dans de nouvelles séries de données. Dans ce but, une projection des nouvelles données sur le modèle ACP est réalisée. Cependant, les sous-étapes du choix de données et de normalisation de celles-ci doivent être aussi accomplies à l'aide des mêmes fonctions « **SelectTIME** » et « **NormalisationD** ». Par la suite, la projection est effectuée par la fonction « **StastisticTest** » pour calculer la valeur des tests T^2 et Q. Enifn, la fonction « **plotQandT2** » permettant graphiquement d'observer les périodes où des fautes peuvent être détectées. Ces périodes sont identifiables lorsqu'un ou deux tests sont au-dessus des limites calculées lors de la construction du modèle ACP. Comme pour la construction du modèle ACP, les fonctions pour la détection des fautes sont aussi expliquées dans le SOP (But, entrées, sorties).

4.2.2.2. Applications

La méthode multivariée a été illustrée sur les données provenant du site d'étude pil*EAU*te pour trois cas d'étude : à la sortie du décanteur primaire, dans le bioréacteur et l'utilisation de la méthode en ligne. Ces illustrations permettent de montrer la modularité de la méthode et aussi aider les futurs utilisateurs de l'outil.

a. Sortie décanteur primaire

Pour cette première illustration, les données proviennent de capteurs installés en sortie du décanteur primaire du pil*EAU*te. Le nombre de variables pris en compte pour cette illustration est au nombre de sept, soit les DCO totale et soluble, les MES, NH_4, K, pH et la température. La période totale d'analyse des séries de données est comprise entre le mois de janvier et avril 2018. Cependant, avant d'analyser ces données par la méthode multivariée, un pré-traitement de chaque série temporelle a été réalisé par la méthode univariée en filtrant ces dernières. Dans cette filtration, une détection des données aberrantes et un lissage du signal ont été effectués (Alferes and Vanrolleghem, 2016) (Figure 73 et Figure 74). L'étape de détection des fautes n'a pas été prise en compte afin de comparer les résultats entre les deux méthodes.

Comme expliqué dans la partie théorique de la méthode multivariée, la première étape, le développement du modèle ACP commence par la sélection d'une période normale et de la normalisation des données (section b). Cette période s'est arrêtée entre 23 janvier et le 28 janvier 2018 représentée par la Figure 75 et la Figure 76. Aussi, dans la construction de ce modèle, le choix du nombre de composantes principales doit être effectué en utilisant le scree plot à la Figure 77. Ce choix du nombre de CP a été fixé à quatre au vu d'une somme cumulative des valeurs propres égale à 96 % de la variation dans les données. Ayant choisi le nombre de CP, la dernière étape du développement du modèle passe par le calcul automatique des deux tests Q et T^2 ainsi que leurs limites comme exposé dans le chapitre 3 (section d). Les deux tests et leurs limites peuvent être représentés graphiquement (Figure 78).

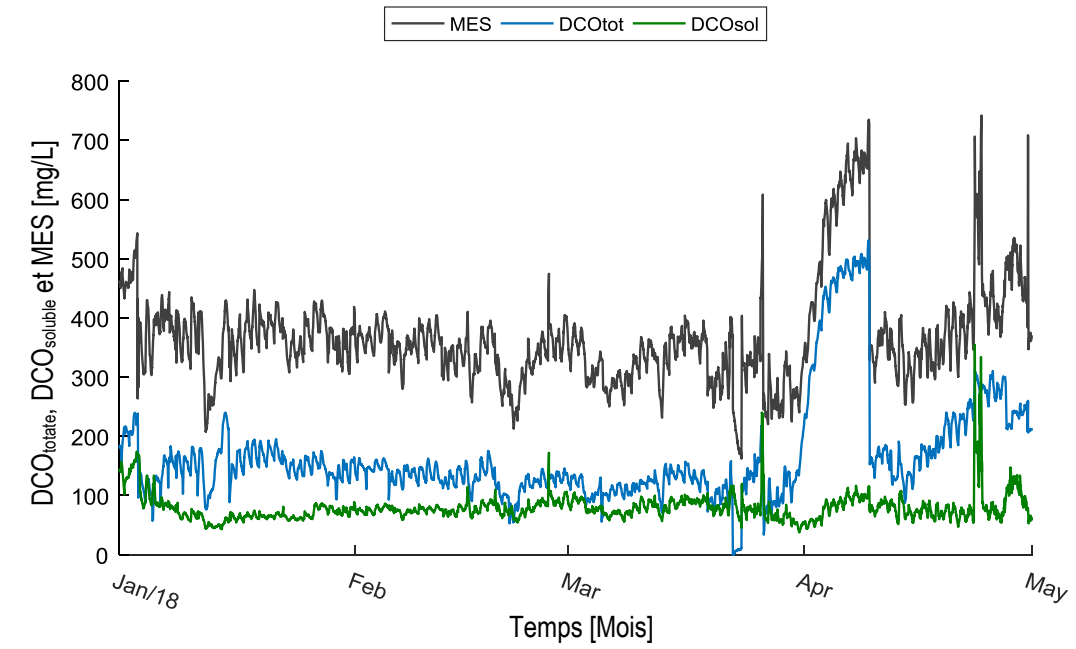


Figure 73. Données prétraitées par la méthode univariée de DCO totale et soluble et MES mesurées en sortie du décanteur primaire au pilEAUte

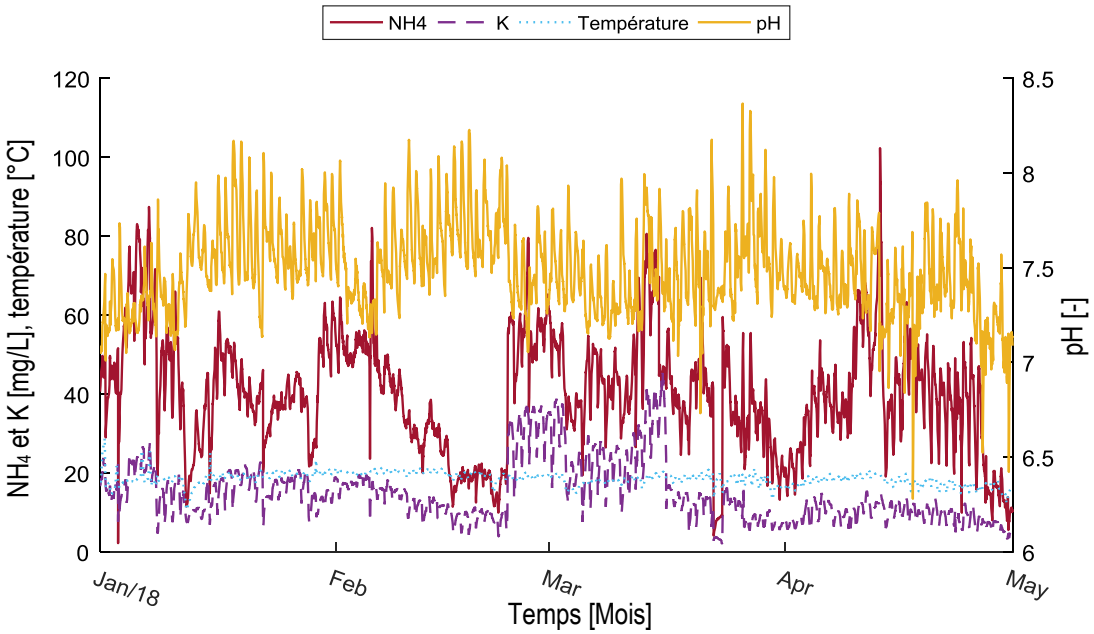


Figure 74. Données prétraitées par la méthode univariée de NH_4, K, température et pH mesurés en sortie du décanteur primaire au pilEAUte

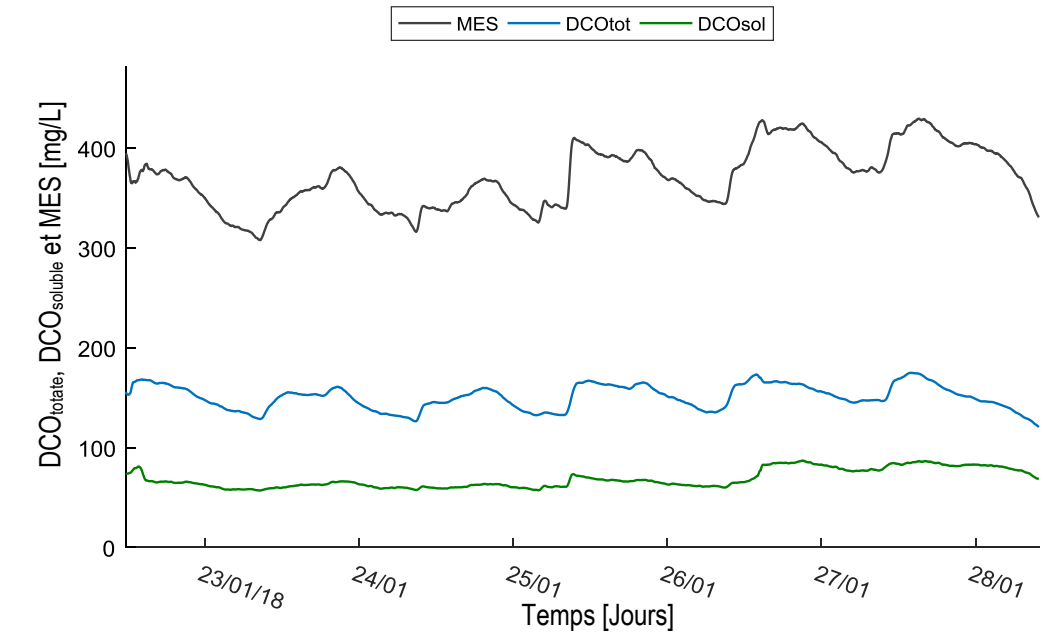


*Figure 75. Données prétraitées par la méthode univariée **normales**, sélectionnées de DCO totale et soluble et MES pour la construction du modèle ACP*

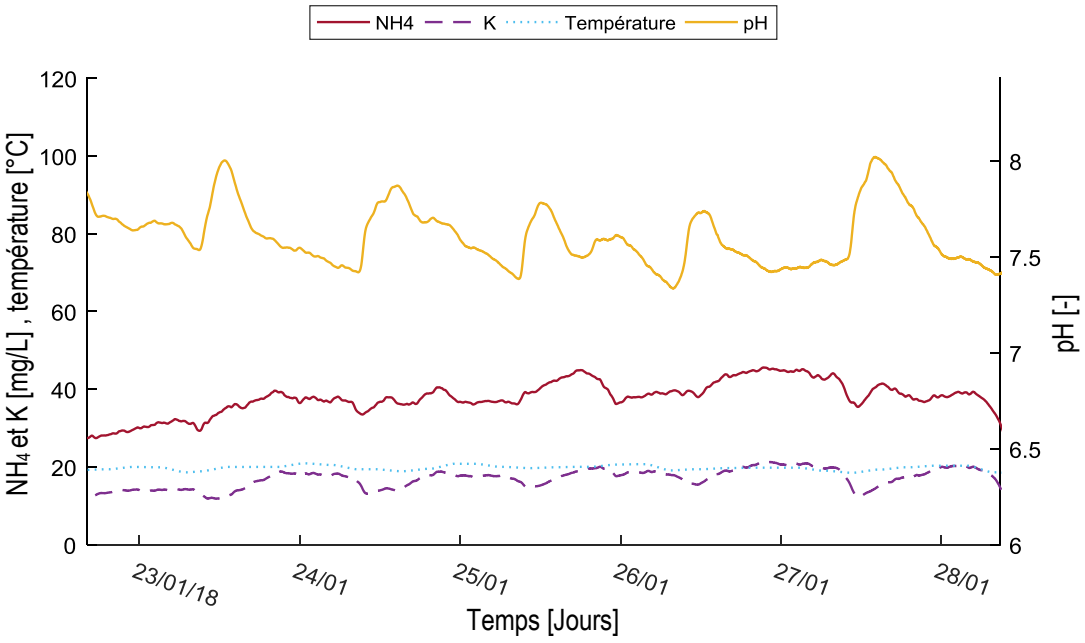


*Figure 76. Données prétraitées par la méthode univariée **normales,** sélectionnées de NH_4, K, température et pH pour la construction du modèle ACP*

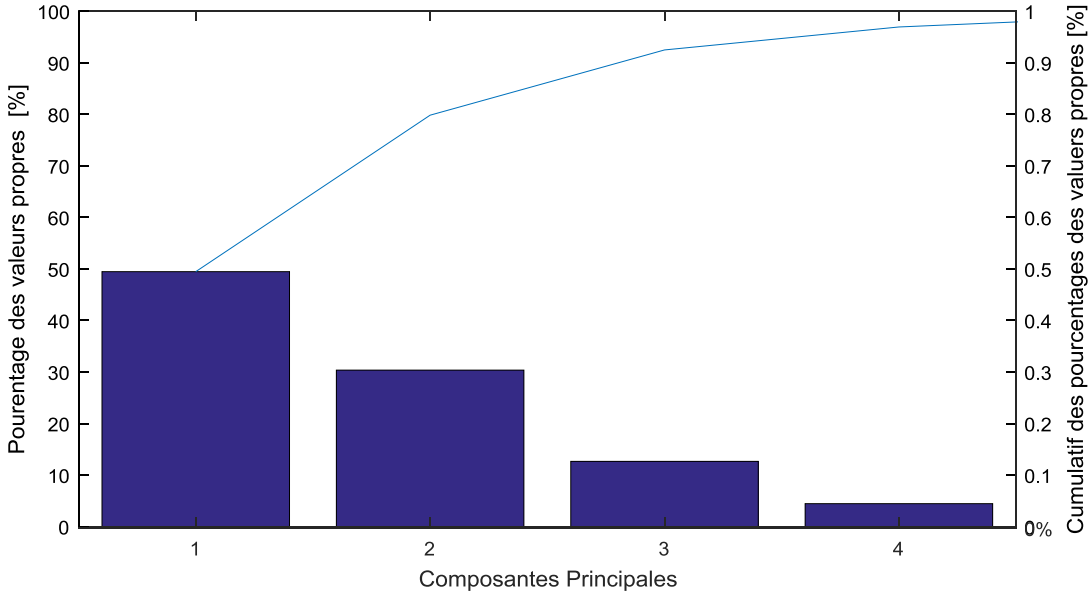


Figure 77. Pourcentages des valeurs propres pour les composantes principales

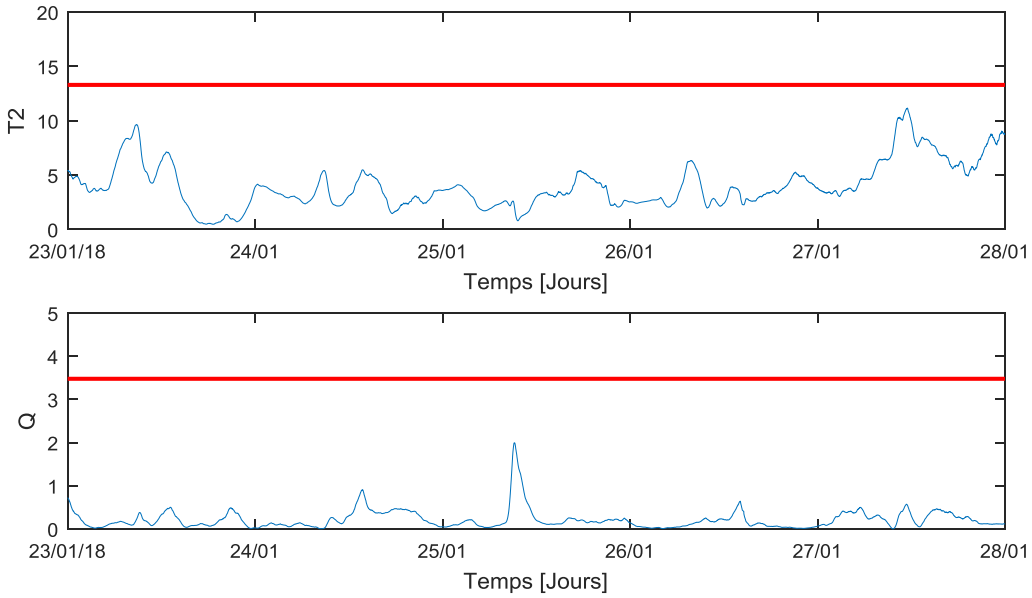


Figure 78. Test Q et T^2 pour les données normales

L'obtention du modèle ACP et des limites des deux tests permettent par la suite de projeter de nouvelles séries de données dans le but de détecter des fautes au sein de celle-ci. Les nouvelles données ont été prises entre le 1er février et le 30 mai 2018. Les deux tests Q et T^2 sont de nouveau évalués automatiquement. La Figure 79 montre leurs valeurs en comparaison avec leurs limites. En observant cette figure, certaines périodes des tests ressortent en étant supérieures aux limites. Ceci s'explique par une faute qui peut être attribuée à des problèmes de capteurs, ou un changement du comportement du système. Certains moments ont été sélectionnés afin de

retrouver les fautes dans les données brutes. Sur la Figure 80, les deux tests ont permis de détecter, le colmatage du capteur fin avril. Autrement, sur la Figure 81, la série de données de DCO soluble, le test Q a donné lieu à la détection de la dérive des données. Ceci est dû au manque de nettoyage du capteur et à la mauvaise procédure de nettoyage. Ces informations avaient été écrites dans le cahier de bord du suivi des capteurs. Enfin, la faute dans la série des DCO n'avait pas été détectée par la méthode univariée. Ceci conclut que la méthode multivariée peut être un complément à l'outil univarié.

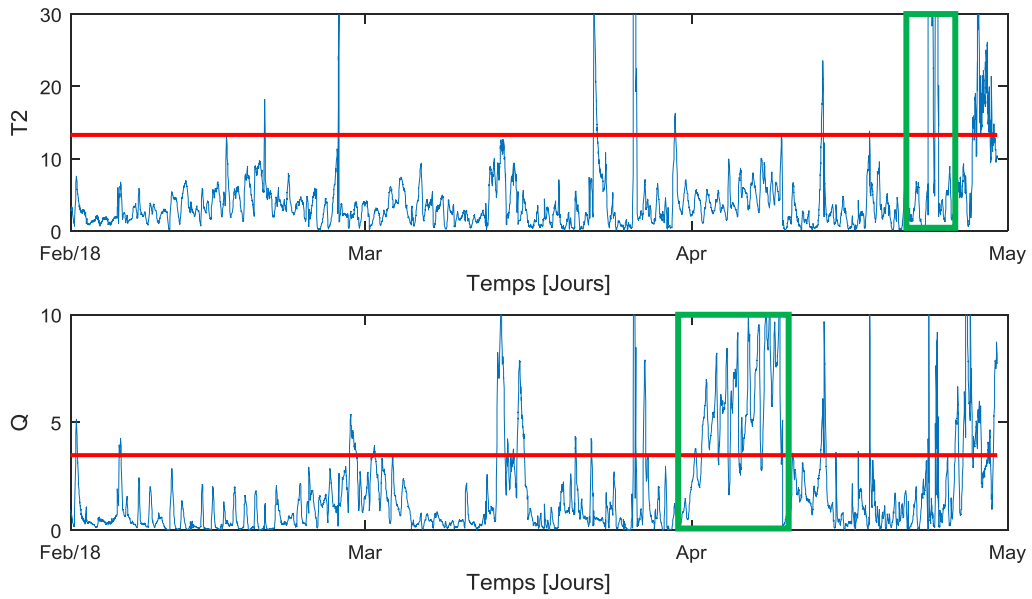


Figure 79. Test Q et T² pour la série de nouvelles données entre février et avril 2018 obtenues à l'affluent du décanteur primaire du pilEAUte

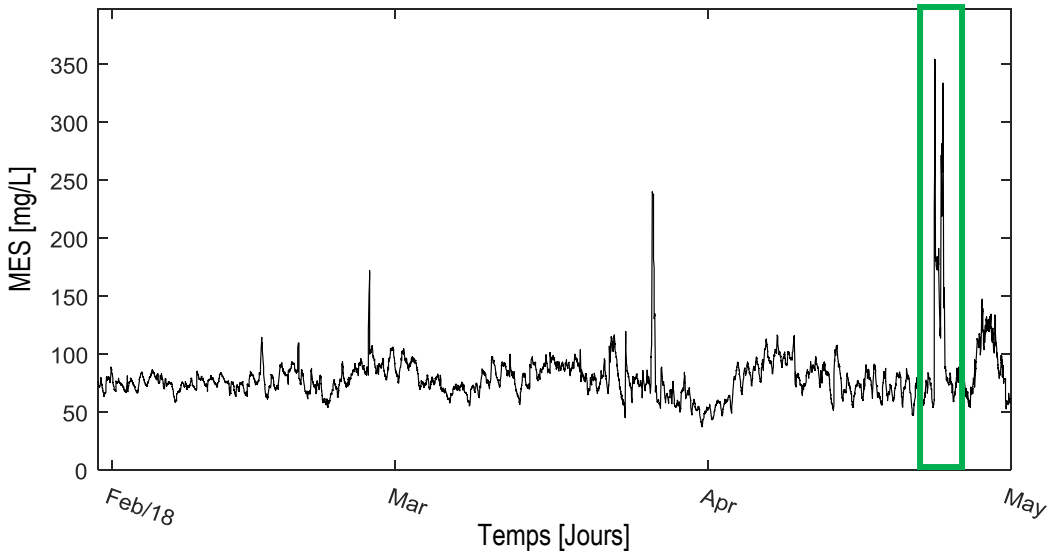


Figure 80. Données prétraitées par la méthode univariée de MES dans l'effluent du décanteur primaire du pilEAUte dont une faute a été détectée par la méthode ACP

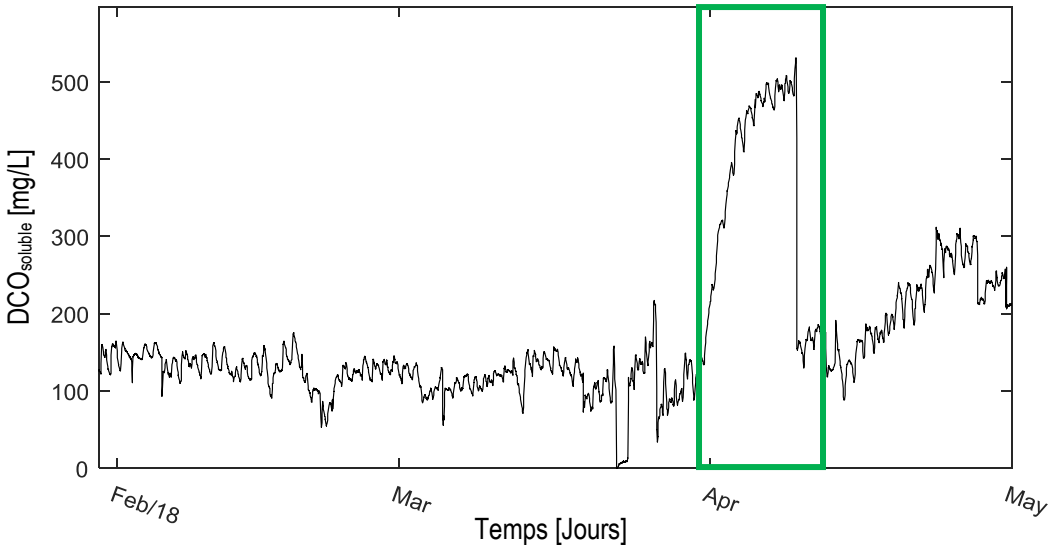


Figure 81. Données prétraitées par la méthode univariée de DCO soluble dans l'effluent du décanteur primaire du pilEAUte dont une faute de dérive a été détectée par la méthode ACP

b. Bioréacteurs

Dans cette deuxième illustration, une application de la méthode multivariée illustre les données provenant de capteurs mesurant les MES, l'oxygène dissous, et le débit d'air au sein d'un bioréacteur à boues activées pi-l*EAU*te et l'ammonium au niveau de l'effluent du décanteur primaire comme indicateur de la charge de la StaRRE. Avant le traitement de ces données, les données brutes ont aussi été filtrées par la méthode univariée en détectant les données aberrantes et en lissant le signal sur la période de janvier 2018 à mai 2018 (Alferes et al., 2013b) (Figure 82 et Figure 83). Comme pour le premier cas d'étude, une période de données a été choisie afin de mettre en place le modèle ACP. Cette dernière se positionne entre le 18 janvier 2018 et 21 janvier 2018 (Figure 84 et Figure 85). Ces données ont été normalisées et le graphique des pourcentages des valeurs propres (scree plot) a été construit automatiquement (Figure 86). D'après cette figure, le choix du nombre de composantes peut s'arrêter sur trois puisqu'elles représentent 95 % de la variation dans les données. Finalement, la dernière étape, dans la mise en place du modèle ACP, est le calcul automatiquement des deux tests Q et T^2 et leurs limites ainsi que leurs représentations graphiques montrées en Figure 87.

