



HAL
open science

Génération de données pour le diagnostic et le pronostic : un exemple applicatif

Michel Batteux, Yani Souami, Josquin Foulliaron, William Lair

► To cite this version:

Michel Batteux, Yani Souami, Josquin Foulliaron, William Lair. Génération de données pour le diagnostic et le pronostic : un exemple applicatif. Congrès Lambda Mu 22 “ Les risques au cœur des transitions ” (e-congrès) - 22e Congrès de Maîtrise des Risques et de Sécurité de Fonctionnement, Institut pour la Maîtrise des Risques, Oct 2020, Le Havre (e-congrès), France. hal-03454685

HAL Id: hal-03454685

<https://hal.science/hal-03454685>

Submitted on 29 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Génération de données pour le diagnostic et le pronostic : un exemple applicatif

Data generation for diagnosis and prognosis: an application example

Michel Batteux
IRT SystemX
Palaiseau, France
michel.batteux@irt-systemx.fr

Josquin Foulliaron
EdgeMind
Vannes, France
josquin.foulliaron@edgemind.net

William Lair
EDF R&D
Palaiseau, France
william.lair@edf.fr

Yani Souami
Air Liquide Research & Development
Les Loges en Josas, France
yani.souami@airliquide.com

Résumé—La complexité des installations industrielles de production est en constante augmentation. La raison principale tient dans la transformation numérique de ces systèmes. Un des enjeux actuels majeurs est d'améliorer les méthodes et outils utilisés pour le maintien en condition opérationnelle de tels systèmes de production. Dans cette publication, nous allons présenter un exemple applicatif, nommé 3-Réservoirs, qui représente un système dynamique hybride dont le modèle nous permet, d'une part d'aborder des approches de pronostic de défaillances basées sur des modèles comportementaux, et d'autre part de générer des données pour aborder des approches basées sur des méthodes d'apprentissage automatique.

Keywords—Maintenance, Génération de données, PDMP, PyCATSHOO

I. INTRODUCTION

La complexité des installations industrielles de production est en constante augmentation. La raison principale tient dans la transformation numérique de ces systèmes, avec l'introduction de mécanismes de contrôle omniprésents et de composants intégrant intensivement des parties logicielles. Un des enjeux actuels majeurs est d'améliorer les méthodes et outils utilisés pour le maintien en condition opérationnelle de tels systèmes de production. Ceci afin que ces méthodes et outils soient capables de prendre en compte ces nouvelles sources de complexité.

La maintenance est définie par la norme NF EN 13306 X 60-319 comme « l'ensemble des actions techniques, administratives et de management durant le cycle de vie d'un bien, destinées à le maintenir ou à le rétablir dans un état dans lequel il peut accomplir la fonction requise ». Il existe deux types de maintenance :

Summary—The complexity of large scale production systems is increasing. The main reason is the digital transformation of these systems. One of the major actual challenges is to enhance methods and tools for the operational condition strategies of these complex systems. This publication presents an application example, namely 3-Tanks, depicting a hybrid dynamic system. The model of this example will make it possible to study fault prognosis methods based on behavioral models. Furthermore this model makes it possible to generate data required for machine learning methods.

- La maintenance corrective qui consiste à intervenir uniquement après l'occurrence de la panne. Cette maintenance est généralement peu coûteuse, mais elle peut engendrer des coûts importants de non-fiabilité et d'indisponibilité.
- La maintenance préventive qui se décline en trois catégories. La maintenance systématique, généralement conseillée/imposée par le constructeur, et qui consiste à planifier des actions de maintenance périodiquement afin d'augmenter la fiabilité et la disponibilité des équipements. La maintenance conditionnelle qui utilise un système de surveillance afin de détecter d'éventuelles anomalies. L'action de maintenance est dans ce cas déclenchée uniquement lorsque l'état de dégradation observé du système dépasse un seuil critique manuellement défini. Enfin la maintenance prévisionnelle qui consiste à prédire l'instant des défaillances afin de les anticiper et de planifier des actions de maintenance.

Dans cette publication nous allons présenter un exemple applicatif, nommé 3-Réservoirs, qui représente un système dynamique hybride composé de trois réservoirs alimentés par deux pompes et trois vannes. Ce modèle nous permettra d'une part d'aborder des approches de PHM (Prognostics and Health Management) basées sur des modèles comportementaux. D'autre part de générer des données pour aborder des approches basées sur des méthodes d'apprentissage automatique.

La suite de la publication se décompose comme suit. La deuxième partie présente le projet 'Maintenance Prévisionnelle et Optimisation' dans lequel ces travaux sont réalisés. La troisième partie présente l'exemple 3-Réservoirs. La quatrième partie propose la modélisation en PyCATSHOO de cet exemple 3-Réservoirs. La cinquième partie présente des générations de données à partir du modèle. La sixième partie présente des premières exploitations de ces données générées. Enfin la dernière partie conclura cette publication.

II. LE PROJET MAINTENANCE PREVISIONNELLE ET OPTIMISATION

Les travaux présentés dans cette publication sont réalisés au sein du projet de recherche MPO, pour Maintenance Prévisionnelle et Optimisation, de l'IRT SystemX. Ce projet, en partenariat avec plusieurs acteurs industriels et académiques, porte sur l'optimisation des politiques de maintenance des systèmes de production, avec un focus particulier sur la maintenance prévisionnelle. Les travaux du projet se concentrent sur 3 principaux axes. Le premier axe concerne la structuration de la chaîne d'acquisition des données complexes et hétérogènes. Le deuxième axe concerne les méthodes et outils liés à la chaîne complète de maintenance prévisionnelle au niveau des composants. Enfin le troisième axe concerne la mise en place d'une chaîne d'optimisation de politiques de maintenance, au niveau du système industriel.

