



HAL
open science

Simuler la dégradation et ses impacts dans un jumeau numérique

Hequet Gaël, Nicolae Brinzei, Jean-François Petin, Sophie Viot Coster, Didier Paen, Olivier Bruneau

► **To cite this version:**

Hequet Gaël, Nicolae Brinzei, Jean-François Petin, Sophie Viot Coster, Didier Paen, et al.. Simuler la dégradation et ses impacts dans un jumeau numérique. Congrès Lambda Mu 24 “ Les métiers du risque : clés de la réindustrialisation et de la transition écologique ”, Institut pour la Maîtrise des Risques (IMdR), Oct 2024, Bourges, France. hal-04895268

HAL Id: hal-04895268

<https://hal.science/hal-04895268v1>

Submitted on 17 Jan 2025

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.



SIMULER LA DEGRADATION ET SES IMPACTS DANS UN JUMENTA NUMERIQUE

SIMULATE DEGRADATION AND ITS IMPACTS IN A DIGITAL TWIN

HEQUET Gaël
Université de Lorraine, CNRS, CRAN
F-54000 Nancy, France
gael.hequet@univ-lorraine.fr
VIOT COSTER Sophie EDF
R&D
91120 Palaiseau, France sophie.viot-
coster@edf.fr

BRINZEI Nicolae
Université de Lorraine, CNRS, CRAN
F-54000 Nancy, France
nicolae.brinzei@univ-lorraine.fr
PAEN Didier
CORYS
38024 Grenoble, France
didier.paen@corys.fr

PETIN Jean-François
Université de Lorraine, CNRS, CRAN
F-54000 Nancy, France jean-
francois.petin@univ-lorraine.fr
BRUNEAU Olivier
CORYS
38024 Grenoble, France
olivier.bruneau@corys.fr

Résumé — Le travail de recherche présenté dans cet article est une preuve de concept pour l'utilisation d'un moteur de calcul de dégradations et de défaillances dans le cadre du projet « Réacteur Numérique » au travers d'un démonstrateur développé pour le projet. Ce moteur utilise le concept des Automates Stochastiques Hybrides Basés sur des Profils pour modéliser, dans ce démonstrateur, un Relais de Commande à Main contrôlant l'ouverture d'une vanne ayant une incidence sur la pression au sein d'une cuve. Ce démonstrateur est constitué du moteur de calculs développé au sein du laboratoire CRAN en lien avec le projet « Réacteur Numérique » et le simulateur d'entraînement du « Réacteur Numérique » développé par CORYS. Ce simulateur d'entraînement est basé sur le simulateur d'EDF SIRENE en utilisant des modèles d'un réacteur de 1300MWe. Ces travaux sont effectués dans le cadre d'une thèse ayant pour objectif le développement d'un outil permettant la représentation des dégradations, de leurs impacts et des défaillances pour une intégration dans un jumeau numérique. Cet article présente une session d'entraînement dans laquelle le scénario suit un opérateur devant contrôler la pression au sein d'une cuve en utilisant un Relais de Commande à Main (RCM) pour envoyer des commandes d'ouverture à une vanne. Cette session est découpée en trois phases afin de présenter les deux modules majeurs du moteur de calcul de dégradation et de défaillances. Dans la première phase l'opérateur manipule un RCM, possédant une dégradation négligeable, grâce au module de simulation d'entraînement. Lors de la seconde phase l'opérateur lance le module de vieillissement pour faire vieillir le RCM pendant une durée prédéfinie. Enfin, dans la dernière phase, le module de simulation d'entraînement récupérera les données du module de vieillissement pour que l'opérateur puisse manipuler le RCM dégradé et observer un phénomène de dérive sur les commandes envoyées ; ces trois phases se passant dans la même session d'entraînement.

Mots-clefs — *sûreté nucléaire, dégradation, automates stochastiques hybrides basés sur des profils, réacteur numérique*

Abstract — The research work presented in this paper is a proof of concept for the use of a degradation and failure calculation engine as part of the “Digital Reactor” project, using a demonstrator developed for the project. This engine uses the concept of Profile-Based Stochastic Hybrid Automata to model, in this demonstrator, a Hand Control Relay (HCR) controlling the opening of a valve affecting the pressure within a tank. This demonstrator consists of the calculation engine developed in the CRAN laboratory in connection with the “Digital Reactor” project and the training simulator developed by CORYS. This training simulator is based on the EDF SIRENE simulator, using models of a 1300 MWe reactor. This work is being carried out as part of a PhD work aimed at developing a tool for representing degradations, failures and their impacts for integration into a digital twin. This paper presents a training session in which the scenario follows an operator who has to control the pressure in a tank by using a Hand Control Relay (HCR) to send opening commands to a valve. This session is divided into three phases in order to present the two major modules of the degradation and failure calculation engine. In the first phase, the operator manipulated an HCR with negligible degradation using the training simulation module. During the second phase, the operator launches the ageing module to age the HCR for a predefined period of time. Finally, in the last phase, the training simulation module recovers the data from the ageing module so that the operator can manipulate the degraded HCR and observe any drift in the commands sent. These three phases take place in the same training session.