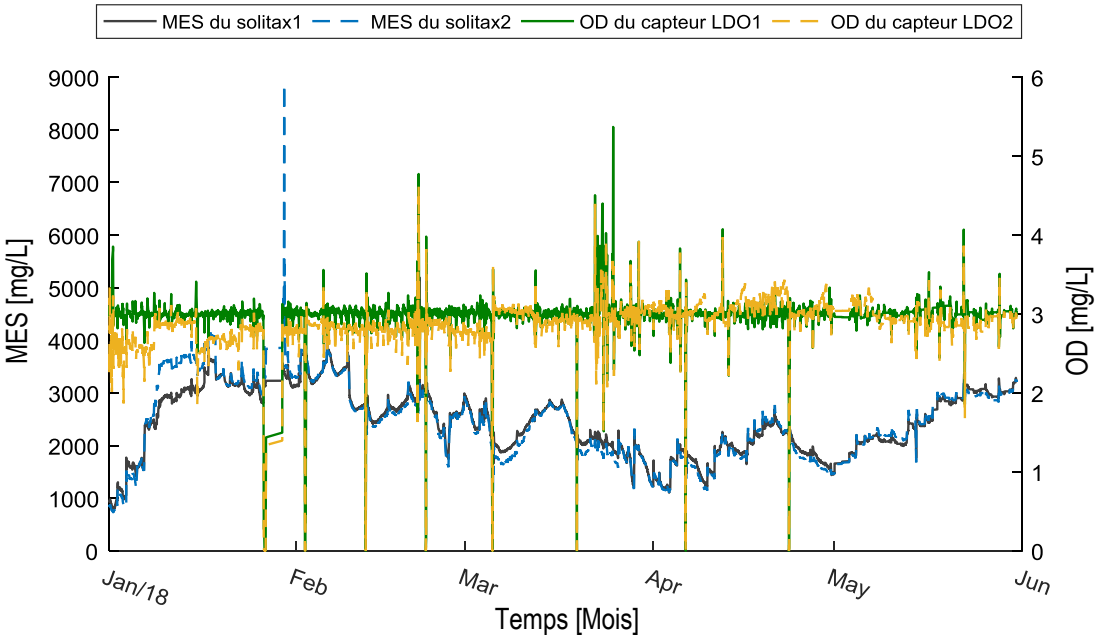


Figure 82. Données des MES et OD mesurées dans le bioréacteur pilEAUte

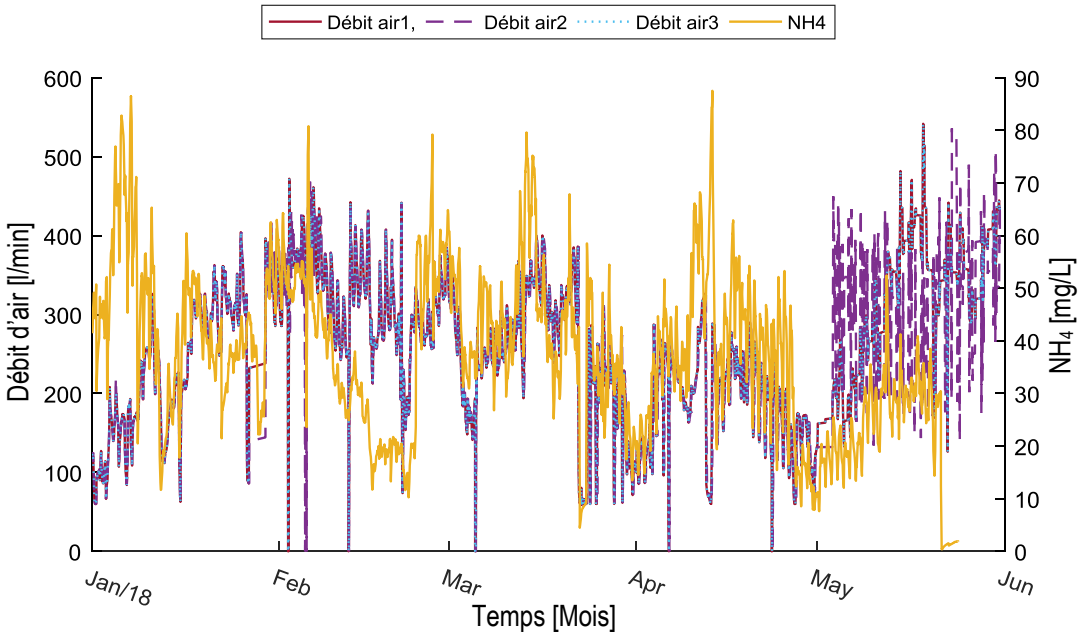


Figure 83. Données prétraitées par la méthode univariée des débits d'air et NH_4 mesurées dans le bioréacteur pilEAUte et à l'effluent du décanteur primaire

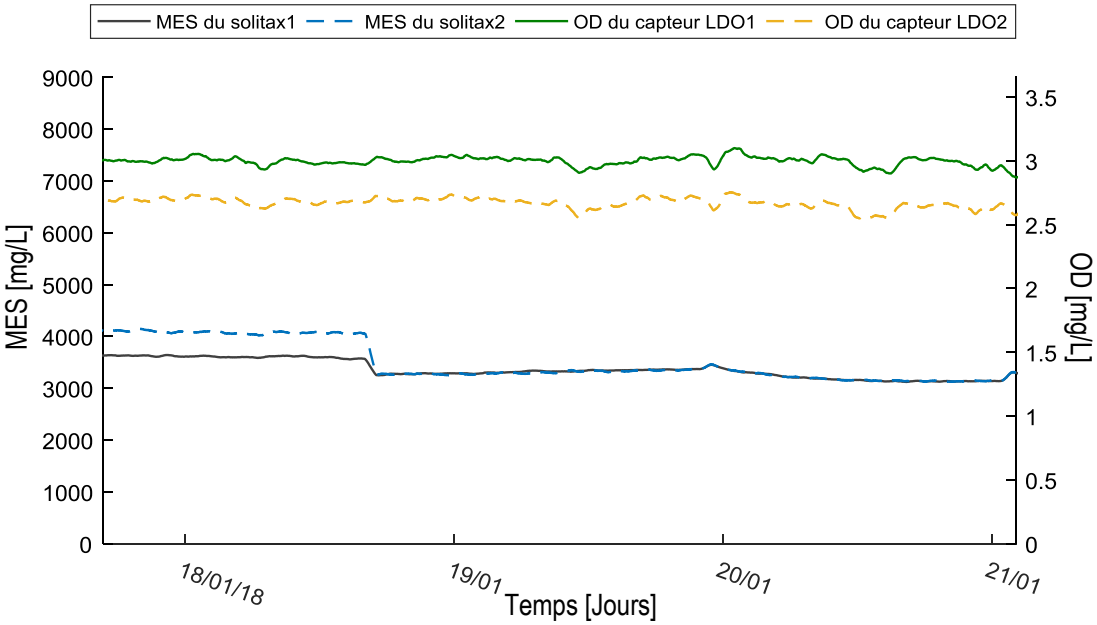


Figure 84. Données normales sélectionnées de MES et d'OD pour la construction du modèle ACP

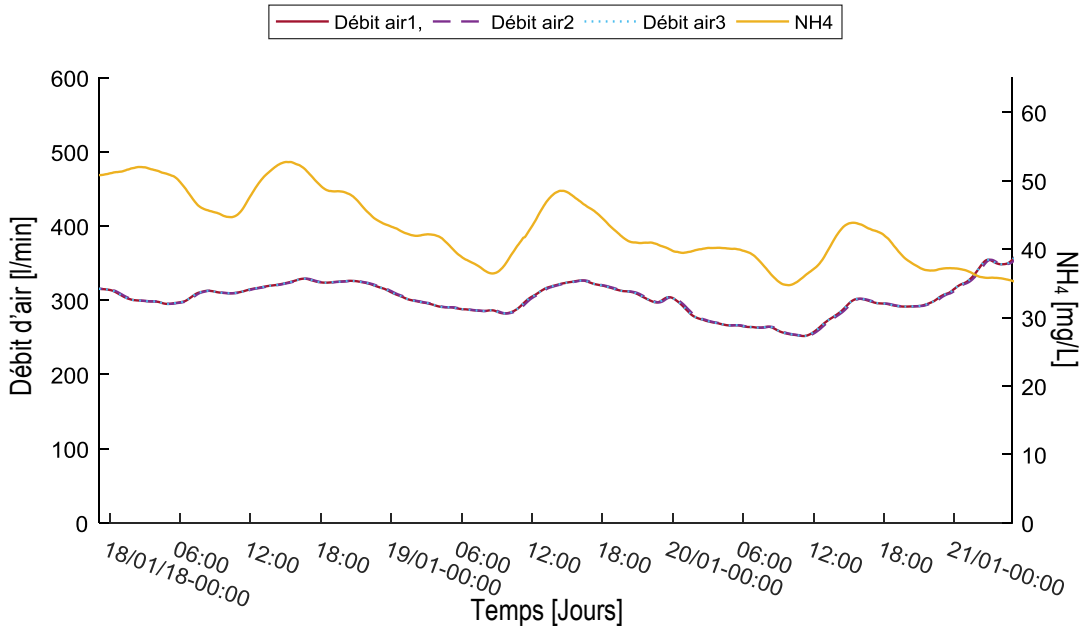


Figure 85. Données normales sélectionnées des débits d'air et de NH_4 pour la construction du modèle ACP

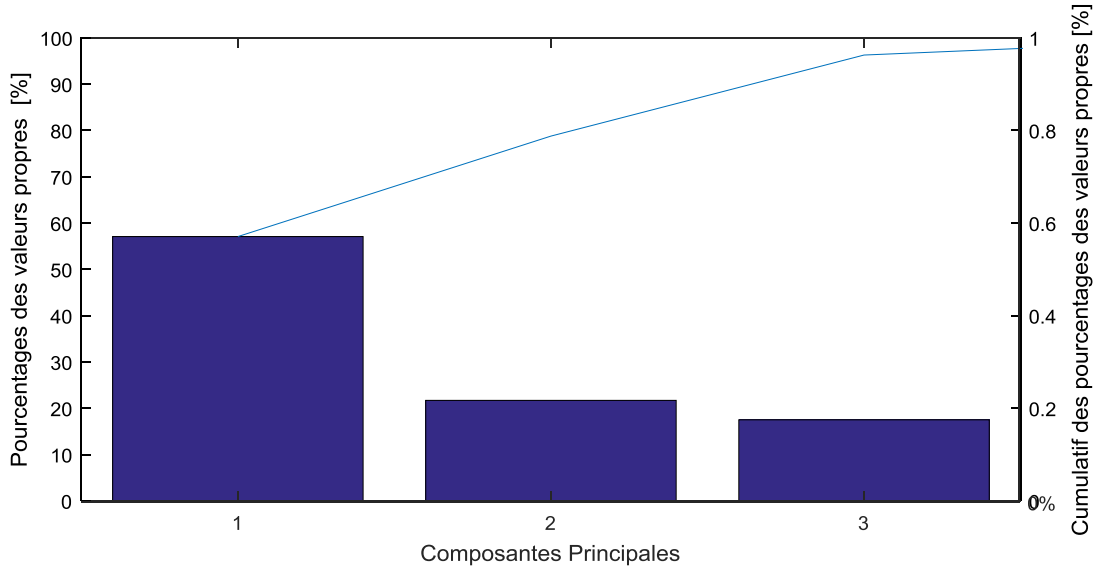


Figure 86. Pourcentages des valeurs propres pour les composantes principales du pilEAUte

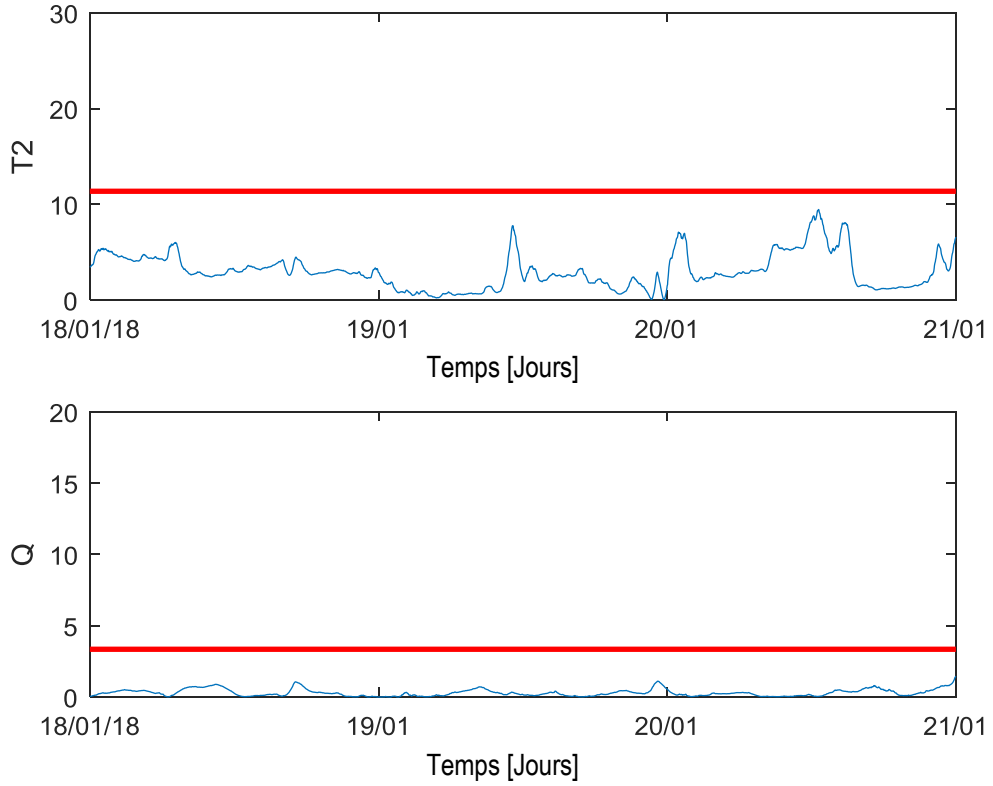


Figure 87. Test Q et T^2 pour les données normales du bioréacteur pilEAUte

Ensuite, ayant obtenu le modèle et les limites des tests, de nouvelles données peuvent être évaluées en les projetant sur le modèle ACP dans le but de détecter les fautes ou d'autres anomalies. La période choisie pour illustrer la méthode est située entre le 21 janvier au 10 février 2018 (Figure 89 et Figure 90). Pour celle-ci, les deux tests ont de nouveau été déterminés automatiquement et représentés graphiquement en Figure 88. D'une

part, il peut être constaté que sur certaines périodes, les deux tests sont au-dessus des limites correspondant à une faute dans les données d'une ou plusieurs variables. Plus généralement, le 5 février le test Q a permis de mettre en évidence le changement soudain de NH_4 pendant une courte période. Cette faute est possiblement due à un colmatage du capteur (Figure 90). En outre, le test T^2 a mis en valeur deux fautes dans la série de données de l'oxygène dissous le 26-31 janvier et le 3 février (Figure 98). La première faute retrouvée est une défaillance complète du capteur avec une diminution de l'oxygène à zéro et une stagnation à 1.5 mg/L et une reprise à la normale par la suite. La deuxième faute correspond à un arrêt de l'aération pendant quelques minutes d'après le cahier de bord du suivi des capteurs. Ces deux fautes avaient été aussi détectées et rejetées par la méthode univariée lors de l'étape de détection des fautes en Figure 54. Ainsi, ceci conclut que la méthode permet aussi de la détection des mêmes fautes que l'outil univarié.

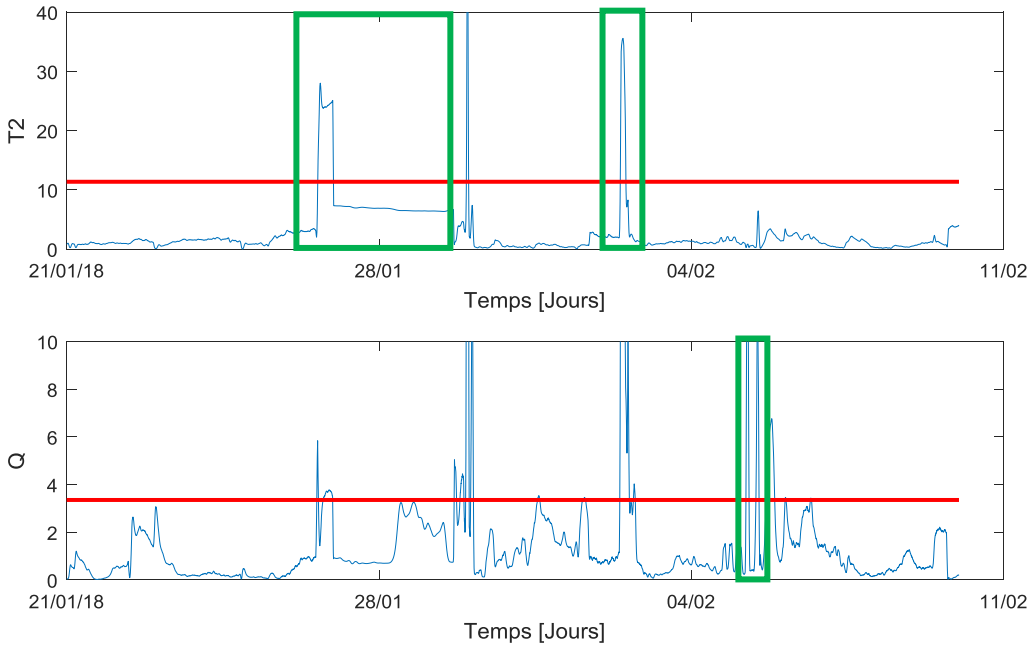


Figure 88. Test Q et T^2 pour les données à traiter bioréacteur pilEAUte

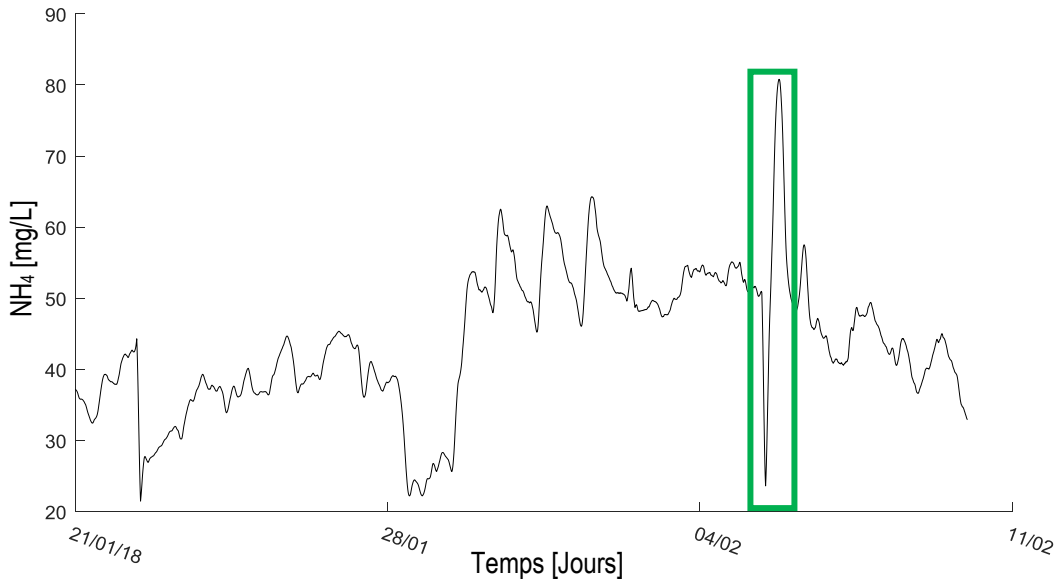


Figure 89. Données prétraitées par la méthode univariée du NH_4 à l'effluent du décanteur primaire dont une faute a été détectée par la méthode ACP

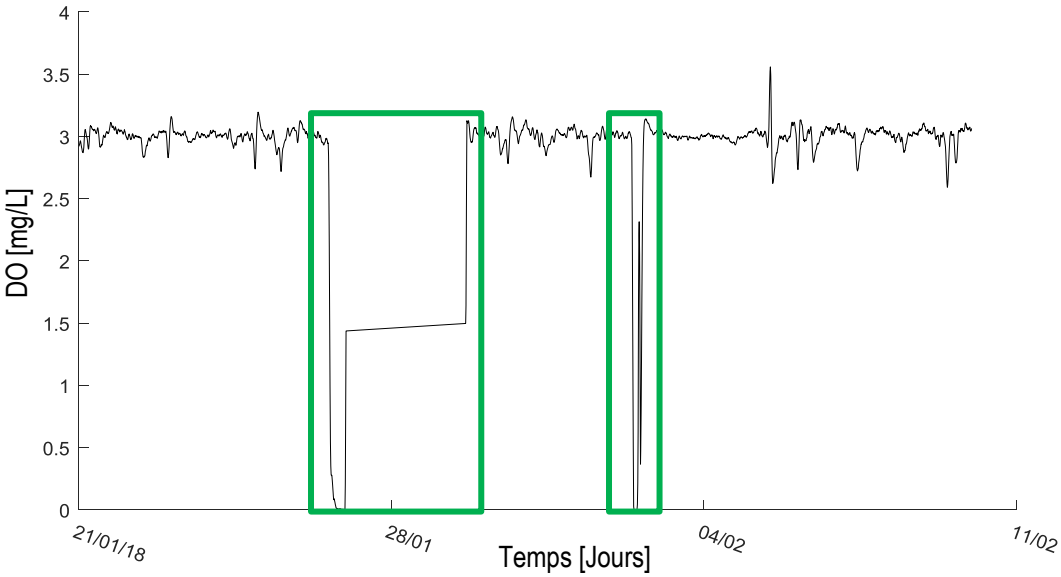


Figure 90. Données prétraitées par la méthode univariée d'oxygène dissous dans le bioréacteur pilEAUte dont des fautes ont été détectées par la méthode ACP

c. Méthode multivariée en ligne

Pour cette dernière illustration, la méthode multivariée sera illustrée dans le but de son installation pour de la détection de fautes en ligne. Premièrement, supposons qu'en entrée d'une StaRRE, une station de mesure a été installée avec plusieurs capteurs mesurant divers variables (Ammonium, potassium, nitrates). Une redondance est retrouvée en termes de variables mesurées, mais aussi en termes de capteurs car certains capteurs

ont été installés en redondance. Deuxièmement, si une pollution toxique arrive à la station, la personne responsable aimerait être avertie directement. Les capteurs devront réagir en même temps pour faire ressortir visuellement le problème de toxicité. Ainsi, la méthode ACP permet de montrer ce résultat au travers de l'utilisation des tests Q et T^2 et des données provenant de l'expérience sur le « Comportement des capteurs » (section 4.1.2). Les séries de données s'échelonnaient du 5 août au 19 août 2018 (Figure 91 et Figure 92).