Nous cherchons donc, et plus précisément, à étudier différentes combinaisons de différentes politiques de maintenances possibles de tels systèmes industriels, afin d'obtenir des scénarios optimaux de stratégies de maintenance, suivant des critères établis : coûts, disponibilité, fiabilité, etc. Dans ce cadre, nous appréhendons d'une part les maintenances possibles au niveau des composants d'un système : surveillance et diagnostic de pannes, ou pronostic de prochaines pannes via estimation de la RUL (la durée de vie résiduelle, ou 'Remaining Useful Lifetime' en anglais). À ce niveau composant, nous basons nos études de surveillance, diagnostic et pronostic, sur l'étude des données complexes et hétérogènes. Ces données proviennent à la fois de capteurs et d'instruments de mesures introduits dans les systèmes, et de données externes contextuelles : données issues de l'environnement physique du système (température extérieure, pression atmosphérique, etc.) ou de l'environnement d'exploitation du système (rapports d'intervention, disponibilité des stocks, etc.). D'autre part, le projet appréhende la combinaison des maintenances des composants au niveau d'un système, avec pour objectif, comme pré-indiqué, d'obtenir des scénarios optimaux de stratégies de maintenances, par utilisation d'algorithmes d'optimisation.

III. PRÉSENTATION DE L'EXEMPLE 3-RÉSÉROIRS

Nous présentons dans cette partie l'exemple 3-Réservoirs que nous avons conçu dans le projet MPO. Cet exemple,

même s'il est adapté, est un classique dans la littérature du diagnostic et/ou du pronostic.

A. Description du système

L'exemple 3-Réservoirs est un système composé de trois réservoirs (L1, L2 et L3), de deux pompes (P1 et P2) et de trois vannes (V1, V2 et V3). Le réservoir L3 possède une source de chaleur qui fonctionne en continu. L'objectif du système est de refroidir, avec de l'eau, le réservoir L3 et d'éviter l'apparition de l'un des 3 événements redoutés suivants :

- Le niveau d'eau de L3 a débordé ;
- Le réservoir L3 est vide ;
- La température dans L3 a dépassé un niveau critique.

L'eau circule de la façon suivante :

- Les deux réservoirs (L1 et L2) sont alimentés par deux sources froides indépendantes grâce à deux pompes (P1 pour L1 et P2 pour L2) ;
- Les réservoirs L1 et L2 alimentent en eau un troisième réservoir (L3) dans lequel se situe la source de chaleur ;
- L'alimentation de L3 par L1 est gérée par une vanne (V1). L'alimentation de L3 par L2 est gérée par une vanne (V2) ;
- L'évacuation de l'eau dans L3 est gérée par une vanne (V3) ;

Un capteur CH1 (respectivement CH2 et CH3) permet de mesurer la hauteur d'eau dans L1 (respectivement L2 et L3). Un capteur CT3 permet de mesurer la température de l'eau dans L3.

Initialement, les deux pompes P1 et P2 fonctionnent et les vannes V1, V2 et V3 sont ouvertes.

La Figure 1 représente le système « 3-Réservoirs » avec ses composants.

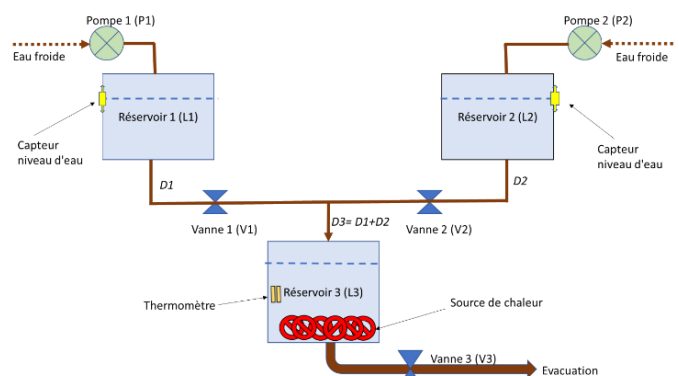


Figure 1 – Représentation de l'exemple 3-Réservoirs

B. Fonctionnement et dysfonctionnement des vannes

Les ouvertures et fermetures des vannes V1, V2 et V3 dépendent du niveau d'eau dans le réservoir L3. Lorsqu'une vanne est ouverte, elle laisse passer l'eau à un débit D_i ($i=1,2$ ou 3). À l'inverse, une vanne fermée ne laisse rien passer. Les vannes V1 et V2 se ferment quand la hauteur d'eau dans le réservoir L3, notée HL3, dépasse une valeur notée HL3max. Elles s'ouvrent quand HL3 est inférieure à une valeur notée

HL3min. La vanne V3 s'ouvre quand HL3 dépasse HL3max. Elle se ferme quand HL3 est inférieure à HL3min.