Keywords — *nuclear safety, degradation, profile-based stochastic hybrid automata, digital reactor*

I. INTRODUCTION

Les systèmes complexes sont présents dans de nombreux domaines et, plus les technologies évoluent, plus la complexité des systèmes augmente. Ainsi, il devient de plus en plus difficile de prévoir le comportement de ces structures constituées de multiples

sous-systèmes interagissant entre eux. En effet, connaître l'impact d'une décision sur ce type de système est primordial, surtout dans le cadre de systèmes critiques.

On peut notamment citer l'industrie de production d'énergie électrique à base de l'énergie nucléaire qui, en France, garde une place importante, que ce soit dans l'opinion publique (Pratviel, 2022) mais aussi dans la production d'électricité française. Avec un parc de 56 réacteurs à eau pressurisée, une grande part de l'électricité produite en France provient du nucléaire (62.7% en 2022) (RTE, 2023).

C'est dans ce contexte qu'un projet innovant a vu le jour, le projet Réacteur Numérique (Levesque & Gouyon, 2021). Ce projet, dirigé par EDF, rassemble 9 partenaires français afin de mettre en commun leur savoir-faire dans un objectif commun (EDF, 2022). Cet objectif est la création du jumeau numérique de chaque réacteur nucléaire permettant la modélisation et la représentation de phénomènes complexes, de la conception à son exploitation, en incluant les phases de maintenance. Ce jumeau numérique a pour ambition de devenir un outil à la fois de suivi, d'entraînement mais aussi d'anticipation, c'est-à-dire d'être capable d'anticiper l'évolution du réacteur sur une période de l'ordre de plusieurs années et se doit d'être représentatif du réacteur piloté au quotidien par l'opérateur. De plus, celui-ci doit être accessible au travers d'un portail web et donc utilisable sans avoir besoin d'utiliser une machine dédiée.

Pour ce faire, divers groupes de travail se sont constitués, rassemblés en lots afin de débloquer divers verrous en lien avec les couplages de codes, la modélisation des différents phénomènes ainsi que les aspects de visualisation, de formation, d'entraînement ainsi que d'accessibilité des outils.

Au sein de ce projet, un travail de recherche est mené, au travers d'une thèse au laboratoire CRAN. Ces travaux ayant pour objectif d'intégrer une modélisation dysfonctionnelle prenant en compte la dégradation comme le vieillissement ou l'usure et les défaillances de systèmes. Ceci, dans un contexte de cosimulation et dans lequel des sollicitations pourront être injectées. Il faut donc des outils capables de représenter des phénomènes continus ainsi que des événements discrets tout en étant capables de modifier ces paramètres internes en cours de simulations ainsi que d'exploiter des sollicitations externes.

Pour répondre à ces problématiques et besoins, les Automates Stochastiques Hybrides Basés sur des Profils (ASHBP) (Hequet et al., 2021) ont été choisis. Ces outils, issus d'une extension des Automates Stochastiques Hybrides (ASH) (Casteñeda et al., 2011 ; Chiacchio et al., 2016), permettent ainsi une modélisation hybride des phénomènes physiques continus et des événements discrets générés par le contrôle-commande ou représentant l'occurrence des défaillances. Avec ces propriétés, il est possible de simuler la vie de systèmes sur une certaine période de temps pouvant être de l'ordre de plusieurs années en temps accéléré (simulation de Monte-Carlo) afin de faire évoluer les paramètres internes des modèles. Il devient alors possible de récupérer les résultats de ces simulations pour les utiliser comme conditions initiales dans un simulateur d'entraînement ; simulateur dans lequel le temps s'écoulerait à la même vitesse que dans le monde réel (temps réel) ou bien en mode pas à pas. Des modifications aux ASHBP avaient déjà été apportées (Hequet et al., 2022) afin de leur permettre de considérer l'injection de commandes externes et de modifications de paramètres internes, pouvant permettre la cosimulation ; l'intérêt étant de permettre à un acteur externe (opérateur humain ou un autre logiciel) d'injecter des commandes dans le simulateur et donc interagir avec les composants modélisés par des ASHBP. Ceci doit permettre d'observer l'impact de l'état du système sur son comportement et sur les actions de cet acteur.