Tout d'abord, le choix des données normales pour la construction du modèle ACP s'est arrêté du 6 août au 8 août 2018 (Figure 93 et Figure 94). Par ce choix, le nombre de composantes principales, pour la construction du modèle, peut être basé sur le scree plot de la Figure 95. D'après cette figure, le nombre de CP retenu sera de trois, correspondant à 94 % de la variation dans les données. Finalement, les deux tests Q et T^2 ainsi que leurs limites ont pu être évalués automatiquement et représentés graphiquement (Figure 96). Il peut être observé que les pics de fortes charges se retrouvent hors des limites. Ce point est à prendre en compte pour la détection des fautes. Lorsque de nouvelles données vont se projeter sur le modèle avec des pics de même intensité ou plus, ils seront automatiquement détectés comme une faute ou anomalie et une alarme pourra être envoyée à l'opérateur de l'usine.

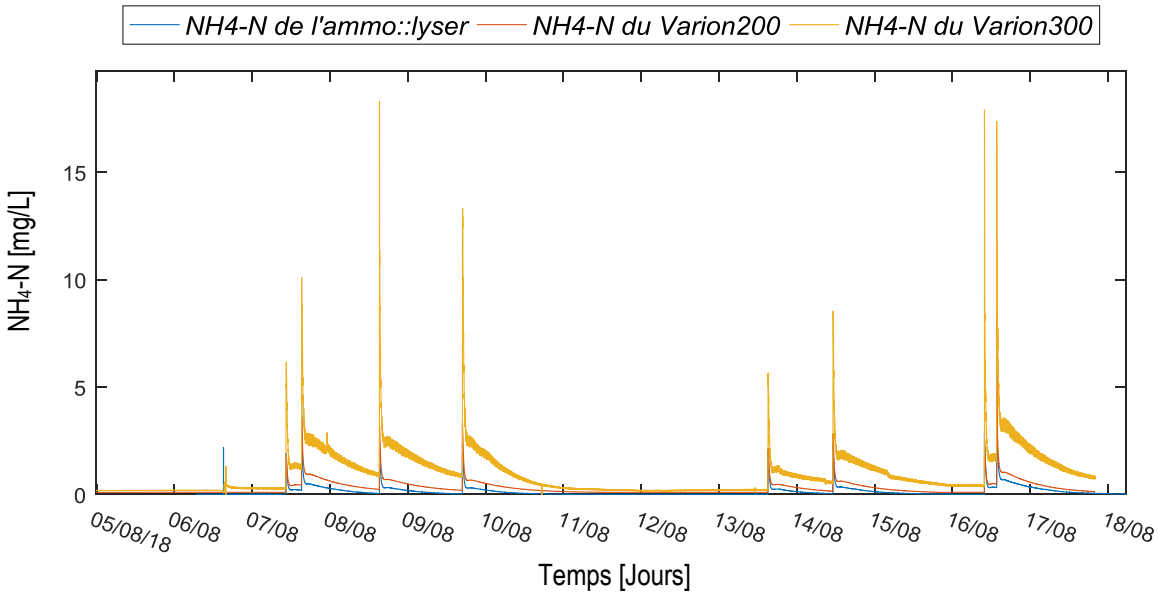


Figure 91. Données brutes de NH_4-N pour l'illustration de la méthode ACP mesurées dans un réacteur aéré du pilEAUte

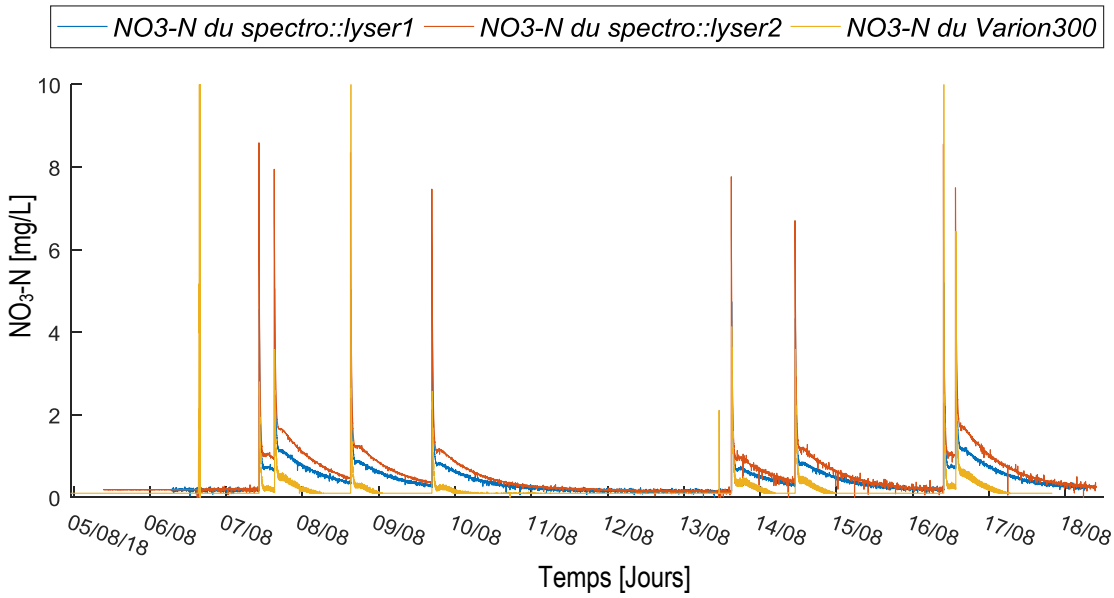


Figure 92. Données brutes de NO_3-N pour l'illustration de la méthode ACP mesurées dans un réacteur aéré du pilEAUte

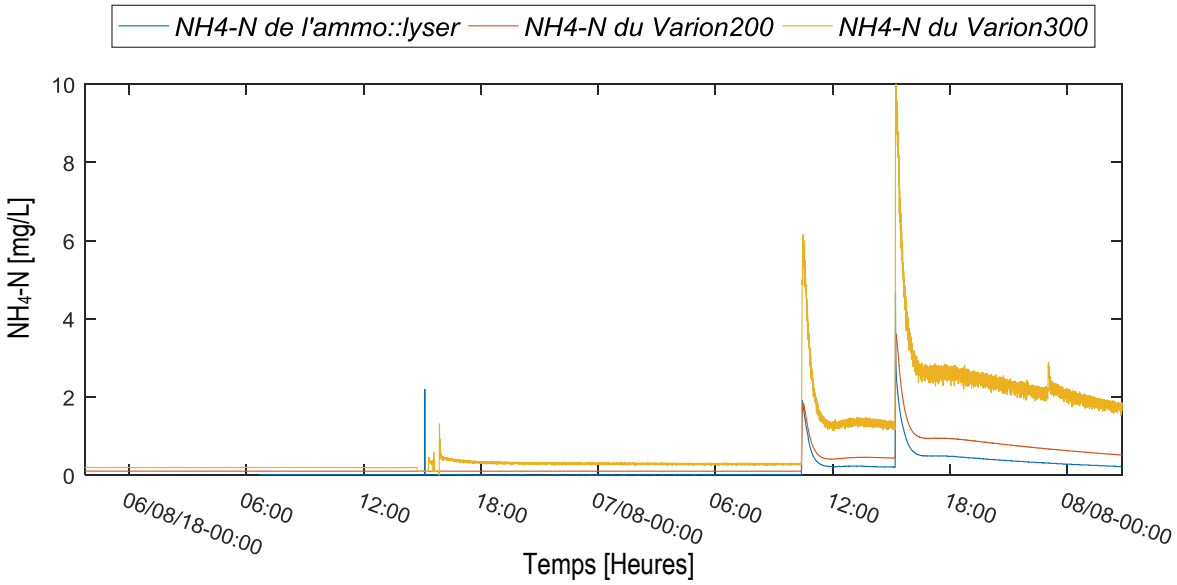


Figure 93. Données normales de NH_4-N pour le développement du modèle ACP mesurées dans un réacteur aéré du pilEAUte

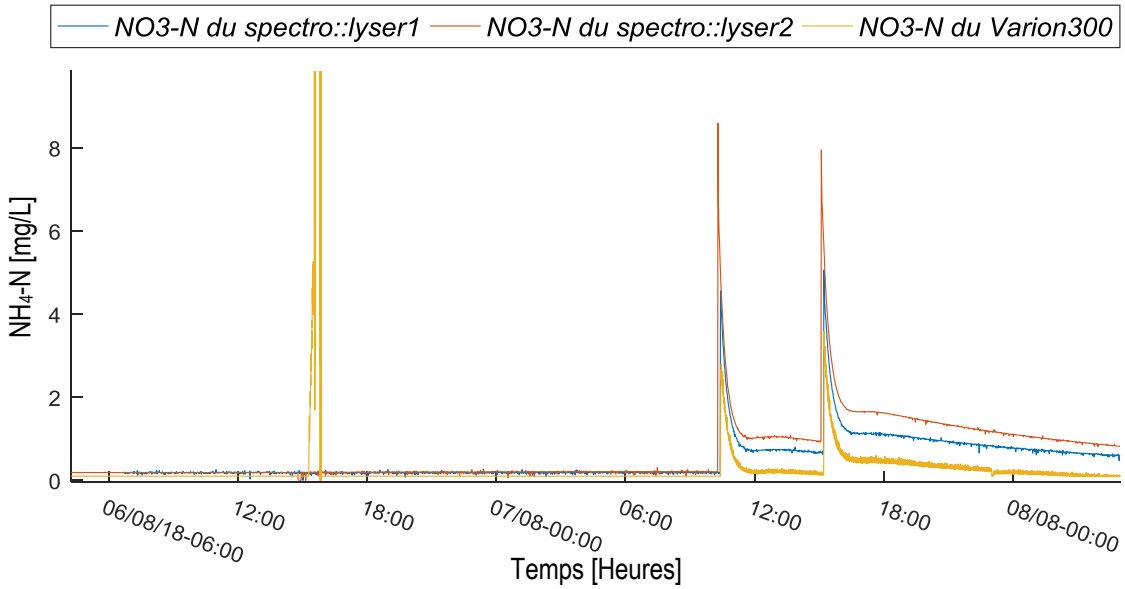


Figure 94. Données normales de NO_3-N pour le développement du modèle ACP mesurées dans un réacteur aéré du pilEAUte

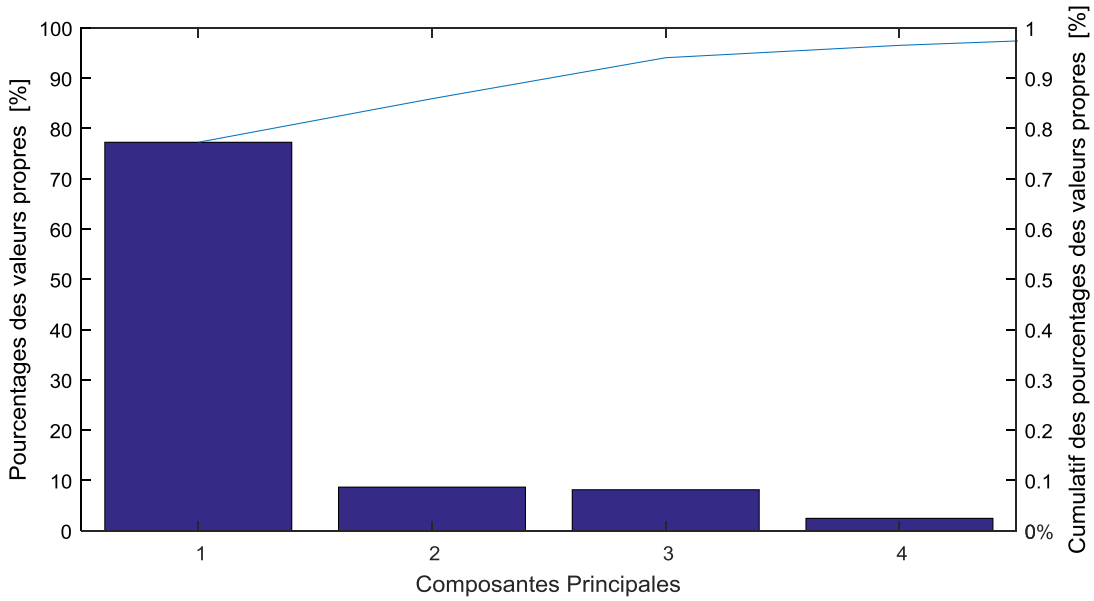


Figure 95. Pourcentages des valeurs propres en fonction des composantes principales réacteur aéré du pilEAUte

.

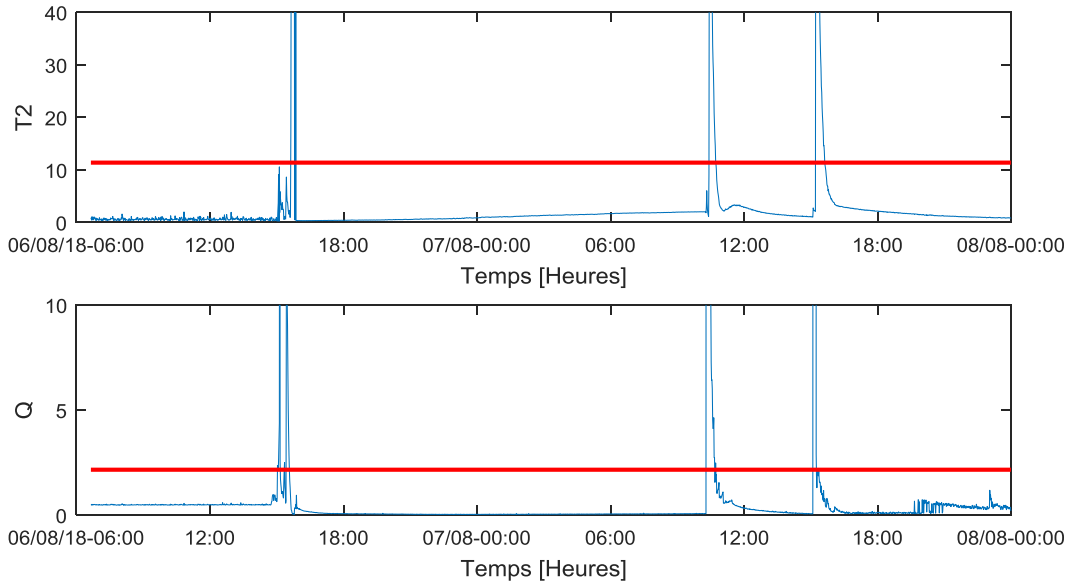


Figure 96. Tests Q et T^2 et leurs limites pour la série normale réacteur aéré du pilEAUte

Ayant mis en place le modèle ACP, de nouvelles données peuvent être projetées avec des pics de fortes intensités afin d'observer si la méthode détecte ces derniers. La période est comprise entre le 9 août et le 18 août 2018 (Figure 91 et Figure 92). Ainsi, pour ces données, les deux tests Q et T^2 et leurs limites ont été calculés automatiquement et construits graphiquement en Figure 97. Il en ressort que les pics de fortes intensités ont été détectés par la méthode multivariée étant donné qu'ils sont au-dessus des deux limites. Ces pics sont en coordination avec les pics observables dans les données brutes tels qu'illustrés par la Figure 98 et Figure 99 pour les deux variables NH_4 et NO_3. Ceci conclut qu'hypothétiquement, la méthode multivariée peut être installée en ligne afin de détecter des pollutions toxiques arrivant dans des StaRRE.

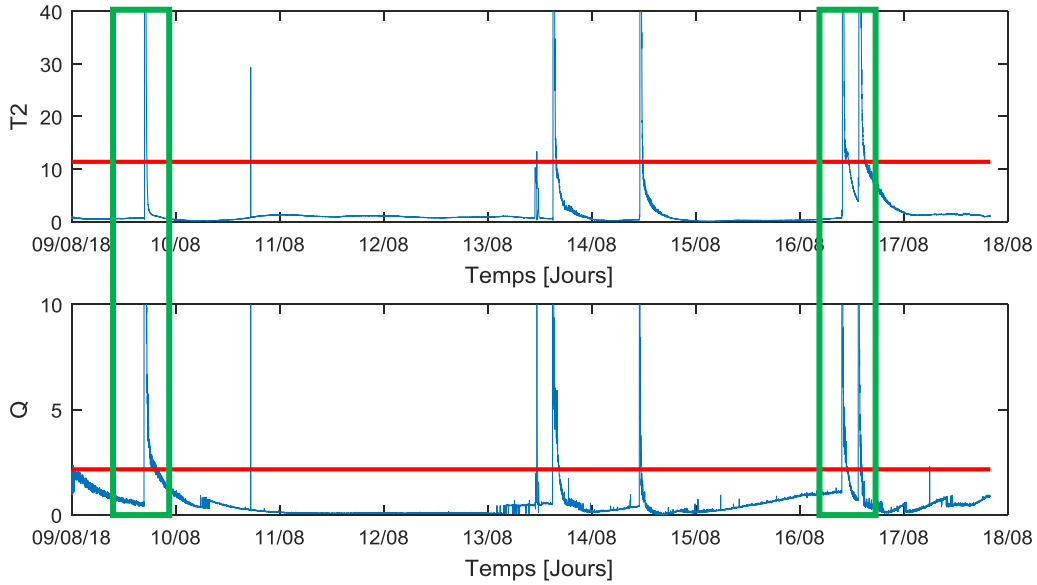


Figure 97. Tests Q et T² pour la série de nouvelles données réacteur aéré du pilEAUte

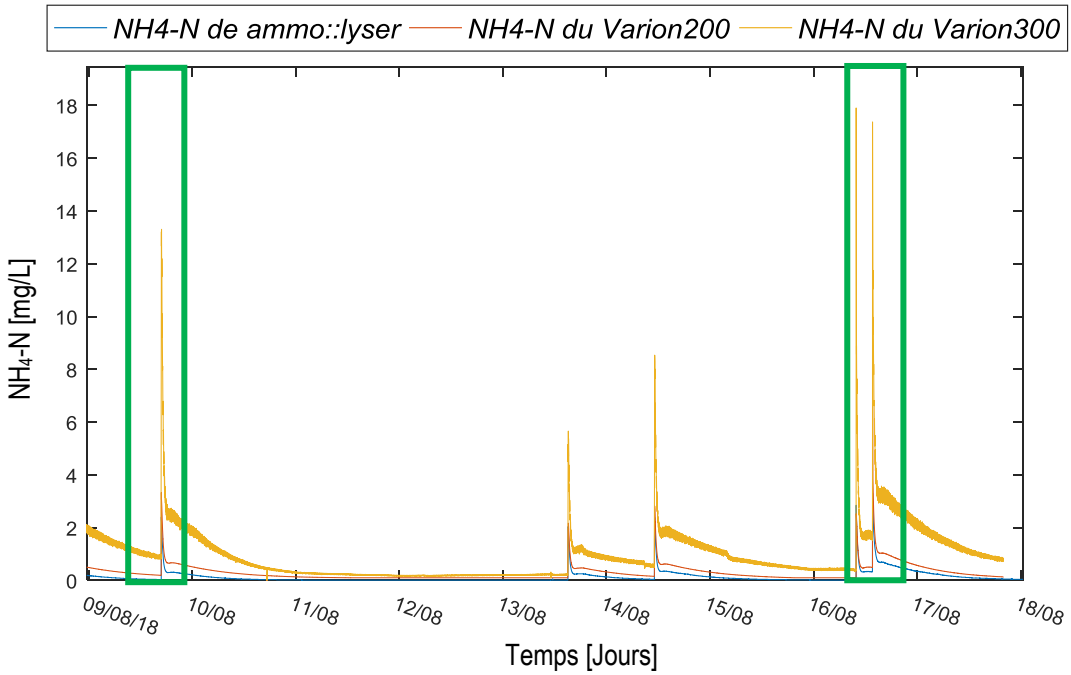


Figure 98. Données brutes de NH_4 mesurées dans un réacteur aéré du pilEAUte pour trois capteurs

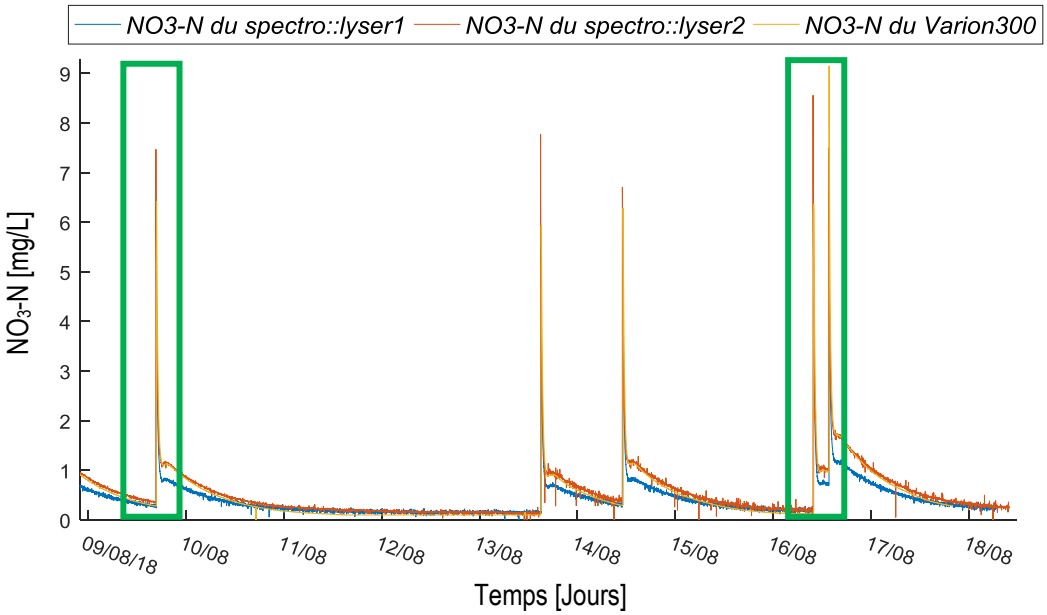


Figure 99. Données brutes de NO_3 mesurées dans un réacteur aéré du pilEAUte pour trois capteurs

4.2.2.3. Conclusion

La méthode multivariée a montré sa performance dans la détection des fautes dans trois cas d'étude. Ce point met en évidence la simplicité et la modularité d'application de cet outil pour divers cas. Ces illustrations permettront d'aider les futurs utilisateurs. Dans le dernier cas d'étude, la méthode a permis de mettre en évidence son utilisation afin de détecter des pollutions toxiques en entrée d'une station pouvant impacter le traitement en aval. Mais, pour ce faire, la méthode devra être implantée en ligne et la programmation devra être effectuée afin qu'elle fonctionne en autonomie.