De plus, les vannes peuvent subir des défaillances intempestives. En effet, une vanne peut soit se coincer dans l'état dans lequel elle se trouve au moment de la défaillance, soit changer brusquement d'état (s'ouvrir si elle est fermée ou se fermer si elle est ouverte) et se coincer. Aucune réparation n'est modélisée dans la version actuelle. Les taux de transition des états sains vers les états des pannes des vannes V1 et V2, notés Λ_{V_1} et Λ_{V_2} , dépendent du nombre de manœuvres (ouvertures et fermetures confondues) noté N_i , et du taux de panne initial de V1 et V2 noté λ_{V_1} et λ_{V_2} . Pour $i=1$ et 2 :

$$\Lambda_{V_i} = \lambda_{V_i} \cdot e^{\beta_{V_i} N_{V_i}}$$

Le taux de panne de V3, noté Λ_{V_3} , dépend de la température dans le réservoir L3, notée T , et du taux de panne initial de V3 noté λ_{V_3} :

$$\Lambda_{V_3} = \lambda_{V_3} \frac{b_1 e^{b_c(T-20)} + b_2 e^{b_d(20-T)}}{b_1 + b_2} \cdot e^{\beta_{V_3} N_{V_3}}$$

Avec $b_1 = 3.0295$, $b_2 = 0.7578$, $b_c = 0.05756$ et $b_d = 0.2301$. Les paramètres λ_{V_i} et β_{V_3} ne sont pas fixés et pourront être modifiés pour différents tests. La Figure 2 (respectivement Figure 3) représente l'automate de transition des vannes V1 et V2 (respectivement de la vanne V3).

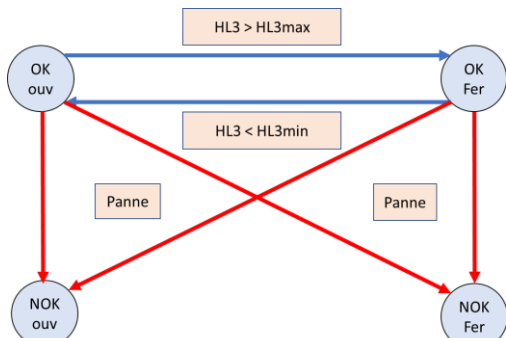


Figure 2 – Automate de transition des vannes V1 et V2

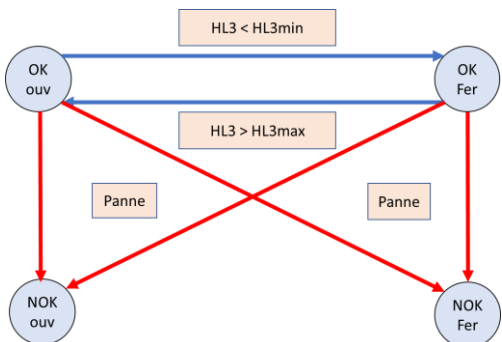


Figure 3 - Automate de transition de la vanne V3

C. Fonctionnement des pompes

Le fonctionnement des pompes est similaire à celui des vannes V1 et V2 et est représenté en Figure 4. La pompe P1 (respectivement P2) se met en fonctionnement quand la hauteur d'eau dans le réservoir L1 (respectivement L2) est inférieure à HL1min (respectivement HL2min). La pompe P1 (respectivement P2) s'arrête quand la hauteur d'eau dans le réservoir L1 (respectivement L2) est supérieure à HL1max (respectivement HL2max). Les pannes des pompes P1 et P2

se comportent comme celles des vannes V1 et V2. De même, les taux de défaillance ont la même forme, pour $i = 1$ ou 2 :

$$\Lambda_{P_i} = \lambda_{P_i} \cdot e^{\beta_{P_i} N_{P_i}}$$

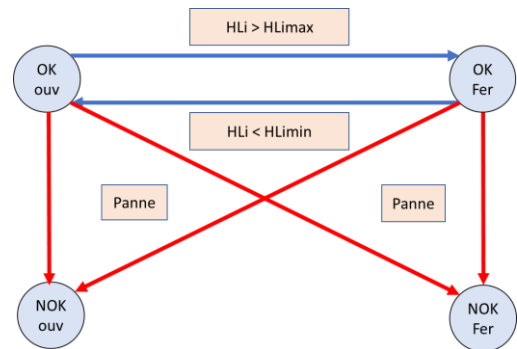


Figure 4 - Automate de transition des pompes P1 et P2

D. Fuites des réservoirs

Nous considérons que les réservoirs peuvent fuir. Chaque réservoir a un risque d'occurrence d'une fissure qui laisse passer l'eau. Nous considérons qu'un réservoir qui fuit ne risque plus de voir de nouvelles fissures apparaître. La position et la taille de la fissure sont considérées aléatoires.

Les fuites sont modélisées par un automate simple à deux états pour chaque réservoir (voir Figure 5) : « Absence de fissure » et « Présence de fissure ». Le taux d'occurrence d'une fissure associée au réservoir i est noté λ_{f_i} . La position de la fissure, notée H_{F_i} , et le débit théorique de fuite, noté d_{F_i} , sont tirés aléatoirement selon une loi uniforme. Le débit effectif, noté D_{F_i} de la fuite est égal au débit théorique de la fuite si le niveau d'eau dans le réservoir fissuré est supérieur à la position de la fuite.

$$D_{F_i} = d_{F_i} \mathbf{1}_{HL_i > H_{F_i}}$$



Figure 5 – Automate de l'occurrence d'une fissure sur un réservoir

E. Evolution de la hauteur d'eau dans les réservoirs

La hauteur d'eau dans les réservoirs évolue dans le temps en fonction des états des pompes et des vannes. Plus généralement, elle dépend du débit d'entrée et de sortie. Soit D_{P_i} le débit associé à la pompe P_i et soit D_{V_i} le débit associé à la vanne V_i , la hauteur de l'eau dans les réservoirs L1 et L2 évolue selon l'équation suivante, pour $i = 1$ ou 2 :

$$\frac{dHL_i}{dt} = D_{P_i} - D_{V_i} - D_{F_i}$$

La hauteur d'eau dans le réservoir L3 évolue selon l'équation suivante :

$$\frac{dHL_i}{dt} = D_{V_1} + D_{V_2} - D_{V_3} - D_{F_3}$$

Les débits sont constants tant que la vanne ou la pompe associée ne change pas d'état. Si la vanne est fermée ou que la pompe ne fonctionne pas, le débit associé est nul.