4.3. Validation des capteurs : Redondance des capteurs

Au sein de cette partie, des illustrations de la méthode de redondance des capteurs seront exposées. Les trois sites d'étude pil*EAU*te, kam*EAU* et bord*EAU*x ont permis d'illustrer la méthode.

4.3.1. Méthodes

La Figure 100 montre le diagramme de la validation des données par la redondance de capteurs avec diverses étapes :

- Placement de mêmes capteurs à une même localisation
- Diagramme de contrôle
- Avertissement/ Action corrective
- Données validées

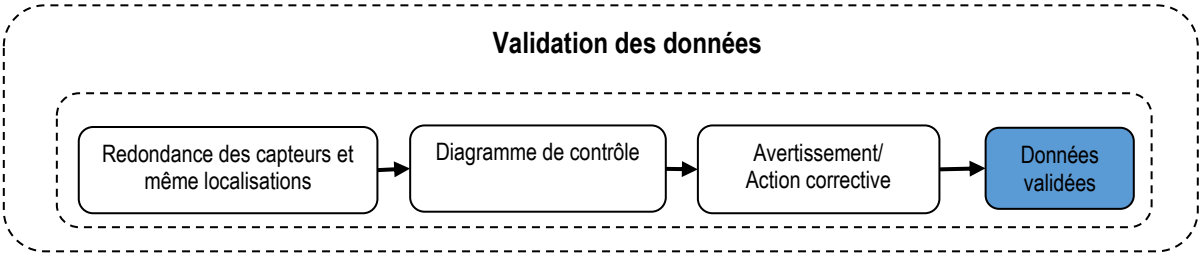


Figure 100. Diagramme de la validation des données par la redondance de capteurs

4.3.2. Applications

4.3.2.1. pilEAUte

Dans le chapitre 3 (section 3.1.1.1), l'ensemble des capteurs installés au sein de la station a été exposé. Grâce à cette grande variété, une redondance a pu être mise en place pour certaines variables illustrées dans le Tableau 37. Graphiquement, la Figure 101 et la Figure 102 montrent la redondance des MES et de l'OD au sein du bioréacteur. Pour ces variables, les mêmes capteurs (marque, type, méthode de mesure) étaient installés ensemble. Il ressort que les séries de données des mêmes variables sont légèrement différentes. Il est noté qu'en janvier 2018, le biais est relativement élevé pour la variable MES. Ceci se traduit par des problèmes de calibration des capteurs d'après le cahier de bord du suivi des capteurs. Cependant, des valeurs aberrantes soudaines (élevées ou faibles) de MES sont observées, qui sont la conséquence des nettoyages hebdomadaires.

Tableau 37. Redondance de variables mesurées sur deux points de mesures dans le pilEAUte

Points de mesures	***Sortie du décanteur primaire***	***Bioréacteurs***	***Effluent***
Variables redondantes	- NH_4 (x2)	- MES (x2) - OD (x2)	- NH_4 (x2) - NO_3 (x2)

Concernant la variable OD, des différences importantes sont notées en début d'année, entre janvier et mars 2018, où la calibration de ces capteurs n'était pas convenable. La calibration a été corrigée par la suite. Enfin, comme pour les MES, les valeurs aberrantes soudaines correspondent aux nettoyages hebdomadaires de ces capteurs comme ayant été détectées. Pour la variable OD, ces valeurs ont été détectées et rejetées lors du traitement de la série par la méthode univariée (Figure 55) (section 3.2.1).

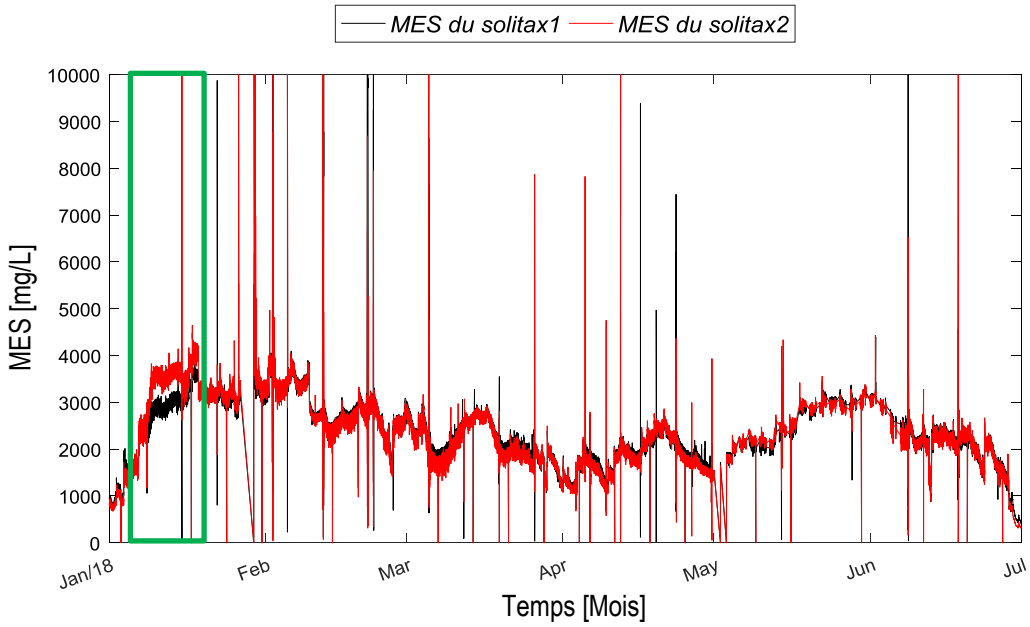


Figure 101. Redondance de la variable MES mesurée dans un bioréacteur du pilEAUte

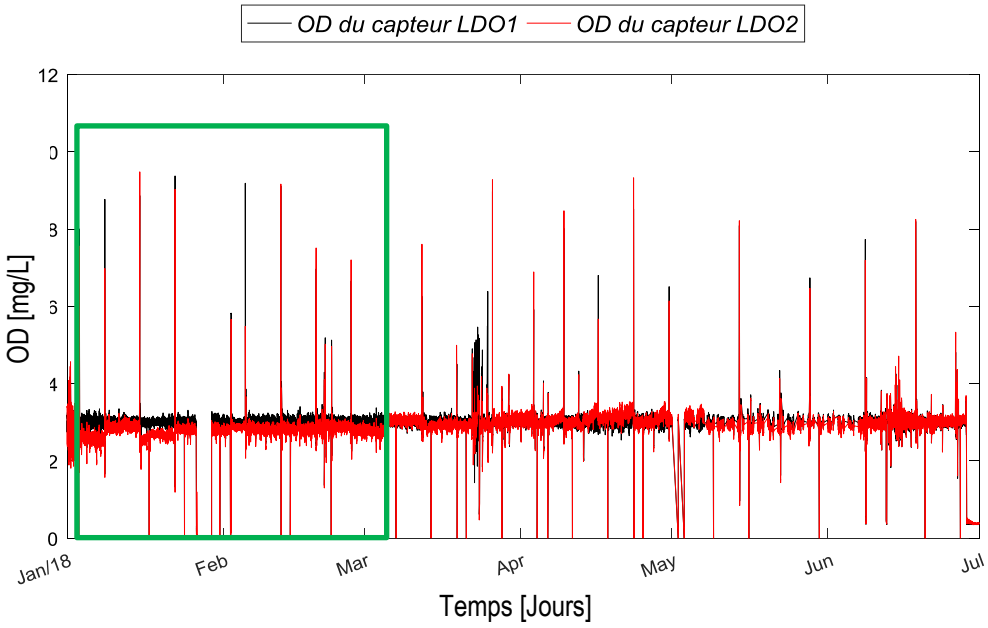


Figure 102. Redondance de la variable OD mesurée dans un bioréacteur du pilEAUte

4.3.2.2. kamEAU

Dans le chapitre 3, (section 3.1.3.1), l'ensemble des capteurs ayant été installé au niveau du site d'étude a été exposé. Une redondance de certaines variables mesurées à un même point était possible, comme indiqué dans le Tableau 38 pour les deux points de mesure à l'entrée et à la sortie du procédé.

La Figure 103, la Figure 104 et la Figure 105 montrent la redondance pour les variables au niveau de l'affluent. L'analyse de cette redondance n'a été effectuée qu'au niveau de ce point dû à une similitude des variables mesurées entre l'affluent et l'effluent (Tableau 38). Cependant, la redondance des variables n'est pas complète pour la température et le pH sur l'ensemble de l'année. Ce manque est dû à un changement de localisation et à des problèmes de mesures et de calibration des capteurs. Le doctorant, Bernard Patry avait besoin de mesurer des valeurs de pH dans le bioréacteur kam*EAU*. Pour la variable température, les périodes de redondance montrent des séries de données qui suivent les mêmes variations. En revanche, pour la variable pH, la redondance sur la période de novembre 2017 à février 2018 illustre des biais importants sur la majorité de la période étudiée. Cette différence peut se traduire entre les deux capteurs de marque différente. Enfin, concernant la troisième variable, les MES, les séries de données montrent des variations assez similaires avec quelques augmentations plus prononcées pour le solitax dues éventuellement à un colmatage plus facile pour ce capteur (Figure 105). Aussi, ces deux capteurs n'utilisent pas la même méthode de mesure. Le spectro::lyser mesure à l'aide de l'absorbance et le solitax par la méthode de diffusion de la lumière.

Tableau 38. Redondance de variables mesurées sur deux points de mesures dans le KAMAK

Points de mesures	***Affluent***	***Effluent***
Variables redondantes	- Température (x3) - pH (x2) - MES (x2)	- Température (x4) - pH (x2) - MES (x2)

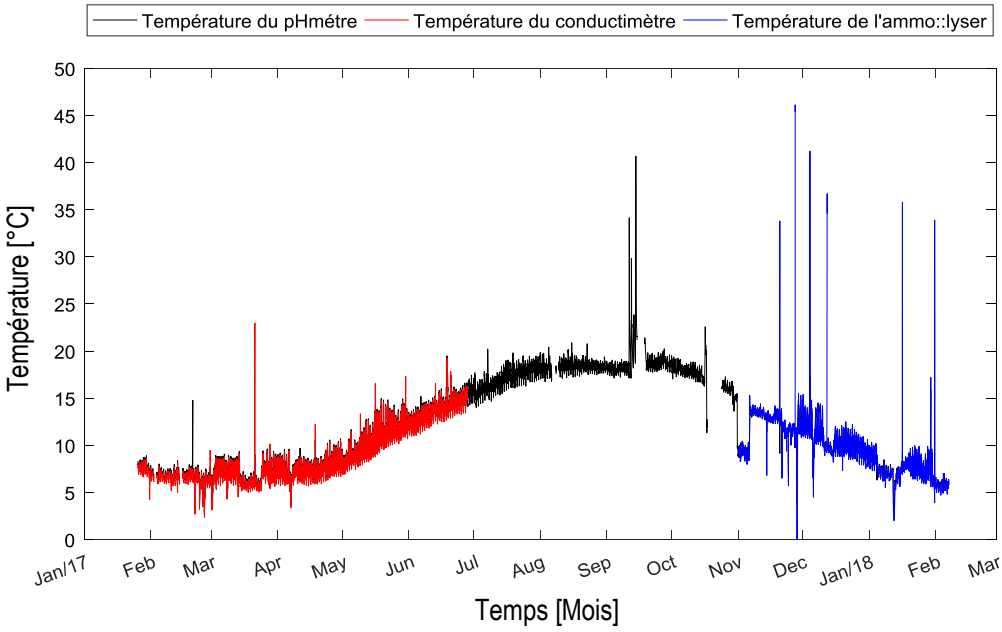


Figure 103. Redondance de la mesure de la température à l'entrée du KAMAK

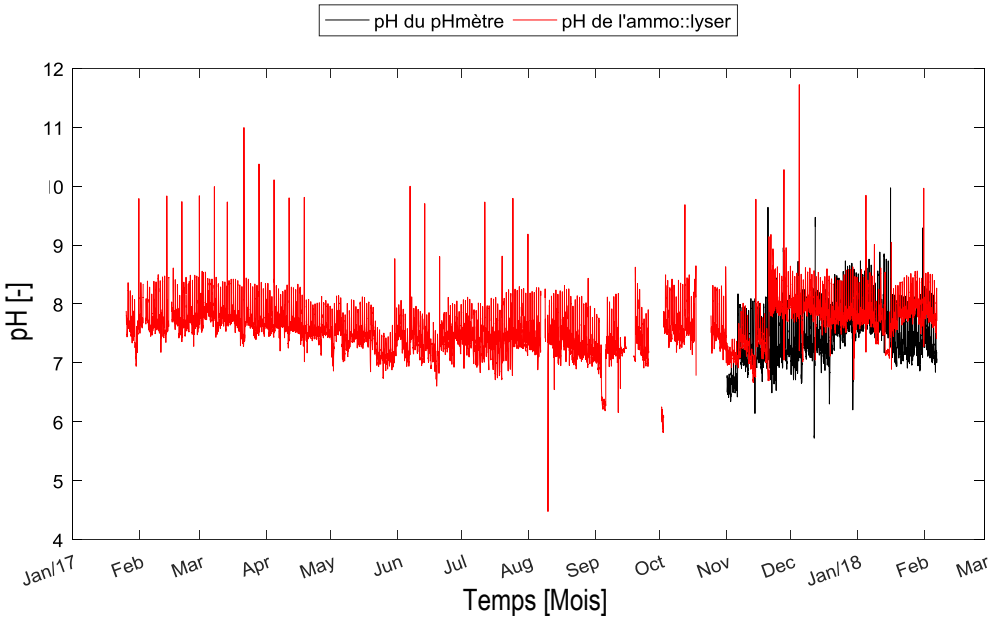


Figure 104. Redondance de la mesure du pH à l'entrée du KAMAK

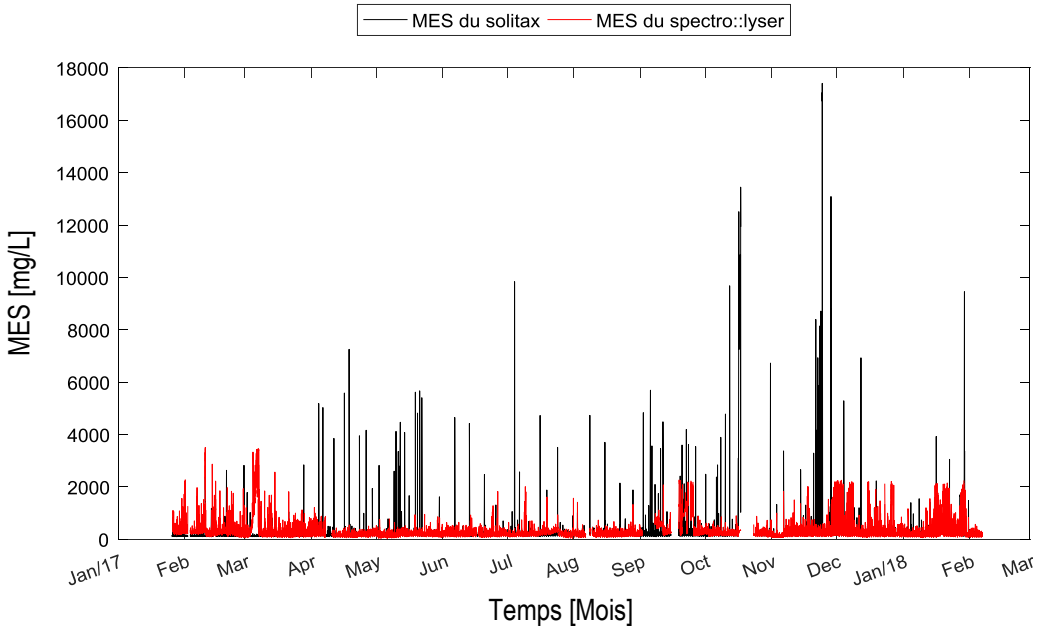


Figure 105. Redondance de la mesure des MES à l'entrée du KAMAK

4.3.2.3. bordEAUx

Dans le projet bord*EAU*x, une redondance au niveau des capteurs est retrouvée chaque fois pour les deux sites de mesure, le réseau d'égout et l'entrée de la StaRRE où deux capteurs de turbidité étaient installés. La Figure 106 montre cette redondance de la variable MES durant les quatre mois de campagne avec quelques différences entre les deux capteurs. Sur la Figure 123, il est observé à certains moments un biais important entre les deux séries de données comme par exemple en juillet où pour l'un des deux capteurs, le spectro::lyser, les données commençaient à dériver. Toutes ces différences peuvent être dues à un colmatage des capteurs à différents moments. Aussi, la différence peut être causée par une erreur dans la mise en place du calcul de la turbidité en MES. La formule a été mise en place avec des données de laboratoire qui apportent une source d'erreur en plus.

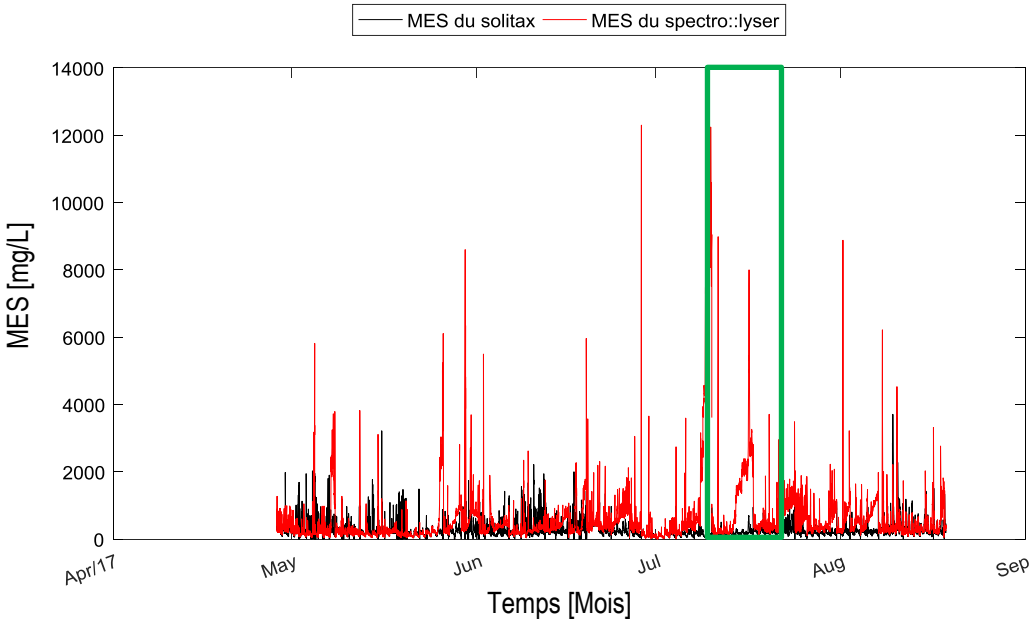


Figure 106. Redondance de la variable MES à l'entrée de la StaRRE à Bordeaux

4.3.3. Compléments

Afin de compléter cette méthode, la première méthode de validation des données, « diagramme de contrôle », exposée dans le chapitre 1, peut être utilisée (section 1.5.1). Dans cette méthode, les données en ligne sont comparées à des données de laboratoire provenant d'échantillons prélevées. La Figure 107 montre les données de MES de deux capteurs solitax exposées précédemment (section 4.3). Premièrement, sur la Figure 107 sans faire de zoom, les données de laboratoire suivent les données en ligne durant les six mois. En zoomant sur une période illustrée en Figure 108, il en ressort que les données de laboratoire sont proches des données en ligne et permettent ainsi de valider les données en ligne des deux capteurs.

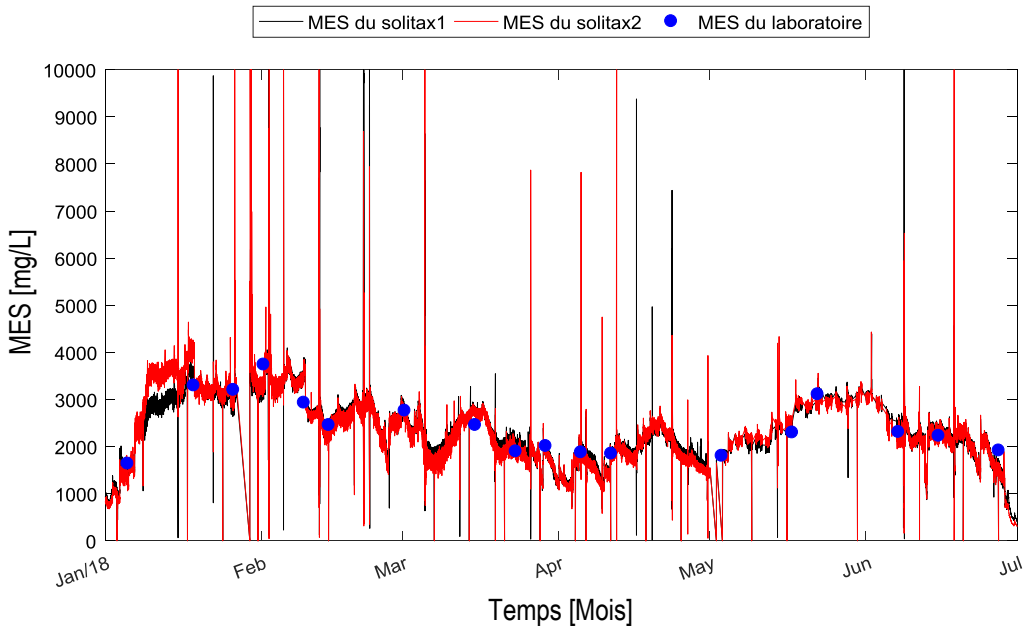


Figure 107. Validation des données en ligne de MES dans un réacteur à boues activées du pilEAUte avec des données de laboratoire

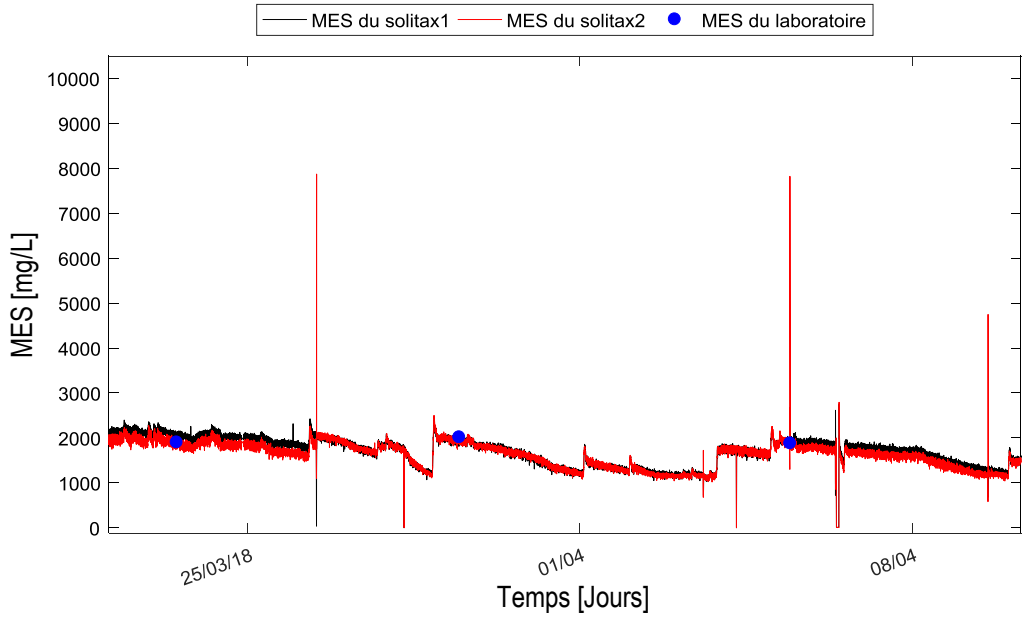


Figure 108. Agrandissement de la Figure 107 sur la validation des données en ligne de MES dans un réacteur à boues activées du pilEAUte avec les données de laboratoire (détail de la Figure 107)

4.3.4. Conclusion

La méthode de validation de données basée sur la redondance des capteurs et des variables mesurées a montré de bons résultats. Mais, celle-ci montre ses limites avec des biais importants sur certaines périodes. Pour diminuer ces limites, plusieurs suggestions pourront être émises : par exemple, l'ajout d'un troisième capteur ou la réalisation de mesures en laboratoire en simultanées afin de valider les données en ligne. Cependant, un coût supplémentaire doit être envisagé, mais certaines usines ne peuvent le couvrir.