F. Evolution de la température dans le réservoir L3

L'évolution de la température dans le réservoir L3, notée T , dépend du débit d'entrée qui est la somme des débits

associés aux vannes V1 et V2, de la température de l'eau qui rentre, notée T_{in} , de la puissance de la source chaude et de la hauteur de l'eau :

$$\frac{dT}{dt} = \frac{(D_{V_1} + D_{V_2})(T_{in} - T) + power}{HL_3}$$

IV. MODÉLISATION DE L'EXEMPLE 3-RÉSERVOIRS

Afin de disposer d'un cas d'usage avec un jeu de données « complet », c'est-à-dire avec des données de pannes et de capteurs, nous avons défini un système fictif dont le fonctionnement et le dysfonctionnement ont été modélisés par un PDMP (Piecewise Deterministic Markov Process). Les données ont été générées en simulateur, par Monte-Carlo, ce PDMP à l'aide de PyCATSHOO [1] (Pythonic Object Oriented Hybrid Stochastic Automata) développé par EDF R&D.

A. Le formalisme de modélisation PDMP

Les PDMP ont été conçus et étudiés par [2] mais c'est [3] qui fut le premier à utiliser ce formalisme en fiabilité afin de résoudre des problèmes issus du domaine de la sûreté nucléaire. Un PDMP est un processus hybride $(I_t, X_t)_{t \geq 0}$. La première composante I_t est discrète à valeurs dans un espace fini ou dénombrable E , et décrit généralement l'état du système à l'instant t (par exemple marche ou panne). La deuxième composante X_t est continue à valeurs dans \mathbb{R}^d , elle décrit l'évolution des variables environnementales qui vont influencer et être influencées par le système (par exemple la température). Les deux composantes sont en mutuelle interaction : le taux de transition d'un état i vers un état j dépend de la composante continue x , on le note $a(i, j, x)$. Entre deux sauts de la partie discrète $(I_t)_{t \geq 0}$, la trajectoire de $(X_t)_{t \geq 0}$ est déterministe, solution d'une équation différentielle qui dépend de l'état discret i . Lors d'un saut de I_t , de i vers j , la variable continue X_t peut aussi sauter et est distribuée selon une loi qui dépend de la composante discrète avant et après le saut et de la composante continue juste avant le saut. Cette loi se note $\alpha_{(i, j, x)}(dy)$. L'espace d'état de la variable continue peut être borné et admettre une frontière. Dans ce cas, quand X_t atteint cette frontière notée Γ , le processus peut sauter selon la distribution $q_{(i, j, x)}(dy)$ avec $x \in \Gamma$, qui dépend de l'état avant le saut i et de l'état après le saut j .

B. L'outil PyCATSHOO

EDF R&D a développé un package Python appelé PyCATSHOO pour Pythonic Object Oriented Hybrid Stochastic Automata (<http://pycatshoo.org/>) [1] facilitant l'implémentation des PDMP, dont les quantités d'intérêt sont calculées par simulation de Monte-Carlo. Quelques applications ont déjà été traitées avec PyCATSHOO : [4] et [5]. Une base de connaissance a été implémentée avec cet outil pour générer les données.

Le système défini ici s'inspire d'un exemple modélisé proposé par les développeurs de PyCATSHOO : <http://www.pycatshoo.org/Model%20Samples.html>. Le code développé pour le système défini ici est une complexification du code de cet exemple développé par Hassane Chraïbi (EDF R&D).

C. Exemples de trajectoires simulées

La Figure 6 et la Figure 7 montrent l'évolution de la hauteur d'eau dans les réservoirs et de la température dans le réservoir 3 quand tous les composants (vannes et pompes) assurent leur fonction pendant 1000 unités de temps.

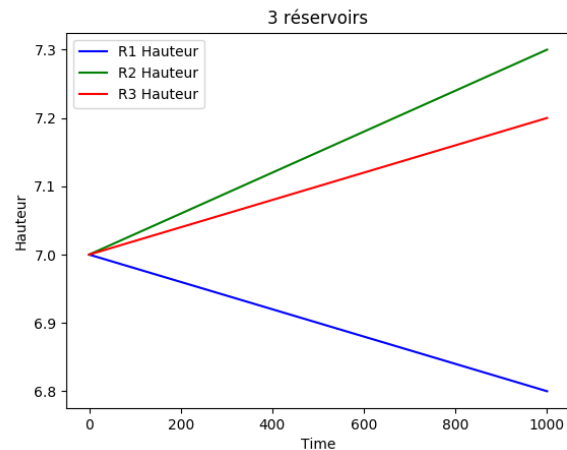


Figure 6 – Evolution des hauteurs d'eau dans les réservoirs quand aucune panne des composants ne survient