En conclusion de ce chapitre, les trois sections ci-dessus permettent de montrer la qualité des données sur différents points et le lien entre chacun. La validation des données est une suite au traitement de ces données afin de connaître la fiabilité des données traitées. Aussi, cette validation peut être utilisée en même temps lors des nettoyages des capteurs ou d'expériences sur la réaction des capteurs afin de vérifier leur précision.

Conclusion et perspectives

Dans ce mémoire de maîtrise axé sur la qualité des données, plusieurs activités ont été effectuées et discutées telles que :

- Mettre en place de la maintenance de capteurs comprenant leur nettoyage et leur validation/calibration,
- Rendre simples et modulaires des outils de traitement des données dans leur application,
- Démontrer l'utilité des méthodes de redondance pour la validation des données.

Chaque activité sera discutée en exposant les conclusions et des perspectives pouvant être proposées.

Maintenance des capteurs

La maintenance des capteurs a été exposée dans la partie intitulée « réactions des capteurs » (section 4.1). Au sein de cette partie, plusieurs points ont été abordés. Tout d'abord, le nettoyage des capteurs peut avoir un effet sur les données en observant un biais entre avant et après le nettoyage. Ce biais est observable si la procédure de nettoyage n'est pas adéquate. La première suggestion afin de minimiser cette faute serait de mettre en place un nettoyage pro-actif afin de réaliser cette maintenance dans le temps imparti lorsque le capteur doit être nettoyé. La deuxième suggestion pour réduire l'impact du nettoyage serait d'utiliser des outils en ligne afin d'être averti directement si le capteur commence à dériver, à se colmater ou à subir une anomalie. Ces outils peuvent être les méthodes étudiées dans ce mémoire de maîtrise, soit la méthode univariée ou multivariée. Plus précisément, les nettoyages de capteurs sont décelés lors de la détection des données aberrantes. D'autres méthodes pourraient aussi être utilisées telles que la méthode des réseaux de neurones ou une moyenne mobile exponentielle pondérée. Il permet aussi de détecter des données aberrantes correspondantes à des nettoyages de capteurs (Hawkins et al., 2002). Ainsi, ces outils pourront tout à fait être employés pour le nettoyage pro-actif en indiquant quand le capteur devrait être nettoyé.

Ensuite, dans cette même partie, un deuxième point a été discuté tel que la compréhension de la réaction de capteurs redondants à une forte charge. Il en est ressorti qu'il existait parfois un biais entre les données de deux capteurs similaires. Ce dernier se traduit soit par un problème de calibration ou par un capteur défectueux. Dans le premier cas, une recalibration aurait dû être effectuée. Dans le deuxième cas, le capteur aurait dû être remplacé. Mais, un but secondaire de la campagne était l'évaluation de la réaction des capteurs à une forte charge d'un réactif. En contrepartie du biais des capteurs, ces derniers réagissent en même temps lors de l'ajout d'une forte charge d'un constituant. Ainsi, les suggestions pour la minimisation du biais, seraient d'effectuer des mesures de laboratoires ponctuelles ou l'installation d'un troisième et même capteur afin de vérifier lequel des capteurs doit être remplacé ou recalibré.

Traitement des données

Deux outils de traitement des données ont été étudiés dans ce mémoire. Un premier outil dont le traitement était univarié (une variable à la fois) a pour objectifs la détection des données aberrantes, le lissage des données afin de minimiser le bruit et la détection des fautes. Le second outil dont le traitement était multivarié (plusieurs variables à la fois), avait pour but de détecter les fautes similaires ou non détectables par la méthode univariée. Ces deux outils avaient été développés par le passé (Alferes et al., 2012) mais ont été retravaillés et rendus plus simples dans leur utilisation. Dans le même ordre d'idée, la rédaction de SOP (SOP méthode univariée et SOP méthode multivariée) pour chaque outil et une quantification des divers paramètres, pour les six cas d'étude, ont été exposés dans ce mémoire afin de faciliter leur utilisation. Elles serviront à des futurs utilisateurs ayant quelques connaissances en programmation et du système d'où proviennent les données. Les méthodes ont aussi été rendues modulaires avec des blocs de fonctions pour chaque étape des outils. Comme expliqué précédemment, les méthodes ont été appliquées à différents jeux de données provenant des projets kam*EAU*, bord*EAU*x et pil*EAU*te en nécessitant un minimum de calibration par l'utilisateur afin de montrer leur simplicité, leur facilité et leur robustesse dans leur application. Les données traitées pour les projets bord*EAU*x et kam*EAU* seront utilisées dans le but de calibrer et valider deux modèles développés par les étudiants au doctorat, respectivement Julia Ledergerber et Bernard Patry. Cependant, afin de tester si ces méthodes sont généralement applicables, il serait intéressant de les utiliser dans d'autres domaines, par exemple les industries pharmaceutique ou agroalimentaire, etc.

Sur le diagramme général de la séquence de traitement de données (Figure 36), des utilisations additionnelles des outils de traitement de données développés dans ce mémoire, ont été positionnées avec des flèches rouges. La perspective serait d'employer ces derniers en temps réel afin de détecter les fautes des capteurs en ligne. Elle permettra aussi de réaliser une maintenance pro-active tout en connaissant le nettoyage, la calibration ou le changement du capteur. Les données traitées et validées devraient s'enregistrer automatiquement dans la base de données dat*EAU*base afin d'avoir une gestion optimale de l'ensemble des données sur une seule plateforme.

De plus, en rentrant plus en détails dans les deux méthodes de traitement de données telles que l'univariée et la multivariée, ces outils ont permis la détection de données aberrantes, d'anomalies, de pannes et de fautes au sein de multiples séries de données. Cependant, deux nuances sont mises en évidence pour la méthode univariée : le choix manuel de paramètres et non automatique et la détection incomplète des données aberrantes et des fautes.

Pour ces deux points, la méthode devra être retravaillée. Un ajout de nouveaux indicateurs de défaillance devra être effectué afin d'avoir un traitement complet. Des paramètres similaires pour chaque série de données devront être mis en place dans le but d'obtenir une méthode la plus autonome possible.

Enfin, dans un diagnostic complet des fautes, ces dernières doivent être isolées (la cause première de la faute) et identifiées (le type ou la nature de la faute) après avoir été détectées. Ainsi, ces deux dernières méthodes pourront être implantées afin d'obtenir des outils les plus optimaux dans le diagnostic des fautes.

Validation des données par redondance

La dernière partie du mémoire traitant la validation des données par la méthode de redondance des capteurs et des variables s'est montrée être satisfaisante. Cette méthode a permis d'identifier des biais et des dérives de capteurs dus à de mauvaises calibrations ou des capteurs défectueux. Mais, elle peut être couplée à la méthode de validation du diagramme de contrôle pour augmenter la fiabilité de la validation des données. Une autre suggestion peut être proposée qui est d'ajouter un troisième capteur redondant afin d'augmenter la fiabilité du résultat et du suivi des données. Cependant, les deux suggestions amènent l'idée d'un coût supplémentaire que certaines petites usines de traitement ne peuvent fournir.

Finalement, l'ensemble des activités a montré certains points déjà observés par d'autres auteurs : la qualité des données est un domaine assez complexe dû entre autres à des méthodes de mesures diverses pour les capteurs. L'utilisation de ces capteurs dans le domaine des eaux usées est aussi compliquée car les systèmes d'eau (réseaux d'égouts, rivières, StaRRE) sont assez agressifs et impactent la durabilité et la fiabilité du capteur. La suggestion de mettre en place les outils de traitement de données en ligne permettrait d'atteindre les défis d'une bonne qualité des données tout en surveillant et détectant les anomalies ou fautes en temps réel. Avant l'implantation de ces outils en temps réel, les outils univarié et multivarié offrent plutôt des données de très grande qualité en permettant de les valider en temps (presque) réel, ce qui augmente la confiance des utilisateurs et stimule l'utilisation qu'ils peuvent en faire. Enfin, afin de rendre l'utilisation des méthodes plus simple, une quantification des paramètres, pour diverses séries de données et des SOP, permettra à l'utilisateur futur de gérer le contrôle et la surveillance des procédés plus facilement.

Bibliographie

Alferes, J., and Vanrolleghem, P.A. (2016). Efficient automated quality assessment: Dealing with faulty on-line water quality sensors. AI Commun. *29*, 701–709.

Alferes, J., Poirier, P., and Vanrolleghem, P.A. (2012). Efficient data quality evaluation in automated water quality measurement stations. In Proceedings of the International Environmental Modelling and Software Society (IEMSs2012), (Leipzig, Germany), p. 197.

Alferes, J., Lynggaard-Jensen, A., Munk-Nielsen, T., Tik, S., Vezzaro, L., Sharma, A.K., Mikkelsen, P.S., and Vanrolleghem, P.A. (2013a). Validating data quality during wet weather monitoring of wastewater treatment plant influents. In Proceedings WEFTEC2013, (Chicago, IL, United States), pp. 4507–4520.

Alferes, J., Tik, S., Copp, J., and Vanrolleghem, P.A. (2013b). Advanced monitoring of water systems using in situ measurement stations: data validation and fault detection. Water Sci. Technol. *68*, 1022–1030.

Bonastre, A., Ors, R., Capella, J.V., Fabra, M.J., and Peris, M. (2005). In-line chemical analysis of wastewater: present and future trends. Trends Anal. Chem. *24*, 128–137.

Bourgeois, W., Burgess, J.E., and Stuetz, R.M. (2001). On-line monitoring of wastewater quality: a review. J. Chem. Technol. Biotechnol. *76*, 337–348.

Bradley, A. (2018). Programming Manual : Logix 5000 Controllers Ladder Diagram. Milwaukee, Wisconsin, United States.

Checkley, M., and Checkley, W. (2008). Drinking Water and Sanitation. In International Encyclopedia of Public Health, (Elsevier), pp. 234–244.

Chow, C.W.K., Liu, J., Li, J., Swain, N., Reid, K., and Saint, C.P. (2018). Development of smart data analytics tools to support wastewater treatment plant operation. Chemom. Intell. Lab. Syst. *177*, 140–150.

Corominas, L., Villez, K., Olsson, G., Cortés, U., and Poch, M. (2018). Transforming data into knowledge for improved wastewater treatment operation: A critical review of techniques. Environ. Model. Softw. *106*, 89–103.

Daneels, A., and Salter, W. (1999). What is SCADA? In Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems, (Trieste, Italy), pp. 339–343.

Derlon, N., Thürlimann, C., Dürrenmatt, D., and Villez, K. (2017). Batch settling curve registration via image data modeling. Water Res. *114*, 327–337.

Dochain, D., and Vanrolleghem, P.A. (2001). Dynamical Modelling and Estimation in Wastewater Treatment Processes (London, UK: IWA Publ).

García, F.P., Pedregal, D.J., and Roberts, C. (2010). Time series methods applied to failure prediction and detection. Reliab. Eng. Syst. Saf. *95*, 698–703.

Garcia-Alvarez, D. (2009). Fault detection using principal component analysis (pca) in a wastewater treatment plant (wwtp). In Proceedings of the International Student's Scientific Conference, p.

Gray, H.F. (1940). Sewerage in ancient and medieval times. Sew. Works J. *12*, 939–946.

Hach (2004). Solitax sc. User Manual (4 ed.) (Loveland, CO, United States: Hach).

Hach (2006a). pHD sc Digital Differential pH/ ORP sensors (Loveland, CO, United States: Hach).

Hach (2006b). LDO Dissolved Oxygen Sensor. User Manual (6 ed.) (Loveland, CO, United States: Hach).

Hach (2008). 3700sc Digital Conductivity Sensor. User Manual (5 ed.) (Loveland, CO, United States: Hach.).

Hach (2014). Sigma 950. User Manual (3 ed.) (Loveland, CO, United States: Hach).

Hawkins, S., He, H., Williams, G., and Baxter, R. (2002). Outlier Detection Using Replicator Neural Networks. In Data Warehousing and Knowledge Discovery, Y. Kambayashi, W. Winiwarter, and M. Arikawa, eds. (Berlin, Heidelberg: Springer), pp. 170–180.

Henze, M., van Loosdrecht, M., Ekama, G., and Brdjanovic, D. (2008). Biological Wastewater Treatment: Principles, Modelling and Design (London, UK: IWA Publishing).

Hill, D.J., and Minsker, B.S. (2010). Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. Environ. Model. Softw. *25*, 1014–1022.

Ingildsen, P., and Olsson, G. (2016). Smart Water Utilities: Complexity Made Simple (London, UK: IWA Publishing).

Jackson, J.E., and Mudholkar, G.S. (1979). Control procedures for residuals associated with Principal Component Analysis. Technometrics *21*, 341.

Johnson, R.A., and Wichern, D.W. (2002). Applied Multivariate Statistical Analysis (Upper Saddle River, N.J: Pearson Prentice Hall).

Jolliffe, I.T. (2002). Principal Component Analysis (New York, NK, United States: Springer Science & Business Media).

Judd, C.M., McClelland, G.H., Ryan, C.S., Muller, D., and Yzerbyt, V. (2018). Analyse des données: Une approche par comparaison de modèles (Paris, France: De Boeck Superieur).

Krajewski, W.F., and Krajewski, K.L. (1989). Real-time quality control of streamflow data a simulation study. J. Am. Water Resour. Assoc. *25*, 391–399.

Ledergerber, J.M., Leray, E., Maruéjouls, T., and Vanrolleghem, P.A. (2017). Optimization of installation and maintenance of water quality sensors in combined sewers. In Proceedings IWA Conference on Urban Drainage Modelling (ICUD-2017), (Prague, Czech Repulic).

Ledergerber, J.M., Maruéjouls, T., and Vanrolleghem, P.A. (2018). Experimental design to support water quality modelling of sewer systems. In IWA Conference on Urban Drainage Modelling (UDM-2018), (Parlermo, Italy), pp. 627–632.

Lee, D.S., and Vanrolleghem, P.A. (2004). Adaptive consensus principal Component Analysis for on-line batch process monitoring. Environ. Monit. Assess. *92*, 119–135.

Lynggaard-Jensen, A. (1999). Trends in monitoring of waste water systems. Talanta *50*, 707–716.

Mirin, S.N.S., and Wahab, N.A. (2013). Fault detection and monitoring using Multiscale PCA. In Proceedings IEEE 4th Control and System Graduate Research Colloquium, (Shah Alam, Malaysia), pp. 1–5.

Mirin, S.N.S., and Wahab, N.A. (2014). Fault detection and monitoring using Multiscale Principal Component Analysis at a sewage treatment plant. J. Teknol. *70*, 87–92.

Montgomery, D.C. (1996). Introduction to Statistical Quality Control (3rd ed.) (New York, NY, United States: Wiley).

Montgomery, D.C. (2009). Introduction to Statistical Quality Control (6th ed.) (Hoboken, N.J, United States: Wiley).

Mourad, M., and Bertrand-Krajewski, J.L. (2002). A method for automatic validation of long time series of data in urban hydrology. Water Sci. Technol. *45*, 263–270.

Nathanson, J.A. (2003). Basic Environmental Technology: Water Supply, Waste Management, and Pollution Control (Upper Saddle River, NJ, United States: Prentice Hall).

Ni, K., Srivastava, M., Ramanathan, N., Chehade, M.N.H., Balzano, L., Nair, S., Zahedi, S., Kohler, E., Pottie, G., and Hansen, M. (2009). Sensor network data fault types. ACM Trans. Sens. Netw. *5*, 1–29.

Patry, B., Ridyard, G., Boutet, E., Lessard, P., and Vanrolleghem, P.A. (2018). Particulate matter accumulation and energy recovery potential in highly loaded enhanced aerated lagoons, ECO STP 2018. In Proceedings IWA Conference on Ecotechnologies for Wastwater Treatment (EcoSTO-2018), (London, ON, United States), p.

Patterson, D.A., Chen, P., Gibson, G., and Katz, R.H. (1989). Introduction to redundant arrays of inexpensive disks (RAID). In Proceedings Conference on IEEE Comput. Soc. Press, (San Francisco, CA, United States: IEEE Comput. Soc. Press), pp. 112–117.

Philippe, R. (2016). Projet de recherche pilEAUte : Suivi de l'élimination de la matière azotée au sein d'un pilote de traitement des eaux usées. Mémoire de ENS 3ème année (ENS3). Université Laval, Québec, Canada.

Plana, Q. (2013). Efficient on-line monitoring of river water quality using automated measuring stations. Mémoire. Universitat Politècnica de Catalunya, Barcelona, Spain.

Plana, Q. (2015). Automated data collection and management at enhanced lagoons for wastewater treatment. Mémoire. Université Laval, Québec, Canada.

Ponzeli, M. (2018). D5: Fermenting Clarifier First experimental results: pilot & lab scale. Rapport technique. Université Laval, Québec, Canada.

Ramalho, R.S. (2012). Introduction to Wastewater Treatment Processes (New York, NY, United States: Academic Press).

Rieger, L., and Vanrolleghem, P.A. (2008). monEAU: a platform for water quality monitoring networks. Water Sci. Technol. *57*, 1079–1086.

Rosen, C., and Lennox, J.A. (2001). Multivariate and multiscale monitoring of wastewater treatment operation. Water Res. *35*, 3402–3410.

Rosen, C., and Olsson, G. (1998). Disturbance detection in wastewater treatment plants. Water Sci. Technol. *37*, 197–205.

Saberi, A. (2015). Automatic outlier detection in automated water quality measurement stations. Mémoire. Université Laval.

s::can (2007). ammo::lyser V1 Manual (1 ed.) (Vienna, Austria: s::can.).

s::can (2011). Manual s::can spectrometer probe V2 (Vienna, Austria: s::can.).

Souidi, R. (2018). Caractérisations hydrauliques d'une station d'épuration pilote par boue activée : "Back Mixing". Rapport technique. Université Laval, Québec, Canada.

Tao, E.P., Shen, W.H., Liu, T.L., and Chen, X.Q. (2013). Fault diagnosis based on PCA for sensors of laboratorial wastewater treatment process. Chemom. Intell. Lab. Syst. *128*, 49–55.

Thomann, M. (2008). Quality evaluation methods for wastewater treatment plant data. Water Sci. Technol. *57*, 1601–1609.

Thomann, M., Rieger, L., Frommhold, S., Siegrist, H., and Gujer, W. (2002). An efficient monitoring concept with control charts for on-line sensors. Water Sci. Technol. *46*, 107–116.

Thürlimann, C.M., Dürrenmatt, D.J., and Villez, K. (2018). Soft-sensing with qualitative trend analysis for wastewater treatment plant control. Control Eng. Pract. *70*, 121–133.

Tohidi, M. (2018). Study of the TSS variation in the pilEAUte's primary settling tank. Rapport technique. Université Laval, Québec, Canada.

Unicef France (2017). 2,1 milliards de personnes n'ont pas accès à l'eau potable salubre. [En ligne]. https://www.unicef.fr/article/21-milliards-de-personnes-n-ont-pas-acces-l-eau-potable-salubre.

Vanrolleghem, P.A., and Vaneeckhaute, C. (2014). Resource recovery from wastewater and sludge: modelling and control challenges. In Proceedings IWA Specialist Conference on Global Challenges : Sustainable Wastewater Treatment and Resource Recovery, (Kathmandu, Nepal).

Villez, K. (2008). Multivariate and qualitative data analysis for monitoring, diagnosis and control of sequencing batch reactors for wastewater treatment. PhD thesis. Ghent University. Faculty of Bioscience Engineering (Ghent, Belgium).

Villez, K., and Habermacher, J. (2015). Shape constrained splines with discontinuities for anomaly detection in a batch process. In Computer Aided Chemical Engineering, K.V. Gernaey, J.K. Huusom, and R. Gani, eds. (Amsterdam, The Netherlands: Elsevier), pp. 1805–1810.

Villez, K., Rieger, L., Keser, B., and Venkatasubramanian, V. (2012). Probabilistic qualitative analysis for fault detection and identification of an on-line phosphate analyzer. Int. J. Adv. Eng. Sci. Appl. Math. *4*, 67–77.

Villez, K., Vanrolleghem, P.A., and Corominas, L. (2013). Structural observability and redundancy classification for sensor networks in wastewater systems. In Proceedings 11th IWA Conference on Instrumentation Control and Automation (ICA2013), (Narbonne, France), p. 4.

Xylem (2012). IQ Sensor NETVARiON®Plus 700 IQ (Yellow Springs, United States).

Yoo, C.K., Vanrolleghem, P.A., and Lee, I. (2003). Nonlinear modeling and adaptive monitoring with fuzzy and multivariate statistical methods in biological wastewater treatment plants. J. Biotechnol. *105*, 135–163.

Yoo, C.K., Villez, K., Van Hulle, S.W.H., and Vanrolleghem, P.A. (2008). Enhanced process monitoring for wastewater treatment systems. Environmetrics *19*, 602–617.

Annexes

A. Évaluation de l'effet du nettoyage d'après Plana (2015)

Cette annexe permet d'expliquer comment évaluer l'effet du nettoyage des capteurs. Cette méthode se base sur la méthode du diagramme de contrôle. Pour ce faire, plusieurs étapes sont nécessaires :

1. Analyser si les données sont normalement distribuées.

2. Déterminer la différence (%) entre les valeurs avant et après du nettoyage :

$$\%d_i = \frac{Valeur_{avant,i} - Valeur_{après,i}}{Valeur_{après,i}} \times 100$$

3. La ligne centrale est définie en 0 due à une définition d'un nettoyage n'ayant pas d'effet, cette différence devra être égale à 0.

$$Ligne\ centrale = 0$$

4. Sélecter 20 valeurs ayant une différence inférieure à 10 %.

5. Déterminer la déviation standard :

$$\sigma_{\overline{\%d}} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} \%d^2}$$

6. Déterminer les limites haute (UCL) et basse (LCL) de contrôle :

$$UCL = +L\sigma_{\overline{\%d}}$$
$$LCL = -L\sigma_{\overline{\%d}}$$

Avec:

L: paramètre égale à 2, correspondant à 95 % des probabilités que les valeurs peuvent être acceptées.

7. Construire le diagramme de contrôle avec les différentes lignes (UCL, LCL). Représenter aussi les pourcentages de différence entre les valeurs avant et après le nettoyage du capteur.

Remarque : Si un point est hors des limites hautes et basses ceci conclut que le capteur aurait dû être nettoyé plus tôt.

B. SOP méthode univariée

UNIVERSITÉ LAVAL DÉPARTEMENT DE GÉNIE CIVIL	**Data treatment with the univariate method** **SOP-0XX-**	
Date:	Révision: 01	Page 132 de 195

Data treatment with the univariate method

NOM DE L'APPAREIL	
MODEL	
N° SERIAL	
PRÉCISION ET REPRODUCTIBILITÉ	
DATE DE POSTE EN FONCTIONNEMENT	
DISTRIBUTION	
WEBSITE	
PROFESSEUR RESPONSABLE	Peter Vanrolleghem

	RÉALISATION	**RÉVISION**
NOM	Romain Philippe	Maryam Tohidi
FONCTION	MSc student	MSc student
DATE	07-08-2018	24-12-2018
SIGNATURE		

VALIDATION		
VALIDATION	**DATE**	**Nom Prénom**
01		

Introduction

The following document introduces a data treatment method called univariate method. With this method, different sets of data obtained by different sensors can be treated. Several steps need to be taken to obtain the treated data.

Note: for more information on the method, several articles and reports can be consulted:

- Publications in international journals with peer review:

Alferes Janelcy (2016)
Alferes, J. and Vanrolleghem, P. A. (2016) Efficient automated quality assessment: Dealing with faulty on-line water quality sensors. AI Communications, 2016
https://content.iospress.com/articles/ai-communications/aic713

Alferes Janelcy (2013)
Alferes, J., Tik, S., Copp, J. and Vanrolleghem, P. A. (2013) Advanced monitoring of water systems using in situ measurement stations: Data validation and fault detection. *Wat. Sci. Tech.*, 68(5), 1022-1030.
http://modeleau.fsg.ulaval.ca/fileadmin/modeleau/documents/Publications/pvr1043.pdf

- MSc:

Romain Philippe (2018)
"In the future "

Queralt Plana (2015)
"Automated data collection and management at enhanced lagoons for wastewater treatment."
https://modeleau.fsg.ulaval.ca/fileadmin/modeleau/documents/Publications/MSc_s/planaqueralt15_msc.pdf

Atefeh Saberi (2015)
"Automatic outlier detection in automated water quality measurement stations."
https://modeleau.fsg.ulaval.ca/fileadmin/modeleau/documents/Publications/MSc_s/saberiatefeh_msc.pdf

- Publications in books and conference proceedings

Alferes Janelcy (2017)
Alferes, J., Copp, J., Weijers, S., Cussonneau, G., Fay, G., Dembele, A. et Vanrolleghem,P.A. (2017) Validating data quality for water quality monitoring: Objective comparison of different data quality assessment approaches. In: *Proceedings 12th IWA Conference on Instrumentation, Control and Automation (ICA2017)*. Québec, Québec, Canada, June 11-14-2017. 215-220.

Alferes Janelcy (2015)
Alferes, J., Copp, J., Weijers, S. et Vanrolleghem, P. A. (2015) Validating data quality for water quality monitoring: Objective comparison of three data quality assessment approaches. In: *Proceedings New Developments in IT & Water Conference*. Rotterdam, The Netherlands, February 8-10-2015.
http://modeleau.fsg.ulaval.ca/fileadmin/modeleau/documents/Publications/pvr1187.pdf

Alferes Janelcy (2014)

Alferes, J., Lamaire Chad, C., Chhetri, R., Thirsing, C., Sharma, K., Mikkelsen, P. et Vanrolleghem, P. A. (2014) Advanced monitoring of wastewater quality: Data collection and data quality assurance. In: *Proceedings 13th International Conference on Urban Drainage (13ICUD)*. Sarawak, Malaysia, September 7-12-2014. http://modeleau.fsg.ulaval.ca/fileadmin/modeleau/documents/Publications/pvr1139.pdf

Alferes, J. et Vanrolleghem, P. A. (2014) Automated data quality assessment: Dealing with faulty on-line water quality sensors. In: *Proceedings 7th International Congress on Environmental Modelling and Software (iEMSs2014)*. San Diego, CA, USA, June 15-19 2014. http://modeleau.fsg.ulaval.ca/fileadmin/modeleau/documents/Publications/pvr1137.pdf

Alferes, J., Copp, J. et Vanrolleghem, P. A. (2014) Forecasting techniques applied to water quality time series in view of water quality assessment. In: *Proceedings 11th International Conference on Hydroinformatics (HIC 2014)*. New York, NY, USA, August 17-21 2014.
https://pdfs.semanticscholar.org/64bf/b211cfc57cb7e7c9f268d6671151c92e6a3b.pdf

Alferes, J., Copp, J., Weijers, S. et Vanrolleghem, P. A. (2014) Innovative water quality monitoring: Automation of data assessment in practical scenarios. In: *Proceedings IWA World Water Congress 2014*. Lisbon, Portugal, September 21-26 2014.
https://modeleau.fsg.ulaval.ca/fileadmin/modeleau/documents/Publications/pvr1138.pdf

Copp, J., Alferes, J. et Vanrolleghem, P. A. (2014) High quality monitoring of water systems using in situ automatic measurement stations that incorporate real-time data quality analysis tools. In: *Proceedings 9th National Monitoring Conference (NWQMC) - Working Together for Clean Water*. Cincinatti, OH, USA, April 28 - May 2014.

Alferes Janelcy (2013)

Alferes J., Poirier P., Lamaire-Chad C., Sharma A.K., Mikkelsen P.S. and Vanrolleghem P.A. (2013) Data quality assurance in monitoring of wastewater quality: Univariate on-line and off-line methods. In: Proceedings 11th IWA Conference on Instrumentation, Control and Automation (ICA2013). Narbonne, France, September 18-20 2013.

Alferes J., Lynggaard-Jensen A., Munk-Nielsen T., Tik S., Vezzaro L., Kumari Sharma A., Steen Mikkelsen P. and Vanrolleghem P.A. (2013)Validating data quality during wet weather monitoring of wastewater treatment plant influents In: Proceedings WEFTEC2013. Chicago, IL, October 3-9-2013
http://modeleau.fsg.ulaval.ca/fileadmin/modeleau/documents/Publications/pvr1098.pdf

Alferes, J., Copp, J. et Vanrolleghem, P. A. (2013) High quality monitoring of water systems using in situ automatic measurement stations. In: *Proceedings 5th CWWA Canadian Wastewater Management Conference & 48th CAWQ Central Canadian Symposium on Water Quality Research*. Hamilton, Ontario, Canada, March 6-8 2013.

Alferes Janelcy (2012)

Alferes, J., Poirier, P. et Vanrolleghem, P. A. (2012) Efficient data quality evaluation in automated water quality measurement stations. In: *Proceedings International Congress on Environmental Modelling and Software*

(iEMSs2012). Leipzig, Germany, July 1-2012. http://modeleau.fsg.ulaval.ca/fileadmin/modeleau/documents/Publications/pvr1029.pdf

Application Fields

The method is used to clean and improve time series. In other words, with this method the faults inside a time series can be detected. Four types of faults can be detected: bias, drift, complete failure and precision degradation. Data can come from different origins with sensors installed in different environments like rivers, sewers, WRRFs, etc…

Principle and theory

In Figure 1 the framework of the univariate method is presented. The method consists of in two steps:

- Data-filtering
- Fault detection

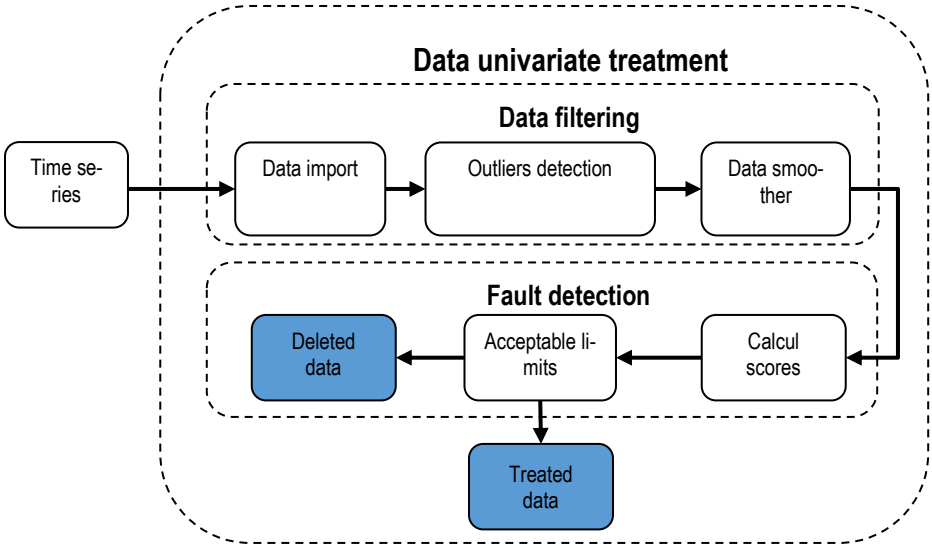


Figure 1. Framework of the univariate method

3.1) Data-filtering

Within data-filtering two tasks are executed: first the outliers are found by the **outlier's detection** stage, second the time series is smoothed through the **data smoother** stage.

- **Outliers detection**

The outliers are identified by comparing the measured values with forecast values considering the prediction error determined by the standard deviation of the forecast error. This method uses a first, second and third-order statistical exponential smoothing model to predict the forecast value:

$$S_T = \alpha x_T + (1-\alpha) S_{T-1}$$

$$S_T^{[2]} = \alpha x_T + (1-\alpha) S_{T-1}^2$$

$$S_T^{[3]} = \alpha x_T + (1-\alpha) S_{T-1}^3$$

With
x_T : current value of the data
S_{T-1} : Estimated data at T-1
T: Time
α: smoothing constant

More specifically, at time T, the forecast value at the next time step (T+1) is calculated by:

$$\hat{x}_{T+1} = \hat{a}_T + \hat{b}_T + \frac{1}{2}\hat{c}_T \hat{x}_{T+1} = \hat{a}_T + \hat{b}_T + \frac{1}{2}\hat{c}_T$$

With $\hat{a}_T$, b_T and $\hat{c}_T$ the coefficients of the model, are calculated with the three-statistical exponential method:

$$\hat{a}_T = 3S_T - 3S_T^{[2]} + S_T^{[3]}$$

$$\hat{b}_T = \frac{\alpha}{2(\alpha-1)^2}[(6-5\alpha)S_T - 2(5-4\alpha)S_T^{[2]} + (4-3\alpha)S_T^{[3]}]$$

$$\hat{c}_T = (\frac{\alpha}{\alpha-1})^2 (S_T - 2S_T^{[2]} + S_T^{[3]}$$

Finally, the outliers are detected by calculating the two limits (upper and lower):

- Upper limit:

$$\lim_{T} U = \hat{x}_T + K \times \hat{\sigma}_{e,T}$$

- Lower limit:

$$\lim_{T} L = \hat{x}_T - K \times \hat{\sigma}_{e,T}$$

With

$\hat{\sigma}_{e,T} = 1.25 \times \hat{\Delta}_T$: prediction error

$\hat{\Delta}_T = \beta |e_T(1)| + (1-\beta) \times \hat{\Delta}_{T-1}$: standard deviation estimation

With

$e_T(1) = x_T - \hat{x}_T$: one step ahead prediction error

Figure 2 represents an example of an outlier's detection.

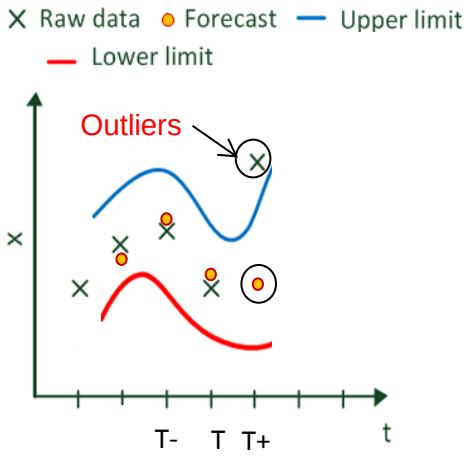


Figure 2. Illustration of the outlier detection method

- **Data smoother**

After the outlier detection and the replacement of these ones by forecast, the next step is to smooth the data to decrease the noise on the data. This one is carried out using a kernel smoother using Nadadya-Watson kernel estimator as a locally weighed average (Alferes & Vanrolleghem, 2016; Plana, 2015; Saberi, 2015). The equation below explains theory :

$$\hat{y}_h(x_0) = \sum_{i=1}^{n} W(x_0, x_i; h) * y(x_i)$$

With

$\hat{y}_h(x_0)$: locally weighed average of an observed point at x_0

n : the number of points

h : the bandwidth

$y(x_i)$: the observation at a point x_i

$W(x_0, x_i; h)$: the weighting functions

the data is a Kernel smoother with a proper bandwidth h:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} = \frac{1}{\sqrt{2\pi}} e^{-\frac{(idx)^2}{2h^2}}$$

With

$$x = \frac{idx}{h}$$

With

idx : The datapoints within the interval between [-h : +h]

h: a parameter to compute the bandwidth of the filter. Large values indicate that the filter weighs the smoothed data over a large quantity of observations while small values smooth much less

Figure 3 illustrates the of data filtering method (green curve).

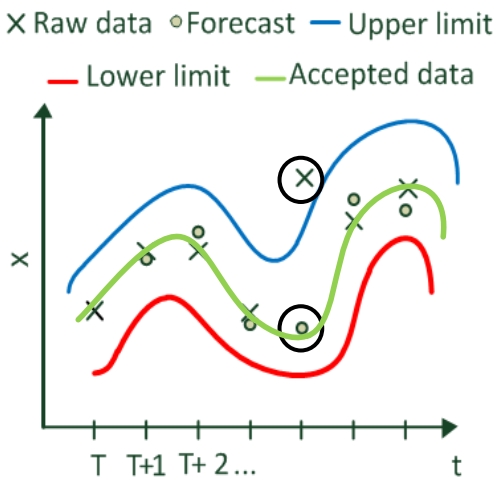


Figure 3. Illustration of the data filtering method

3.2) Fault detection

Inside the fault detection step, some scores are calculated in the **« data features calculation »** stage. For each score, some limits (Min and Max) are chosen in the **« break acceptable limits »** stage. Based on the scores some data are identified to be faults and are deleted from the data set. Finally, the treated data and the deleted data together with the scores in relation to their limits the are obtained in **« treated data »** step.

- **Data features calculation and acceptation limits**

Four scores are computed:

- Run-test : Evaluates whether residuals are randomly distributed. If the run test fails, residuals are auto-correlated and if so, either the smoothed data is not representative, or the noise is not randomly distributed. First, the sign of the difference between each smoothed data and accepted data point is calculated.

  $\mathrm{Sign(donnéesacceptées\text{-}donnéeslissées)}$ $\mathrm{Sign\ (Accepted\ data - Smoothed\ data)}$

Secondly, the score Q_{corr} is determined by the equation (Dochain & Vanrolleghem, 2001) :

$$Q_{corr} = \frac{R - \frac{N}{2}}{\sqrt{\frac{N}{2}}}$$

With

R : Number of sign changes in the series of calculated signs for the data series

N : Number of data in the selected window

- Slope: gives information about the dynamics of the data and helps detection of too sudden changes.

$$\frac{dx}{dt} \cong \frac{((Smoothen\ data)_i - (Smoothen\ datax)_{i-1})}{\Delta t}$$

- Standard deviation: Estimation of variance of the data, large standard deviation of residuals can be a sign of faulty data.

$$\text{Std} \cong \frac{\sum(x_i - \hat{x}_c)^2}{N - 1}$$

- Range: Investigates whether the data lies inside the expected range for the system under study.

x_i > Max or x_i < Min => Data is faulty

Finally, for the acceptation limits, the maximum and minimum limits of each test are chosen by the person using the method. A plot can be used by the user to identify the faulty data (see the next part).

- **Treated data**

The last stage is to obtain the treated data. The acceptation limit which is chosen in the previous step is used to evaluate treated data: when a score is above or below the limit, the data is not accepted, and is replaced by NaN.

- Treated data: data respecting all the scores.
- Deleted data: either the outliers or the data which exceed on of the score.

Explanation of data structures

In this section, the script to use the method with some helps and comments is provided. Two scripts have been created: the **(Univariate_method2)** allows to treat all the data at once, and the **(Univariate_method)** treats a selected time. For the two scripts, the method uses several functions. Each function is commented inside to understand the aim of the function, the input and its output.

- The structure includes the name and the variable of the script X. If one wants to change any of those properties**, they have to be changed in the whole script.**
- If one wants to change anything related with a function, **it should not be changed in the function, it should be changed in a new script.**

Figure 1 shows the different steps of the method. Within the next sections, the following steps will be explained in detail:

1) Load time series
2) Select times series to be treated
3) Data filtering
    - Set parameters
    - Outlier detection
    - Data smoother
4) Fault detection
    - Data feature calculation
    - Treated data
5) Plot tools

Nomenclature

In the script, several abbreviations have been used to reduce the size of the script (Table 1).

Table 1. Abbreviations in the scripts

Original name	*Data*	*Accepted Data*	*Smoothed Accepted Data*
Abbreviations	D	AD	Smoothed_AD

Load Times series

The first step in the method is to import the data into MATLAB. In this part, several functions are used. The main function for the data import is carried out by the function **"DataImport".** Table 2 presents the input and the output of this function:

*Table 2. Input and output of the function "**DataImport**"*

Function		***"DataImport"***
INPUT	Path	Data location in your computer
	'datEAUbaseCSVto-MAT'	The main function uses this function to import a .csv file in a matrix in MATLAB
	'SENSOR.mat'	The name of the structure
OUTPUT	SENSOR	The structure created by the function "**DataIm-port**". You choose the name of your structure.

Note: The imported file should be in a **.csv** format, e.g. generated by the dat*EAU*base:

Date and Time	***Sampling Point***	***Parameter / from***	***Value***	***Unit***
Aaaa-mm-jj hh:mm				

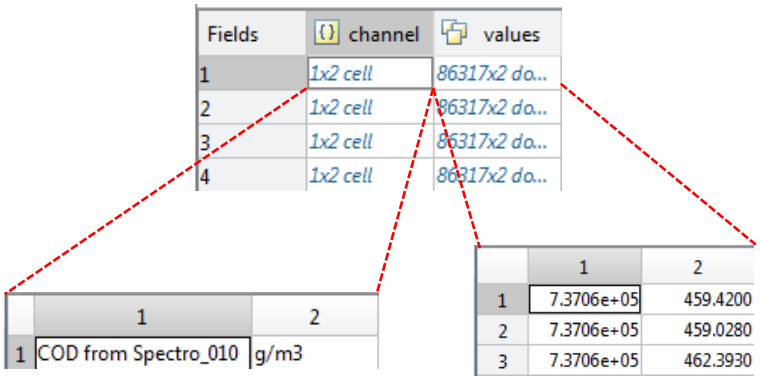
Below, some lines of the script for data importation are provided:

```
% The line below allows to import the raw data with the function
% DataImport; You have to put the address where your data are.

path = '.csv';
SENSOR = DataImport (path,'datEAUbaseCSVtoMAT','SENSOR.mat');
```

Note 1: If the order of the columns inside the .csv file is different; you should change this in the function **"datEAU-baseCSVtoMAT"**.

Note 2: If you already have a structure created, this one should be having the following format:



- **Complements:**

If you want to add new data, you can use the function **"Concatenate"**. Table 3 presents the input and the output of this function.

*Table 3. Input and output of the function «**Concatenate** »*

Function	***"Concatenate".***	
INPUT	SENSOR	The main structure
	Sen	The second structure with the new data
OUTPUT	SENSOR	The main structure

Below, some lines of the script to import new data into the first structure.

```
% Add new data

path1 = 'G:\Documents\......\New_data.csv';
Sen = DataImport (path1,'datEAUbaseCSVtoMAT','Sen.mat');
SENSOR = Concatenate (SENSOR, Sen);
save ('SENSOR.mat')% Save the data.
```

Select times series to be treated

The second step in the method is "**select times series to be treated**". Below, some lines are giving an example of how a time series is selected:

```
% Selection of the period of the data series to be treated

channel = 1; % Variable to be filtered
SENSOR(1).values = sortrows(SENSOR(1).values,1);
T0 = datenum('26-01-2017 00:00:00', 'dd-mm-yyyy HH:MM:SS');
TF = datenum('02-03-2018 00:00:00', 'dd-mm-yyyy HH:MM:SS');
TimeSeries = find(SENSOR(1).values > T0 & SENSOR(2).values < TF);
Sensor(1).values = SENSOR(1).values(TimeSeries,:);
```

Data filtering

In this section, the different steps of data filtering are explained. Three main steps, **"set parameters", "outlier detection"** and **"data smoother",** will be presented.

- **Set parameters**

Before carrying out the outlier detection stage, some parameters of the algorithm should be set. The list below defines all parameters:

- **param.nb_s :** Multiplicative factor that drives the calculation of the prediction interval. At each time step, the confidence interval is forecasted by +/- [nb_s * 1.25 * Mean_Absolute_Deviation]. In other words, a large value of nb_s (e.g. nb_s = 10) accepts most datapoints and rejects only the most obvious outliers while a small value of nb_s (e.g. nb_s = 1) is much more restrictive.

- **param.nb_reject :** Number of consecutive rejected data before the outlier detection method is re-initialized. If nb_reject data are rejected, this is called an out of control.

- **param.nb_backward :** Number of data before the last rejected data (the last of nb_reject data) where the outlier detection method is reinitialized for a forward application.

- **param.MAD_ini :** Mean of absolute deviation used to start a reinitialization of the outlier detection method.

- **param.min_MAD :** Minimum mean of absolute deviation to be used. If the computed MAD falls below this value, it is replaced by min_MAD. A specific example of this occurs when a constant value appears in the time series. Under this circumstance and without a min_MAD value, the MAD will fall to zero and

all datapoints different from the constant will be flagged as outlier until the next restart. The default value of 0 means that min_MAD will be initialized in the ModelCalibration function.

- **param.h_smoother :** The smoother parameter defines how many datapoints are used to smooth a specific value. The datapoints within the interval between [i-h_smoother : i+h_smoother] are used in the weighting formula.
- **param.Verbose :** Displays some warning and error messages when TRUE. The warnings displays are:

'DataCoherence warning: NaN values are present in the dataset'
'DataCoherence warning: Large gap is present in the dataset'
'DataCoherence warning: the timestep is not constant'
'DataCoherence warning: Negative time step found.'

- **param.DT_RelRol :** Permitted variation of the timestep for it to be considered constant. If the variation between a timestep and the median timestep is smaller than this parameter, the timestep is considered constant. Otherwise, it is considered variable and the algorithm must be used with caution: the filter currently assumes a constant timestep.
- **param.restart :** Set to TRUE if the filtering must be restarted from scratch. If set to FALSE, a sequential filtering is performed, and new filtered data is either appended to existing ones or replace them.
- **param.N_Reset :** If a series of data is refiltered, the exponential moving average filter must be applied to a number of datapoints in the so-called warmup period. The warmup period of the filter is defined by N in the equation: ALPHA = 1/(1+N). In theory, 86% of the warmup is done after N datapoints are filtered. To get closer to 100%, the parameter N_Reset allows to use more than one period, thus more datapoints based on the calibrated parameter ALPHA. No value larger than 4 or 5 should be used, since no further improvement will be observed.