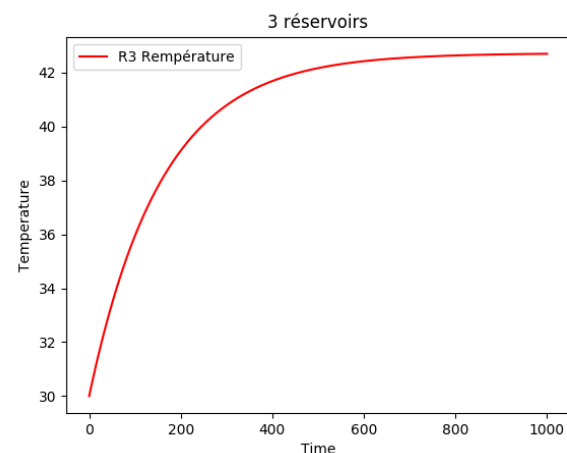


Figure 7 – Evolution de la température dans le réservoir 3 quand aucune panne des composants ne survient

La Figure 8 et la Figure 9 montrent comment se comportent ces indicateurs quand des pannes se produisent. Dans cet exemple :

- la vanne V2 se ferme et reste coincée à l'instant 303 ;
- la pompe P2 se ferme et reste coincée à l'instant 387.

Afin de rester dans des conditions de fonctionnement acceptables, les vannes V1 et V3 s'ouvrent et se ferment ce qui induit des variations observées dans les graphiques.

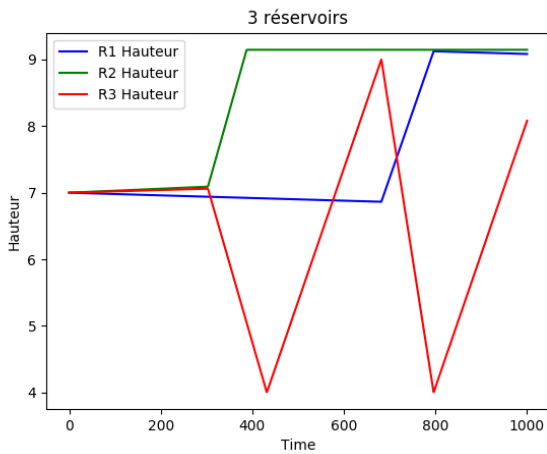


Figure 8 – Evolution des hauteurs d'eau dans les réservoirs quand des composants tombent en panne

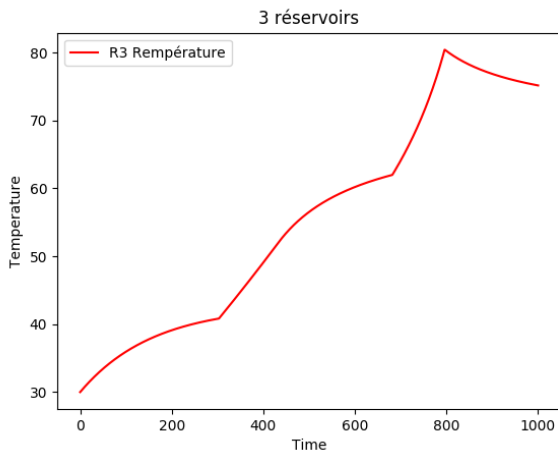


Figure 9 – Evolution de la température dans le réservoir 3 quand des composants tombent en panne

V. GENERATION DE DONNEES DE L'EXEMPLE 3-RESERVOIRS

Le code PyCATSHOO produit un ensemble de simulations de Monte-Carlo du système. Un horizon temporel maximal est défini manuellement pour l'ensemble des simulations. Une simulation se termine lorsque le système rencontre un des événements redoutés ou que cet horizon est dépassé. L'ensemble des résultats issus des simulations est initialement produit sous forme d'un fichier XML. Un script python a été développé pour extraire, à partir de ce fichier, trois bases de données distinctes au format CSV, de façon à ce que celles-ci soient facilement utilisables comme point d'entrée d'algorithmes d'apprentissage automatique.

La première base générée est celle représentant les données de capteurs. Elle contient le suivi au cours du temps de l'ensemble des valeurs des capteurs présents sur toute la durée des simulations effectuées. La RUL correspondant à chaque instant est automatiquement calculée pour les simulations au cours desquelles le système a atteint l'un des événements redoutés. Un exemple de telles données est représenté en Figure 10.

1	ID	R1.level	R2.level	R3.level	R3.temperature	TIME	RUL
1006	2	7	7	7	7 30.9261	0.0	638.889
1007	2	7	7	7	7 33.2035	1.001	637.888
1008	2	7	7	7	7 35.3181	2.002	636.8870000000001
1009	2	7	7	7	7 37.2815	3.003	634.885
1010	2	7	7	7	7 39.1046	4.004	634.885
1011	2	7	7	7	7 40.7972	5.00501	633.88399
1012	2	7	7	7	7 42.3689	6.00601	632.8829900000001
1013	2	7	7	7	7 43.8282	7.00701	631.88199
1014	2	7	7	7	7 45.1831	8.00801	630.88099
1015	2	7	7	7	7 46.4411	9.00901	629.87999
1016	2	7	7	7	7 47.6093	10.01	628.879

Figure 10 - Extrait d'un exemple de données de capteurs générées. (La colonne ID représente l'identifiant de la simulation)

La seconde base générée concerne les données de manœuvre. Elle représente l'ensemble des transitions d'état pour les pompes et les vannes déclenchées automatiquement au cours de la simulation, pour la régulation des niveaux d'eau et de température. Un exemple est donné en Figure 11.