The **"DefaultParam"** function is used to set the parameters. Table 4 presents the input and the output of this function.

*Table 4. Input and output of the function **«Defaultparam »***

Function	***"Defaultparam".***	
INPUT	OutlierDetectionMethod	Choice of the used method. Other methods can be chosen either Neural network, exponentially weighted moving variance and standard deviation
OUTPUT	param	All sets of parameters used by the method.

The following lines of the script allow to generate the default parameters.

```
%%%%%%%%%%%%%%%%    Generate default parameters    %%%%%%%%%%%%%%%%%%%%%%%%

% Load default parameters

paramX = DefaultParam('Online_EWMA');
```

Note: to modify a parameter, one should add this line to the script:

```
 % Set parameters: « Example »
paramX.nb_reject     = 60;
```

- **Outlier detection**

This stage allows to detect the outliers in the time series. The whole function is in the **"Outlier_detection"** folder. Firstly, the **"select a subset of the data"** stage allows to calibrate some parameters of the outlier detection method. The following lines show an example of a time series selection:

```
%%%%%%%%%%%%%% Select a subset of the data for calibration%%%%%%%%%%%%%%%

% The subset should be as large as possible to better represent the sys-
tem and sensor behavior

Sensor(channel).values = sortrows(Sensor(channel).values,1);
Tini = datenum('05-02-2018 00:00:00', 'dd-mm-yyyy HH:MM:SS');
Tend = datenum('11-02-2018 00:00:00', 'dd-mm-yyyy HH:MM:SS');
posSensorX = find(Sensor(channel).values > Tini & Sensor(channel).values
< Tend);
calibX = Sensor(channel).values(posSensorX,:);
```

Before performing the outlier detection stage, one stage allows to detect most common data problems by the "**DataCoherence"** function. It does not alter the database but returns meaningful errors or warning codes. Table 5**Erreur ! Source du renvoi introuvable.** presents the input and the output of this function.

Table 5. Input and output of the function ***« DataCoherence »***

Function	***"DataCoherence"***	
INPUT	Data	The data to filter is a two-columns matrix. The first column contains the time and the second the values.
	param	The structure containing the parameters of the filter. Some parameters allow to classify problematic situations as warning or errors.
OUTPUT	flag	The flag returns a code that depends on the DATA. 0: No error was encontered. 1: NaN were detected in the dates or in the observations. 2: A variable time step was detected. 3: A large gap in the data was detected. 4: A negative time step was detected.

The following lines of the script describe a method to test the data for missing values, NaN, etc.

```
%%%%%%%%%% Test the dataset for missing values, NaN, etc.%%%%%%%%%%
flag = DataCoherence(calibX, paramX);
if flag < 0
    return;
end
```

The function **"OutlierDetection"** allows to detect the outliers in a time series. Table 6 presents the input and the output of this function.

Table 6. Input and output of the function ***« OutlierDetection »***

Function	***"OutlierDetection"***	
INPUT	Data	Original data to filter. Column 1 = dates of the observations in MATLAB format. Column 2 = Raw observations.
	Calibperiod	The time series selected in the general script for the calibration model.
	Channel	The number of parameters that are to be treated in the structure.
	param	Structure of parameters. They have been initialized previously with the **"defaultparam"** function.
OUTPUT	Data	• ACCEPTED_DATA: Data without outliers, but unfiltered. • SEC_RESULT: A structure containing secondary results. Specifically: ▪ Orig : Original dataset, for reference ▪ Forecast_outlier: Forecast of the data based on the outlier filter. ▪ UpperLimit_outlier: limit above which an observation becomes an outlier. ▪ LowerLimit_outlier: Limit below which an observation becomes an outlier. ▪ outlier: Detected outliers. outlier(i) = 1 for detected outlier. Outlier(i) = 0 for accepted data. ▪ out_of_control_outlier: Data was in or out of control. 1 for "out of control", 0 for "in control" ▪ reini_outlier: Indicates points of reinitialization of the outlier filter when out of control.

The following lines show the outlier detection stage in the script.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%% Outlier detection %%%%%%%%%%%%%%%%%%%%%%%%%%%%

[WTW, paramX] = OutlierDetection(WTW, calibX, channel, paramX);
```

- **Data smoother**

This stage allows to smooth the data provided in the "outlier detection" stage. The theory of smoothening of data is explained previously (3.1). In this stage, the **"paramX.h_smoother"** parameter is used. The latter has been initialized in the "**defaultparam"** function. The parameter defines how many data points are used to smoothen a specific value. After initializing the parameters, the function **"kernel_smoother"** function is used for the data smoothing. Table 7 presents the input and the output of this function.

*Table 7. Input and output of the function **« kernel_smoother »***

Function	***"kernel_smoother"***	
INPUT	Data	A vector containing filtered data for outliers.
	channel	The number of parameters to be treated by the structure.
	param	The importation of param.h_smoother.
OUTPUT	SmoothedAD	The final smoothing data without outliers.
	err	Difference between the smoothed data and the accepted data.

The following lines show an example of the data smoother stage for the parameters.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%% DATA SMOOTHER   %%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%   Generate default parameters  %%%%%%%%%%%%%%%%%%%%%%

% Set parameters
paramX.h_smoother    = 30;

% Data filtration ==> kernel_smoother function.

[Sensor(channel).Smoothed_AD, err]=kernel_smoother(Sensor, channel,
paramX);
```

Note: The higher the values of the parameters, the smoother the data will be.

Fault detection

The fault detection step allows to detect the fault inside the times series. For that, two main stages are used in the method:

- Data feature calculation
- Treated data

• Data feature calculation

First, this stage allows to calculate some scores to detect faults in the time series. In the "principle and theory" section, different scores have been presented (Aim, Equation) (3.2). Next, some limits are set to determine the faults in the time series. A main function is used in this stage: **"D_score".** Table 8presents the input and output of this function.

Table 8. Input and output of the function ***« D_score »***

Function	***"D_score"***	
INPUT	Data	▪ AcceptedData: This variable contains the accepted data providing the outliers detection. ▪ Smoothed_AD: This variable contains the smoothed data providing the kernel smoother process. It is a pre-treatment.
	Param	This variable contains the parameters chosen by the operator, defined in the function **"DefaultParam"**
	channel	This variable contains the number of parameters that are to be treated in the structure.
OUTPUT	Score	▪ Q_corr: checks the independency of the residuals over different intervals. It evaluates whether the residuals are randomly distributed. The calculation is carried out by the "single_sample_runs_test" function. ▪ Q_slope: The slopes are calculated between two smoothed data values. It gives information on the dynamics of the data and helps the detection of a too sudden change. ▪ Q_std: The standard deviation of residuals. A large standard deviation of the residuals can be a sign of faulty data (measurement noise). ▪ Q_range: Investigates whether the data lies inside the expected range.

The following lines of codes show the determination of the scores. Before calculating the scores, the max and min range, which depends on the variable, should be set. Afterwards, the max and min should be set for the other scores. For that, one should use the **"plotD_score"** function. This is explained in the part **"plot tools".**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%% FAULT DETECTION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Definition range (min and max) for Q_range:
paramX.range_min = NaN;      % Minimum real expected value of the variable
paramX.range_max = NaN;      % Maximum real expected value of the variable

% Calcul Q_corr, Q_std, Q_slope, Q_range:
Sensor(channel).Score = D_score(Sensor, paramX, channel);

% Plot score with the function plotData_feature
% Definition limit of data feature:
paramX.corr_min= NaN;
paramX.corr_max= NaN;

paramX.slope_min= NaN ;  % Maximum expected slope based on a good data
series
paramX.slope_max= NaN;   % Minimum expected slope based on good data se-
ries

paramX.std_min = NaN;    % Maximum variation between accepted data and
smoothed data
paramX.std_max = NaN;    % Minimum variation between accepted data and
smoothed data
```

- **Treated data**

Finally, the last stage, **"Treated data",** allows to determine the treated data in a time series in which the outliers have been detected and faults have been removed. For the latter, the main function **"TreatedD"** allows to obtain the treated data. Table 9 presents the input and the output of the **"TreatedD.**

Table 9. Input and output of the function ***« Treated data »***

Function		**"Treated data"**
INPUT	Data	The structure containing: ▪ Raw data ▪ SMOOTHED_AD: smoothed data ▪ Q_corr: a calculated score ▪ Q_std: a calculated score ▪ Q_range: a calculated score ▪ Q_slope: a calculated score
	param	The different max and min of each score sets in the script.
	channel	The number of parameters which are to be treated in the structure.
OUTPUT	Final_D	A matrix in the main structure with two columns: Treateddata Deleteddata

The following lines show the **"TreatedD"** stage:

```
%%%%%%%%%%%%%%%%%%%%%%%%%% TREATED DATA  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% To allow to determinate the treated data and deleted data:
Sensor(channel).Final_D = TreatedD(Sensor, paramX,channel);
```

- **Complements:**

Complement 1

At the end of the script, the last function allows to calculate the percentage of outliers and deleted data. The function **"Interpcalculator"** carries. Table 10 presents the input and the output of this function.

Table 10. Input and Output of the function ***« Interpcalculator »***

Function		**"Interpcalculator"**
INPUT	Data	Raw data Outlier Deleted data
	channel	The number of parameters which are to be treated in the structure.
OUTPUT	Intervariable	% Outliers % Deleted data

The following lines show this stage:

```
% Percentage of outliers and deleted data
[Sensor(channel).Intervariable] = Interpcalculator (Sensor, channel);
```

- Complement 2

The second complement contains lines that allow to save all parameters in the main structure as well as saving the main structure itself. The following lines describe this function:

```
% Save the param in the struct:
[Sensor(channel).param] = paramX;

save ('Sensor.mat')% Save all data
```

Plot tools

Inside the method, some tools are used to plot the data after each step (Figure). Table 11 presents all sets of input and output data for each plot function.

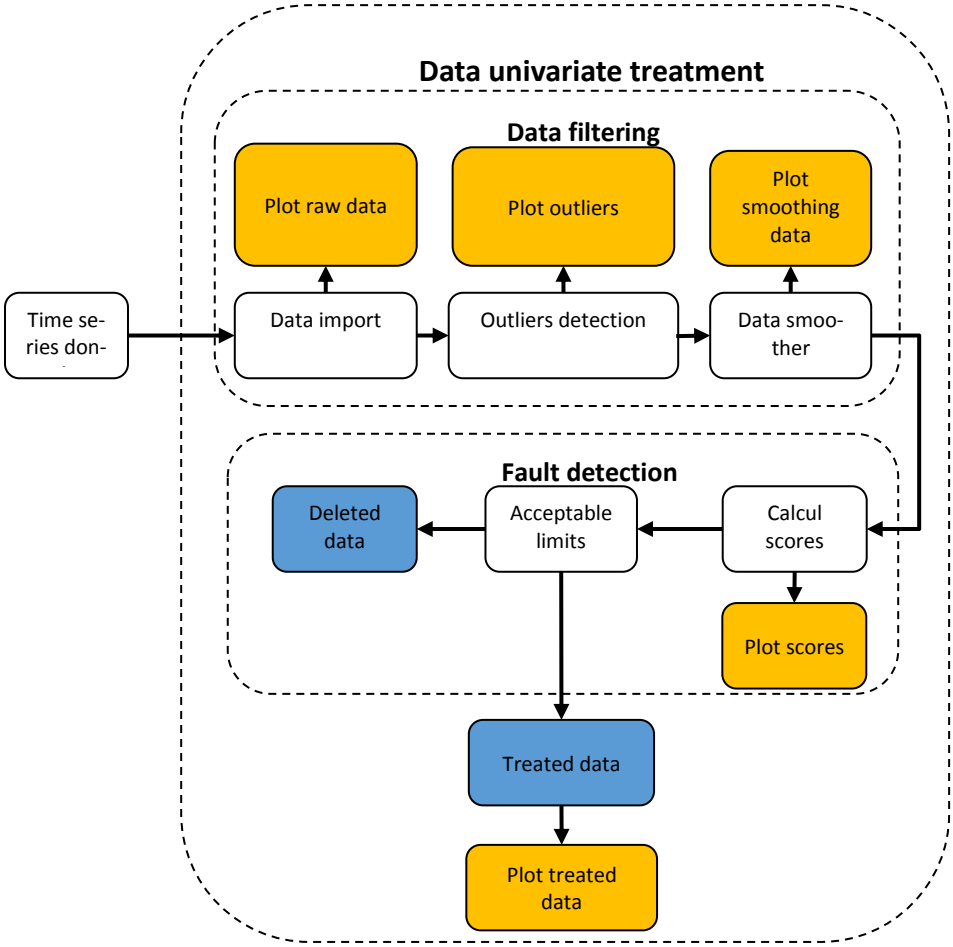


Figure 4. Univariate method diagram with plot function

Table 11. Input and output of the « plot » functions

Function	***"plotRawD"***	
INPUT	Data	The main structure: • Raw data
OUTPUT	Plot of the raw data.	

Function	***"plot_Outliers"***	
INPUT	Data	The main structure with: • Time • Accepted data • Outliers • Upper and lower limits
	channel	The number of parameters which are to be treated in the structure.
OUTPUT	Plot of the outliers, the accepted data with the upper and lower limits.	

Function	***"plotFiltered_D"***	
INPUT	Data	The main structure with: • Raw data • Accepted data • Smoothed data • Outliers • Upper and lower limits
	channel	The number of parameters which are to be treated in the structure.
OUTPUT	Plot of the raw data, smoothed data, outliers, the accepted data with the upper and lower limits	

Function	***"plotD_score"***	
INPUT	Data	The main structure with: • Time • Smoothed data • Four scores: Run test, slope, standard deviation and range.
	channel	The number of parameters which are to be treated in the structure.
	param	The parameters are the limits of each score.
OUTPUT	Plot the 4 scores with their limits.	

Function	***"plotTreatedD"***	
INPUT	Data	The main structure with: • Raw data • Treated data • Deleted data
	channel	The number of parameters which are to be treated in the structure.
OUTPUT	Plot the raw data, treated data and deleted data	

The following lines show all parts of the script by which some data can be plotted such as raw data, outliers, smoothed data, scores, and treated data.

- Plot raw data

```
% Plot raw data
plotRaw_D (Sensor)
```

- Plot Outliers detection

```
% Plot the outliers detected
Plot_Outliers(WTW, channel)
```

- Plot smoothed data

```
% Plot filtered data
plotFiltered_D(Sensor,channel);
```

- Plot scores

```
% Plot scores
plotD_score(Sensor, paramX, channel);
```

- Plot treated data

```
% Plot the raw data and treated data:
plotTreatedD(Sensor, channel)
```

Annexe

```
%%%%%%%%%%%%%%%%%%%%%%% UNIVARIATE METHOD FOR DATA %%%%%%%%%%%%%%%%%%%%%%%

% This script contains the different steps of the univariate method for
on-line data treatment.  It is a type script with the different step for
an example of parameter X of a Sensor.
% ------------------------------------------------------------------------
% ----------------------------- Sensor -----------------------------------
% ------------------------------------------------------------------------

% Generate the functions
addpath ('pilEAUte/DataFiltrationFramework')
SetFiltersPaths

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%% TIME SERIES: RAWDATA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Import the raw data
path = '.csv';
SENSOR = DataImport (path,'datEAUbaseCSVtoMAT','SENSOR.mat');

% Add new data
path1 = 'G:\Documents\......\New_data.csv';
Sen = DataImport (path1,'datEAUbaseCSVtoMAT','Sen.mat');
SENSOR = Concatenate (SENSOR, Sen);
save ('Sensor.mat') % Save the data.

% Plot raw data
plotRaw_D (SENSOR)



------------------------------------------------------------------------
----------------------------------X-------------------------------------
------------------------------------------------------------------------

% Selection of the period of the data series to be treated

channel = 1; % Variable to be filtered
SENSOR(1).values = sortrows(SENSOR(1).values,1);
T0 = datenum('26-01-2017 00:00:00', 'dd-mm-yyyy HH:MM:SS');
TF = datenum('02-03-2018 00:00:00', 'dd-mm-yyyy HH:MM:SS');
TimeSeries = find(SENSOR(1).values > T0 & SENSOR(2).values < TF);
Sensor(1).values = SENSOR(1).values(TimeSeries,:);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%%%%%% DATA FILTERING %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%%%%%% OUTLIERS DETECTION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%% Generate default parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Load default parameters

paramX = DefaultParam('Online_EWMA');

% Set parameters: « Example »
paramX.nb_reject     = 60;


%%%%%%%%%%%%%% Select a subset of the data for calibration %%%%%%%%%%%%%%%%
% The set should be as large as possible to better represent the system
and sensor behaviour

Sensor(channel).values = sortrows(Sensor(channel).values,1);
Tini = datenum('05-02-2018 00:00:00', 'dd-mm-yyyy HH:MM:SS');
Tend = datenum('11-02-2018 00:00:00', 'dd-mm-yyyy HH:MM:SS');
posSensorX = find(Sensor(channel).values > Tini & Sensor(channel).values
< Tend);
calibX = Sensor(channel).values(posSensorX,:);


%%%%%%%%%%%%%% Test the dataset for missing values, NaN, etc.%%%%%%%%%%%%%%
flag = DataCoherence(calibX, paramX);
if flag < 0
    return;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%% Outlier detection %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[WTW, paramX] = OutlierDetection(WTW, calibX, channel, paramX);

% Plot the detected outliers
Plot_Outliers(WTW, channel)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%%%% DATA SMOOTHER   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%% Generate default parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Set parameters
paramX.h_smoother = 30;

% Data filtation ==> kernel_smoother fucntion.

[Sensor(channel).Smoothed_AD, err]=kernel_smoother(Sensor, channel,
paramX);

% Plot filtered data
plotFiltered_D(Sensor,channel);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%%% FAULT DETECTION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Definition of range (min and max) for Q_range:
paramX.range_min = NaN;  % Minimum real expected value of the variable
paramX.range_max = NaN;  % Maximum real expected value of the variable

% Calcul Q_corr, Q_std, Q_slope, Q_range:
Sensor(channel).Score = D_score(Sensor, paramX, channel);

% Definition of limit of scores:
paramX.corr_min= NaN;
paramX.corr_max= NaN;

paramX.slope_min= NaN ;  % Maximum expected slope based on a good data
series
paramX.slope_max= NaN;   % Minimum expected slope based on good data se-
ries

paramX.std_min = NaN;    % Maximum variation between accepted data and
smoothed data
paramX.std_max = NaN;    % Minimum variation between accepted data and
smoothed data
```

```
% Plot scores
plotD_score(Sensor, paramX, channel);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%% TREATED DATA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% To allow to determine the treated data and deleted data:
Sensor(channel).Final_D = TreatedD(Sensor, paramX,channel);

% Plot the raw data and treated data:
plotTreatedD(Sensor, channel)

% Percentage of outliers and deleted data
[Sensor(channel).Intervariable] = Interpcalculator (Sensor, channel);

% Save the param in the struct:
[Sensor(channel).param] = paramX;