1	ID	TIME	COMP	TRANSITIONS
2	2 626.376	V1	OPEN_to_CLOSE	
3	2 626.376	V2	OPEN_to_CLOSE	
4	2 638.388	P1	OPEN_to_CLOSE	
5	2 638.388	P2	OPEN_to_CLOSE	
6	3 250.188	V1	OPEN_to_CLOSE	
7	4 266.454	P2	OPEN_to_CLOSE	
8	5 238.614	V3	OPEN_to_CLOSE	
9	5 246.622	V1	OPEN_to_CLOSE	
10	5 246.622	V2	OPEN_to_CLOSE	
11	5 246.622	V3	CLOSE_to_OPEN	

Figure 11 - Extrait d'un exemple de données de manœuvre générées.

Enfin, la troisième base de données contient l'ensemble des défaillances observées sur les différents composants, incluant l'apparition des événements redoutés. Un exemple est donné en Figure 12.

1	ID	TIME	COMP	DEFAILLANCE
2	1 430.243	V2	OPEN_to_STUCKOPEN	
3	2 624.374	V3	OPEN_to_STUCKCLOSE	
4	2 638.889	R3	TOK_to_THEigh	
5	3 122.623	V2	OPEN_to_STUCKOPEN	
6	3 248.186	V3	OPEN_to_STUCKCLOSE	
7	3 258.196	R3	LTM_to_OVF	
8	4 8.25826	V3	OPEN_to_STUCKOPEN	
9	4 254.442	V2	OPEN_to_STUCKCLOSE	
10	4 266.454	R3	LNE_to_DO	
11	5 206.582	P2	OPEN_to_STUCKCLOSE	
12	5 259.384	R3	TOK_to_THEigh	
13	6 475.288	P1	OPEN_to_STUCKCLOSE	
14	6 528.091	R3	TOK_to_THEigh	

Figure 12 - Extrait d'un exemple de données de défaillances générées.

Les 3 événements redoutés sont identifiés par les labels suivants :

- TOK_to_THEigh : La température a atteint un niveau critique
- LNE_to_DO : Le réservoir est vide
- LTM_to_OVF : Le réservoir a débordé

Les trois bases de données ainsi générées permettront notamment le développement et la validation de modèles de diagnostic et de pronostic.

VI. EXPLOITATION DES DONNEES

Des premiers tests de pronostic ont été effectués. Ils ont pour objectif d'utiliser un algorithme de classification (ici XGBoost [6]) pour déterminer la RUL du système, c'est-à-dire le temps restant avant l'atteinte de l'un des trois

événements redoutés (débordement, assèchement, ou surchauffe). Plus précisément, nous cherchons à classer, à chaque pas de temps, si le système a « plus » ou « moins » de X unités de temps de vie restant. Un algorithme de régression a également été testé mais la prise en compte des trajectoires censurées (celles pour lesquelles la panne du système n'est pas observée) n'a pas été possible ce qui a provoqué de mauvais résultats.

Pour tester la classification par XGBoost, 1000 trajectoires ont été régénérées à l'aide de la base de connaissance développée avec PyCATSHOO. Ces 1000 trajectoires ont été séparées en deux : 70% pour l'apprentissage de l'algorithme de classification et 30% pour le test. Plusieurs valeurs de X ont été testées : 10, 50, 100 et 200. Pour chaque valeur, un algorithme XGBoost a été appris sur 70% des données et testé sur les 30% restantes. Une cross validation de type « 5-fold cross-validation » a été mise en place.

Chaque trajectoire a été simulée sur 1000 unités de temps. À chaque unité de temps, un indicateur qui servira de cible est créé de la façon suivante :

- Si le temps restant à vivre est supérieur à X, cet indicateur vaut « sup » ;
- Si le temps restant à vivre est inférieur à X, cet indicateur vaut « inf ».

Chaque trajectoire est donc composée de au plus 1000 observations (1 par unité de temps). Si la trajectoire mène à une défaillance, le nombre d'observations est égale aux nombre d'unités de temps de fonctionnement. Dans notre test, l'échantillon d'apprentissage (700 trajectoires) est donc constitué de au plus 647 031 observations. L'échantillon d'apprentissage est déséquilibré comme on peut l'observer dans le Tableau 1

Tableau 1 : Nombre d'observations pour lesquelles la fin de vie est atteinte avant ou après X unités de temps

X	Nombre de « inf »	Nombre de « sup »
10	2452	644579
50	11350	635681
100	22514	624517
200	44823	602208

Les tests ont été réalisés sur les 30% de données qui n'ont pas servi à l'apprentissage. Les résultats sont dans le Tableau 2. Les indicateurs de performance proposés sont :

- Précision : Nombre de défaillances prévues et qui se sont produites / Nombre de défaillances prévues.
- Rappel : Nombre de défaillances prévues et qui se sont produites / Nombre de défaillances qui se sont produites.
- Spécificité : Nombre de cas prévus sans défaillance et dont la défaillance ne s'est pas produite / Nombre de cas dont la défaillance ne s'est pas produite.
- Justesse : (Nombre de défaillances prévues et qui se sont produites + Nombre de cas prévus sans défaillance et dont la défaillance ne s'est pas produite) / Nombre de cas

L'efficacité de l'algorithme augmente à mesure que l'instant de la défaillance se rapproche mais reste très

performant même 200 unités de temps avant. Dans ce cas, 86% des pannes sont anticipées 200 unités de temps avant la panne et 78% des pannes annoncées dans moins de 200 unités de temps ont bien lieu.

Tableau 2 : Performance de l'algorithme XGBoost

X	Justesse	Précision	Rappel	Spécificité
10	0.99	0.96	0.97	0.99
50	0.99	0.93	0.93	0.99
100	0.99	0.87	0.91	0.99
200	0.97	0.78	0.86	0.98

Le graphique de la Figure 13 représente en ordonnée l'identifiant des trajectoires simulées pour lesquelles la défaillance du système est observée et en abscisse le temps restant à vivre. On remarque que 200 unités de temps avant la panne, la plupart des trajectoires passent du vert au rouge indiquant que XGBoost est performant dans ce cas.

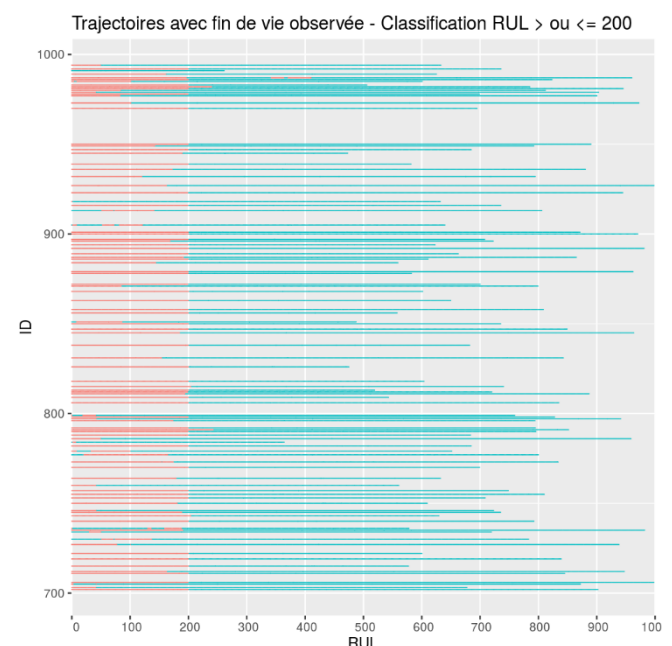


Figure 13 – Représentation de l'efficacité de XGBoost anticipant la défaillance 200 unités de temps à l'avance pour les trajectoires dont la panne est observée (rouge : XGBoost prévoit la défaillance dans moins de 200 unités de temps ; vert : XGBoost prévoit la défaillance dans plus de 200 unités de temps)

Le graphique de la Figure 14 représente en ordonnée l'identifiant des trajectoires simulées pour lesquelles la défaillance du système n'est pas observée et en abscisse le temps de fonctionnement. On remarque que le XGBoost prévoit assez rarement que la panne va survenir dans moins de 200 unités de temps. L'algorithme prévoit, pour quelques trajectoires, que la panne va survenir après l'horizon de simulation fixé à 1000 unités de temps ce qui n'a pas pu être vérifié.

Ces graphiques peuvent être affichés pour les différentes valeurs de X choisies pour l'étude. Pour X = 10, les prévisions, pour les trajectoires dont la défaillance du système est observée, sont quasi parfaites, voir la Figure 15.

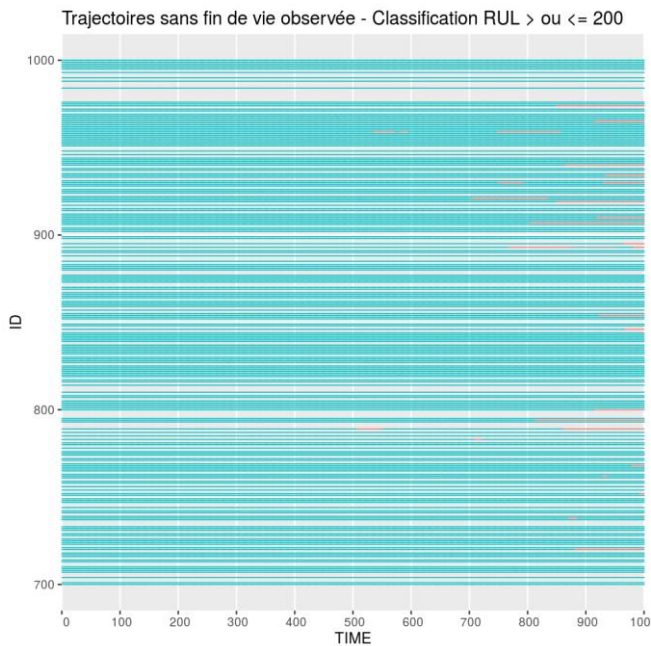


Figure 14 – Représentation de l'efficacité de XGBoost anticipant la défaillance 200 unités de temps à l'avance pour les trajectoires dont la panne n'est pas observée (rouge : XGBoost prévoit la défaillance dans moins de 200 unités de temps ; vert : XGBoost prévoit la défaillance dans plus de 200 unités de temps)