% save ('Sensor.mat') % Save all data

% Allow to clear the different created variables in the workspace.
clear flag calibX posSensorX dateSensorXini dateSensorXend paramX err i
```

C. SOP méthode multivariée

Data treatment with the multivariate method

NOM DE L'APPAREIL	
MODEL	
N° SERIAL	
PRÉCISION ET REPRODUCTIBILITÉ	
DATE DE POSTE EN FONCTIONNEMENT	
DISTRIBUTION	
WEBSITE	
PROFESSEUR RESPONSABLE	Peter Vanrolleghem

	RÉALISATION	RÉVISION
NOM	Romain Philippe	Gamze
FONCTION	MSc student	PhD student
DATE	21-09-2018	20 12 2018
SIGNATURE		

VALIDATION		
VALIDATION	**DATE**	**Nom Prénom**
01		
02		
03		

Introduction

This document explains the data treatment with the multivariate method which allows to treat data provided by different sensors. Application procedure of multivariate method includes several steps for data treatment.

Note: to have more information on the method, several articles or reports explain this method with some examples:

- MSc:

Romain Philippe (2018)
"In the future "

- Publications in books and conference proceedings

Alferes Janelcy (2013)
"Advanced monitoring of water systems using in situ measurement stations: Data validation and fault detection"
http://modeleau.fsg.ulaval.ca/fileadmin/modeleau/documents/Publications/pvr1043.pdf

Application Fields

The method is used to clean and improve time series. In other words, within this method the fault inside the time series could be detected. Four faults can be detected: bias, drifting, complete failure and precision degradation. The whole of data provides some sensors sensor installed in different environment like rivers, WRRFs, etc…

Principle and theory

The framework is shown in the Figure1 which is the principle of the multivariate method. The method is separated into 2 steps:

- PCA model development
- Fault detection

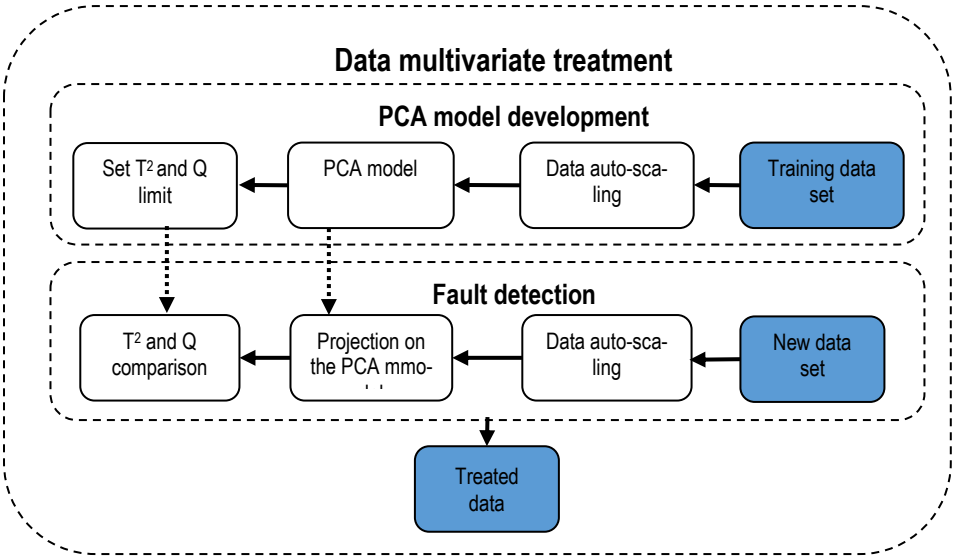


Figure 1. Framework of the univariate method

1) PCA model development

The aim of this step is to build a PCA model. Several steps should be carried out:

- Choice of training data set
- Data pre-processing: Normalization
- PCA model
- Set T² and Q limit

• Choice of training data set

The data used to build the PCA model represents the system. The whole raw data can be represented in a matrix:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,j} & \cdots & x_{1,M} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,j} & \cdots & x_{2,M} \\ \vdots & \vdots & & \vdots & & \vdots \\ x_{i,1} & x_{i,2} & \cdots & x_{i,j} & \cdots & x_{i,M} \\ \vdots & \vdots & & \vdots & & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,j} & \cdots & x_{N,M} \end{bmatrix}$$

With:

X : Theory matrix of raw data

N : Number of values

M : Number of variable

- **Data pre-processing: Normalization**

The second step is the normalization of data or more specifically the centering and scaling of theses ones. The normalization is based on two steps:

- Centering
- Scaling

The equation represents a variable centering and scaling:

$$Z_{ij} = \frac{(x_{ij} - \overline{x_j})}{\sigma_j}$$

With :

$x_{ij}\mathrm{X_{ej}}$: Raw data

$\dot{\mathrm{X}}_{\mathrm{j}}\ \overline{x_j}$: Average of each variable

σ_j : Standard deviation of each variable

- **PCA model**

The third step is to build the PCA model to obtain the principal components. A principal component is a linear combination of initial variables integrating a high variance. Firstly, the covariance matrix should be calculated with the equation:

$$\boldsymbol{Cx = \frac{1}{N-1} Z'Z}$$

With

Cx : covariance matrix

N : Number of samples

Z' : Normalized transposed matrix

Z : Normalized matrix

Secondly, a SVD (Singular Value Decomposition) decomposition on Cx should be calculated as follows:

$$\boldsymbol{Cx = V \Lambda V^T}$$

With

V: columns of matrix V are eigenvalues of Cx

$\mathrm{V^T}$: transposed of V

Λ : a diagonal matrix that contains in its diagonal the eigenvalues of Cx sorted in decreasing order ($\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \geq 0$).

Thirdly, a reduced dimension matrix P should be obtained by choosing the «a» eigenvalues of Cx associated with the largest «a» eigenvalues captures the largest fraction of the data variance. To choose the proper «a» value, the method is based on the eigenvalue scree plot (Jollife, 2002). Figure 2 shows an example of "eigen-value scree plot":

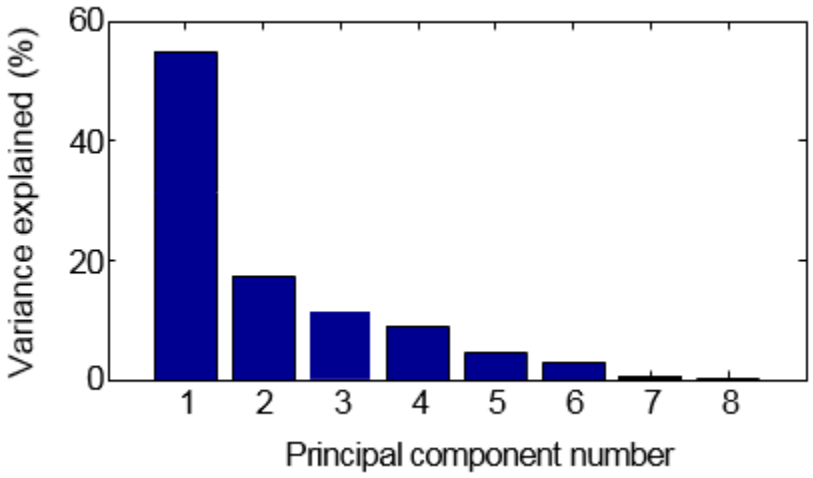


Figure 2. An example of "eigenvalue scree plot

As it can be seen from the Figure2, «a» should be chosen as 5, because the sum of percentage of the variance becomes 94 % (choice of the user).

Finally, the PCA model can be obtained with the equation below:

$$\overline{\boldsymbol{X}} = \boldsymbol{TP}^{\boldsymbol{T}} + \boldsymbol{E}$$

With:

$\overline{X}$: Normalized matrix

P: matrix of loading

T: matrix of scores of the PCA.

E: matrix of residual

The new reorganization of the information can be visualized as in Figure 3. Each variable is represented in the new coordinate space (PC1, PC2) by a vector (length and direction). This length and this direction indicate the contribution of the variable to the two first PCs (PC1, PC2) for each observation. Each point in the plot corresponds to a measurement.

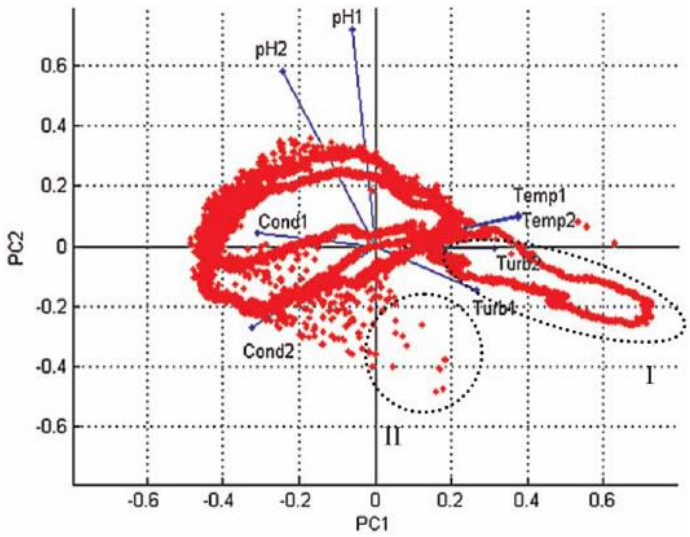


Figure3. Theory example of reorganization of the information in its components

- **Set T² and Q limit**

The analysis of the PCA is carried out with two tests and these ones are calculated automatically: T² and Q. Measurement of the variation within the PCA model is obtained at time by the T² statistic test which is defined as the sum of normalized squared scores:

$$\boldsymbol{T^2 = Z^T P \Lambda_a^{-1} P^T Z}$$

With

Z^T : Normalized transposed matrix

P : matrix of loading

Λ_a^{-1} : the diagonal matrix containing the «a» eigenvalues associated with the «a» eigenvectors or PCs retained in the model.

Z : normalized matrix

The Q test is defined as the random noise in the measurement.

$$\boldsymbol{Q = Z^T (I - PP^T) \times Z}$$

With :

I : identity matrix

A geometric interpretation of Q and T² can be seen in the Figure 4.

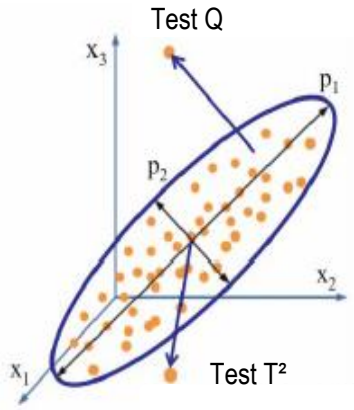


Figure 4. Geometrical interpretation of T² and Q statistics (Montgomery, 2009)

Finally, some limits can be calculated automatically by using the result of those two tests according to the following equation:

$$\boldsymbol{T_{\alpha}^{2}} = \frac{(\boldsymbol{N-1})\times \boldsymbol{N+1})\times \boldsymbol{C}}{\boldsymbol{N}\times(\boldsymbol{N-C})}\boldsymbol{F(\alpha, N-C)}$$

With

N : Number of variables

C : Number of chosen PC

F : Fisher-Snedecor distribution

α : Significient level

$$\boldsymbol{Q_{\alpha}} = \boldsymbol{\theta_1}\times[\boldsymbol{t_{\alpha}}\times\frac{\sqrt{\boldsymbol{2\times\theta_2\times h_0^2}}}{\boldsymbol{\theta_1}}+\boldsymbol{1}+\frac{\boldsymbol{\theta_2\times h_0\times(h_0-1)}}{\boldsymbol{\theta_1^2}}]^{\frac{\boldsymbol{1}}{\boldsymbol{h_0}}}$$

With

t_{α} : higher percentile for the normal standard distribution N(0,1) and a significant α

$h_0 = 1 - \frac{2\times\theta_1\times\theta_3}{3\times\theta_2^2}$

$\theta_i = \sum_{j=a+1}^{m}\lambda_j^i$

With

a : Number of chosen component

M : Number of variables

λ_j : j-i^{ème} eigenvalues

2) Fault detection

The aim of this step is to detect the fault inside the times series with the PCA model created before. Several steps are carried out:

- New data set
- Data auto-scaling
- Projection on the PCA model
- T^2 and Q comparison

• New data set

The one who is using this method may choose a new data set to be treated. **It should be noted that the variables have to stay the same**.

• Data auto-scaling

Explained in the "building of PCA model" Section.

• Projection on the PCA model

The projection with the PCA model is carried out by using the equation:

$$\boldsymbol{T = Z_{new} \times P}$$

With

T : The matrix of scores of new data

Z_{new} : normalized matrix of new data

P : loading matrix of PCA model

• T^2 and Q comparison

Finally, the last step is the comparison of the new T^2 and Q tests with the limits calculated beforehand.

$$T^2_{new} > T^2_{lim}$$

$$Q_{new} > Q_{lim}$$

Explanation of Data Structure

This part explains how to use the scripts and some comments to make use of the scripts easier. Inside the scripts, the method uses several functions. Every function is explained as a comment inside the script to understand its aim, input and output.

In this section, the general script will be descripted which is written in the scope of the MSc. Project.

- The structure includes the name and the variable of the script X. If one wants to change any of those properties**, they have to be changed in all script.**
- If one wants to change anything related with a function, **it should not be changed in the function, it should be changed in a new script.**

Figure 1 shows previously the different steps of this method. The different points will be presented within this section:

1) Setting the parameters
2) Initialization and building data
3) PCA model development

- Training data set
- Data pre-processing
- PCA model
- Statistic Tests

4) Fault detection

- New Data Set
- Data Auto-Scaling
- Projection on the PCA model and Statistic tests

Nomenclature

In the script, several abbreviations are used to reduce the size of the script. The abbreviations are shown with a table below:

Table 1. Abbreviations in the scripts

Original name	*Data*	*PCA*	*PC*
Abbreviations	D	Principal Component Analysis	Principal Component

Initialization and building data

Before beginning the initialization and building the data, it is necessary to generate the default parameters and indicate the number of variables in the data set that will be treated.

All the parameters are defined inside the function with «**DefaultsParamPCA**» as follows.

- **Param.Time:** this parameter allows to select the interval which we want to interpolate the data by the function «Initialization»
- **Param.Xstdmin:** the minimum of the standard deviation
- **Param.Normalisation:** this parameter allows to do the data normalization. If the operator does not want to do, one can add this parameter in the script and put false.
- **Param.p:** An appropriate level of significance for performing the test which typically the value of 0.95 or 0.99 for the warning and action limits respectively.
- **Param.calfa:** the standard normal deviate
- **Param.alfa:** this parameter alfa corresponds as a confidence level.
- **Param.n:** number of variables in the PCA analysis
- **Param.a:** number of principal components chosen

Below, some lines of the script can be seen including the parameters need to be set:

```
% Set the parameters:
Param = DefaultsParamPCA;

Param.n = 1:6; % number of variables
Param.Time = 1/24/60; % the interval time between two variables to inter-
polate the data
```

The first step in the method is the initialization and building data in MATLAB. To achieve this, the "**InitalizationD**" is used.

Table 2 presents the input and the output of this function:

*Table 1. Input and output of the function «**InitializationD**»*

Function	**«InitalizationD»**	
INPUT	Struct	The structure of the whole data (Smoothed data, treated data). In the line where y is defined, it is possible to change the Smoothed data by the name called in your structure.
	dt	The time interval between two variables. This allows to interpolate the whole data.
OUTPUT	DataforPC	A matrix with the whole data: First column: the time Other columns: The whole variables.

Note: The data have to be in single structure

Below, some lines of the script for the initialization data can be seen:

```
DforPCA = InitializationD(Sensor,1/24/60);

% save ('DateforPCA2.mat') save your matrix
```

PCA model Development

- **Training data set**

The first step of this part is to choose a training data. To achieve this, one should have trust in this training data This choice is carried out within the function. The function **« SelectTime»** is used to select the time series function depending on the desired time period. The Table 3 presents the input and the output of this function:

Table 2. Input and output of the function ***«SelectTime»***

Function	***"SelectTime"***	
INPUT	SX	The matrix with all data. The first column is the time and the rest are the data
	Tcal	This variable is chosen by the operator. Several specific periods inside the times series.
	Param	This input allows to import the data number. line 19 in the general script.
OUTPUT	D_PCA	Structure having all data with three parts: -TrainingD with all data (Time and Variables) -Time: Just a matrix the time -PCAprevious: Matrix with the data without the time

Below, some lines of the script for the training set are shown:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%% PCA MODEL DEVELOPMENT %%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%   TRAINING SET   %%%%%%%%%%%%%%%%%%%%%%%%%

%Set the specific periods of interest.

Tcal=[datenum('12-03-2018', 'dd-mm-yyyy') datenum('18-03-2018', 'dd-mm-
yyyy')];
% datenum('23-12-2017', 'dd-mm-yyyy') datenum('01-01-2018', 'dd-mm-
yyyy')];

D_PCA  = SelectTIME( DforPCA, Tcal, Param );
```

- **Data pre-processing**

The second step is the auto-scaling. In certain case, this step can be skipped. If not, the parameter **«Param.Normalisation»** needs to be considered. If it is equal to «true», the data will be normalized. To normalize the data, function **«NormalisationD»** is used. The Table 4 presents the input and the output of this function:

*Table 4. Input and output of the function «**NormalisationD**»*

Function	***«NormalisationD»***	
INPUT	Time	The time series chose
	SX	The matrix without the time
OUTPUT	SX1	Normalized matrix

A few lines of the script for the data pre-processing are shown below:

```
%%%%%%%%%%%%%%%%%%    DATA PRE-PROCESSING    %%%%%%%%%%%%%%%%%%%%%%%%%%

if Param.Normalisation == true

SX = NormalisationD (D_PCA, Param);

else
    SX = PCAprevious;
end
```

- **PCA model**

This third step to build the PCA model. To achieve this, the function **«ModelPCA»** is used. It allows to calculate the principal components, the scores, the eigenvalues, T^2 and the accumulated percentage of eigenvalues. Table 5 presents the input and the output of this function:

*Table 5. Input and output of the function **«ModelPCA»***

Function	***«ModelPCA»***	
INPUT	SX	This variable is the matrix with the normalized data
OUTPUT	coefs	The coefficients of principal compounds, also known as loadings, for the n-by-p data matrix X.
	scores	The scores of principal compounds
	eigenvalues	The eigenvalues of the covariance matrix of X
	T2	The sum of squares of the standardized scores for each observation.
	explained	The percentage of the total EV explained by each principal component
	mu	The estimated mean of each variable in SX
	percent_eigenvalues	The percentage of eigenvalues
	sum_eigenvalues	The cumulative of percentage of eigenvalues

Below, a few lines of the script for the PCA model is shown:

```
%%%%%%%%%%%%%%%%%%%%%%%%%     PCA MODEL     %%%%%%%%%%%%%%%%%%%%%%%%%%

%%Calculation of principal compounds, scores, eigenvalues, t2, accumu-
lated eigenvalues with the function ModelPCA.

Result_PCA = ModelPCA(SX );
```

- **Statistic Tests**

The last step of the PCA model development is the calculation of two statistics tests and their limits. Two functions are used to determine: **«StatisticTests»** and **«StatisticTestsLimits».** Table 6 and Table 7 present the input and the output of these functions:

*Table 6. Input and output of the function **«StatisticTests»***

Function	***«StatisticTests»***	
INPUT	SX	The matrix with the whole of data
	TrainingD	The times series selected.
	coefA	Selection by the operator in the script coefficients
OUTPUT	StatTest	A structure with: - Q: All Q values - T^2: AllT2 values

*Table 7. Input and output of the function **«StatisticTestsLimits»***

Function	***«StatisticTestsLimits»***	
INPUT	D_PCA:	The structure where there are the whole of data.
	Result_PCA	The structure with the results of PCA (scores, coefs, etc....)
	Param	Initialized in the script in the beginning by the Default-ParamPCA function.
OUTPUT	LimitTest	A structure with: -LimitQ: The limit of the Q test. -LimitT^2: The limit of the T^2 test.

A few lines of the script for the statistic tests are shown below:

```
%%%%%%%%%%%%%%%%%%%%    STATISTIC TESTS      %%%%%%%%%%%%%%%%%%%%%%%%%%

StatTest = StastisticTest(SX,D_PCA, Result_PCA, Param);

LimitTest = StatisticTestsLimits(D_PCA, Result_PCA, Param);
```

Fault detection

After developing the PCA model, the second step is to detect the fault inside the times series with the PCA model. Several steps and functions need to be carried out.

- **New Data Set**

First, the operator using the script should select a time series to be treated. It's the same procedure that in the **«Training data set»** part.

```
%%%%%%%%%%%%%%%%%%%%%%%%     FAULT DETECTION      %%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%       NEW DATA SET       %%%%%%%%%%%%%%%%%%%%%%%%

% This part allows to project new data on the PCA model built previously
%Set the specific periods of interest. For example, below, one periods
have been selected.

NewTcal=[datenum('01-01-2018', 'dd-mm-yyyy') datenum('20-02-2018', 'dd-
mm-yyyy')];

NewD  = SelectTIME( DforPCA, NewTcal, Param );
```

- **Data Auto-Scaling**

For the auto-scaling, the same procedure as in the part «**data pre-processing»** should be applied.

A few lines of the script for the Data Auto-Scaling are shown below:

```
%%%%%%%%%%%%%%%%%%%%%%%%      DATA AUTO-SCALING      %%%%%%%%%%%%%%%%%%%%%%%%

if Param.Normalisation == true

SX1 = NormalisationD (NewD, Param);

else
    SX1 = PCAprevious1;
end
```

- **Projection on the PCA model and Statistic tests**

As the last step, for the projection on the PCA model and statistic tests, the same procedure as the part **«statistic tests»** should be applied.

Below, some lines of the script for the Projection on the PCA model and the Statistic tests:

```
%%%%%%%%%%%%%%%%%%      PROJECTION ON THE PCA MODEL      %%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%         AND STATISTIC TEST     %%%%%%%%%%%%%%%%%%%%%%

% Statistic test: «Q and T2»

NewStatTest = StastisticTest(SX1,NewD, Result_PCA, Param);
```

Plot tools

Inside the method, some tools allow to plot the data after certain tests such as:

- PCA model
- Statistic tests

Table 8 presents the whole input and output of each plot function.

Table 3. Input and output of the whole functions plot

Function	***"plotScoresANDPC".***	
INPUT	percent_EV	Eigen values calculated in the function PCA.
OUTPUT	Plot the %Variance calculated by the function PCA and the sum of %Variance	

Function	***"plotScoresANDPC".***	
INPUT	coefs	The coefs have been calculated by the PCA function
	Scores	The scores have been calculated by the PCA function
OUTPUT	Plot of the outliers, the accepted data with the upper and lower limits.	

Function	***"plotPC"***	
INPUT	PC	The coefs have been calculated by the PCA function
OUTPUT	Graphical reprensentation of the principal compounds	

Function	***"plotScores"***	
INPUT	Score	The scores have been calculated by the PCA function
OUTPUT	Graphical reprensentation of the scores	

Function	**"plotQandT²"**	
INPUT	Time	The time of selected data
	limitQ	The limit determines in the PCA model building
	limitT^2	The limit determines in the PCA model building
	T^2	The values of T^2 for your times series
	Q	The values of Q for your times series
OUTPUT	Plot the raw data, treated data and deleted data	

The following lines show the whole parts of the script where one can plot some data such as scores, PC, limits of tests.

- Plotpar

```
% Plot pecent_EV

plotpar (Result_PCA)
```

- PlotScoresANDPC

```
% Graphical representation of scores and principal compounds

vbls = { 'TSS_Solitax' 'TSS_Spectro' 'NH4' 'DCO' 'DCOf' 'NO3'};
plotScoresANDPC(Result_PCA,vbls)
```

- PlotScores

```
% Graphical representation of scores

vbls = { 'TSS_Solitax' 'TSS_Spectro' 'NH4' 'DCO' 'DCOf' 'NO3'};
plotScores(Result_PCA,vbls)
```

- PlotPC

```
% Graphical representation of scores and principal compounds

vbls = { 'TSS_Solitax' 'TSS_Spectro' 'NH4' 'DCO' 'DCOf' 'NO3'};
plotPC (Result_PCA, vbls)
```

- PlotQANDT²

```
% Plot Q and T2 tests

plotQandT2(D_PCA, StatTest,  LimitTest)
```

Annexe

```
%%%%%%%%%%%%%%%%%%%%%%%%%% MULTIVARIATE METHOD %%%%%%%%%%%%%%%%%%%%%%%%%%
% This script is a general script for the data treatment with the PCA
method. Several STEP will be necessary to treat the data for this method.

% Set the parameters:

Param = DefaultsParamPCA;
%%%%%%%%%%%%%%%%%%%% Initilization and building of data %%%%%%%%%%%%%%%%

Param.n = 1:6; % number of variables
Param.Time = 1/24/60; %the interval time between two variables to
% interpolate the data

DforPCA = InitialisationD(Sensor,Param.Time);

save ('DateforPCA2.mat') % You can save your matrix

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%% PCA MODEL DEVELOPMENT   %%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%   TRAINING SET   %%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Set the specific periods of interest.

Tcal=[datenum('07-04-2018', 'dd-mm-yyyy') datenum('08-04-2018', 'dd-mm-
yyyy')];
% datenum('23-12-2017', 'dd-mm-yyyy') datenum('01-01-2018', 'dd-mm-
yyyy')];

D_PCA  = SelectTIME( DforPCA, Tcal, Param );


%%%%%%%%%%%%%%%%%%%%%   DATA PRE-PROCESSING   %%%%%%%%%%%%%%%%%%%%%%%%%%

if Param.Normalisation == true

SX = NormalisationD (D_PCA, Param);

else
    SX = PCAprevious;
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%    PCA MODEL    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Calculation of principal compounds, scores, eigenvalues, t2, accumu-
lated eigenvalues with the function ModelPCA.

Result_PCA = ModelPCA( SX );


% Plot pecent_EV:
plotpar (Result_PCA)

% Graphical representation of scores and principal compounds:
vbls = { 'TSS_Solitax' 'TSS_Spectro' 'NH4' 'DCO' 'DCOf' 'NO3'};
plotScoresANDPC(Result_PCA,vbls)
plotScores(Result_PCA)
vbls = { 'TSS_Solitax' 'TSS_Spectro' 'NH4' 'DCO' 'DCOf' 'NO3'};
plotPC (Result_PCA, vbls)

% Select number of principal components

Param.a = 1:5; % number of selected components

%%%%%%%%%%%%%%%%%%%%%%%% STATISTIC TESTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

StatTest = StastisticTest( SX,D_PCA, Result_PCA, Param);

LimitTest = StatisticTestsLimits(D_PCA, Result_PCA, Param);

% Plot Q and T2 tests
plotQandT2(D_PCA, StatTest,  LimitTest)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%    FAULT DETECTION      %%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%     NEW DATA SET       %%%%%%%%%%%%%%%%%%%%%%%%%%

% This part allows to project new data on the PCA model built previously
%Set the specific periods of interest. For example, below, one periods
have been selected.

NewTcal=[datenum('01-01-2018', 'dd-mm-yyyy') datenum('31-03-2018', 'dd-
mm-yyyy')];

NewD  = SelectTIME( DforPCA, NewTcal, Param );

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%    DATA AUTO-SCALING      %%%%%%%%%%%%%%%%%%%%%%%%%%

if Param.Normalisation == true

SX1 = NormalisationD (NewD, Param);

else
    SX1 = PCAprevious1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%     PROJECTION ON THE PCA MODEL     %%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%     AND STATISTIC TEST    %%%%%%%%%%%%%%%%%%%%%%%%%%%

% Statistic test: «Q and T2»

NewStatTest = StastisticTest( SX1,NewD, Result_PCA, Param);

% Plot Q and T2 tests:
plotQandT2(NewD, NewStatTest,  LimitTest)
```