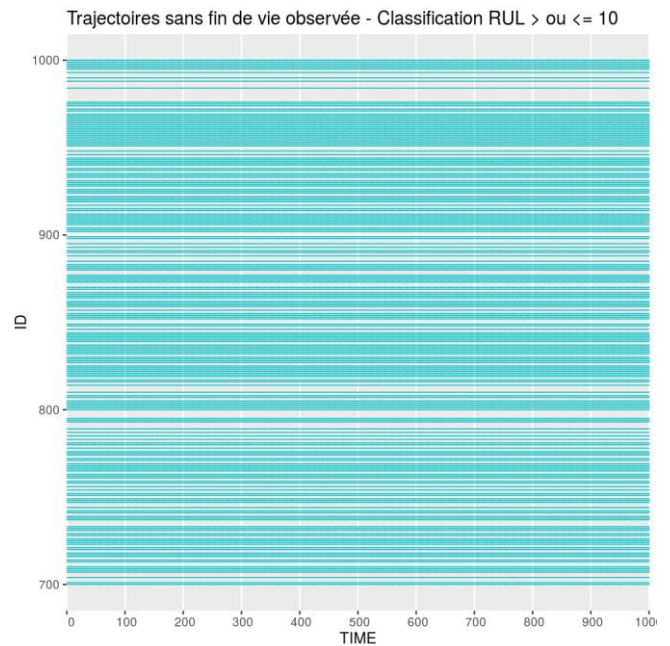


Figure 16 – Représentation de l'efficacité de XGBoost anticipant la défaillance 10 unités de temps à l'avance pour les trajectoires dont la panne n'est pas observée (rouge : XGBoost prévoit la défaillance dans moins de 10 unités de temps ; vert : XGBoost prévoit la défaillance dans plus de 10 unités de temps)

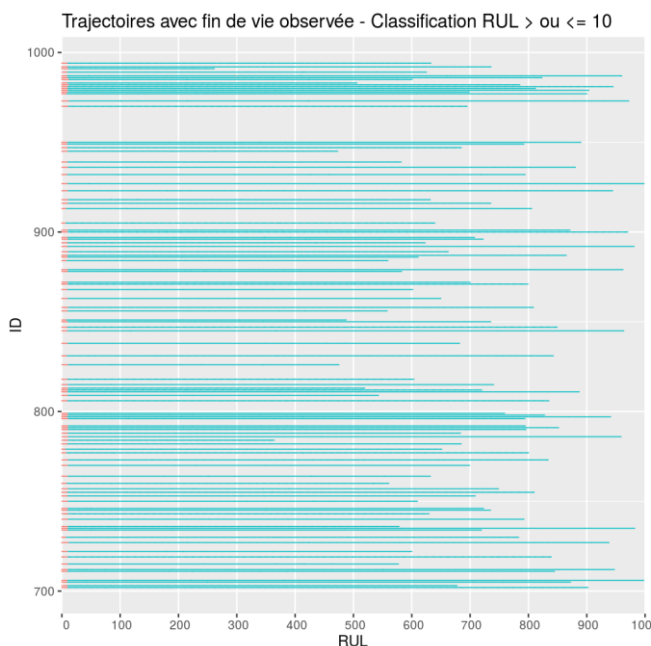


Figure 15 – Représentation de l'efficacité de XGBoost anticipant la défaillance 10 unités de temps à l'avance pour les trajectoires dont la panne est observée (rouge : XGBoost prévoit la défaillance dans moins de 10 unités de temps ; vert : XGBoost prévoit la défaillance dans plus de 10 unités de temps)

Pour les trajectoires dont la défaillance n'est pas observée, XGBoost ne semble pas se tromper, voir la Figure 16.

Ces tests ont montré que l'emploi d'un algorithme de classification XGBoost est performant pour estimer la RUL.

VII. CONCLUSION

Dans cette publication, la modélisation d'un système fictif nommé 3-Réservoirs a été présentée. Ce modèle a permis de mettre en place un processus de génération automatique de données de capteurs et de données opérationnelles dans un format adapté à l'apprentissage de modèles de d'apprentissage automatique. Les données simulées ont été testées sur une problématique de calcul de pronostic à l'aide d'un modèle de Gradient Boosting (XGBoost) ayant pour but de prédire à chaque instant l'imminence ou non de la fin de vie du système. Les premiers résultats ont montré de bonnes performances concernant des prévisions obtenues.

Une perspective possible pour la suite des travaux est la complexification du modèle des 3-Réservoirs, en rajoutant par exemple de nouveaux paramètres influant sur la dégradation des composants (par exemple un taux d'encrassement variable de l'eau), ou encore le rajout d'actions de maintenance correctives et préventives sur le système.

RÉFÉRENCES

- [1] Chraïbi H., Houdebine J. C., Sibling A., 2016, PyCATSHOO: Toward a new platform dedicated to dynamic reliability assessments of hybrid systems. PSAM, 2016.
- [2] Davis M.H.A., Piecewise Deterministic Markov Processes: A general class of non-diffusion stochastic models., Journal of the Royal Statistical Society. Series B (Methodological), 46(3), pp. 353-388, 1984.
- [3] Devooght, J., Dynamic Reliability. Springer, 1997.
- [4] Rychkov V., Chraïbi H., Dynamic probabilistic risk assessment at a design stage for a sodium fast reactor, ICFRRFC, 2017.
- [5] Rychkova I., Boissier F., Chraïbi H., Rychkov V., A pragmatic approach for measuring maintainability of DPRA models, ER, 2017.
- [6] Chen T. and Guestrin C., XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785-794. DOI:https://doi.org/10.1145/2939672.2939785, 2016.