Dans cet article, nous présenterons le moteur de calcul développé dans le cadre des travaux de thèse, en communication avec un simulateur développé par CORYS, ainsi que deux modules développés dans ce moteur. Un premier module permet de modéliser le comportement d'un composant selon son état de dégradation dans une simulation en temps réel, modélisé à l'aide PyCATSHOO (Desgeorges et al., 2021). Le second module permet de simuler l'évolution des grandeurs internes de ce composant sur une période donnée et de retourner les données pouvant être exploitées ensuite par le premier module en temps réel. L'objectif est d'observer la différence de comportement du composant modélisé selon sa dégradation, notamment en termes de dérive. Ainsi, nous modéliserons la dérive d'un composant et l'impact de cette dérive au travers de la commande envoyée, de l'action réalisée puis au travers de l'évolution d'une grandeur physique (la pression) en lien avec ces actions.

Les résultats présentés dans cet article fournissent aussi une preuve de concept pour l'utilisation des ASHBP pour modéliser la dégradation et ses impacts dans un simulateur en Temps Réel pour le projet « Réacteur Numérique ».

II. MODELISATION

Au sein de ce projet, notre mission est de développer un outil capable de représenter la dégradation (le vieillissement ou l'usure) ainsi que son impact sur le comportement fonctionnel des composants.

A. La modélisation de la dégradation Avant de présenter l'outil développé pour répondre à ces besoins, il a fallu étudier les différents modèles de dégradation existants.

Avec le développement de systèmes critiques de plus en plus complexes, les outils traditionnels de la sûreté de fonctionnement connaissent parfois quelques limites. Pour ces systèmes critiques, afin de pouvoir prévoir leur comportement, il faut pouvoir améliorer la précision des modèles mais aussi être capable de prendre en compte de plus en plus de paramètres (Zio, 2016). Ces modélisations deviennent alors possibles grâce à l'émergence de nouvelles technologies plus performantes. Ainsi il devient possible d'étudier de nouveaux phénomènes tel que la dégradation. Il est couramment admis que plus un système se dégrade et plus celui-ci tendra vers sa défaillance (Ye & Xie, 2015).

Il existe plusieurs modèles de dégradations construits de différentes manières. On trouve par exemple les modèles statistiques comme les distributions de Weibull (Lu & Meeker, 1993) ou de Bernstein (Gebrael et al., 2009) issues d'études statistiques. Des modèles stochastiques comme le processus Gamma (Abdel-Hameed, 2013 ; Lawless & Crowder, 2004) ou la loi

inversegaussienne (Chen et al., 2015 ; Ye & Chen, 2014), utilisant des modèles d'évolution de la dégradation par des incréments aléatoires. On peut encore citer des modèles de dégradation basés sur la physique (physique des défaillances) pour représenter les mécanismes physiques, au travers d'équations continues, engendrant la dégradation (Chookah et al., 2011) ou les modèles à multi-états par modèles semi-markoviens (Black et al., 2005), représentant les différents états de dégradation. Enfin, d'autres approches tentent de modéliser la dégradation en utilisant des données de capteurs en temps réel sur des grandeurs internes mais aussi externes au système (Bian et al., 2015).

Dans nos travaux, nous devons modéliser ces dégradations mais aussi leurs impacts sur les performances du système représenté. En effet, la dégradation peut entraîner différentes conséquences sur un système. Les dégradations peuvent entraîner des défaillances franches (Ye & Xie, 2015) mais peuvent aussi altérer le comportement du système pouvant modifier ses performances (Kurz et al., 2009 ; Aghaei et al., 2022).

B. Contexte des jumeaux numériques

En plus de penser aux différents phénomènes à représenter, il faut aussi penser à l'intégration. En effet, l'objectif étant d'intégrer les travaux au sein d'un environnement de simulation d'un jumeau numérique, il faut comprendre la nature de ce concept ainsi que ses objectifs.

Selon les objectifs du jumeau numérique conçu, les définitions peuvent varier (Broo & Schooling, 2021 ; Glaessgen & Stargel, 2012 ; Kritzinger et al., 2018 ; Van Der Horn, 2021). Cependant, la définition communément admise est celle d'un objet numérique pouvant recevoir et envoyer des données à un objet physique de manière continue et automatique (Kritzinger et al., 2018) (figure 1).

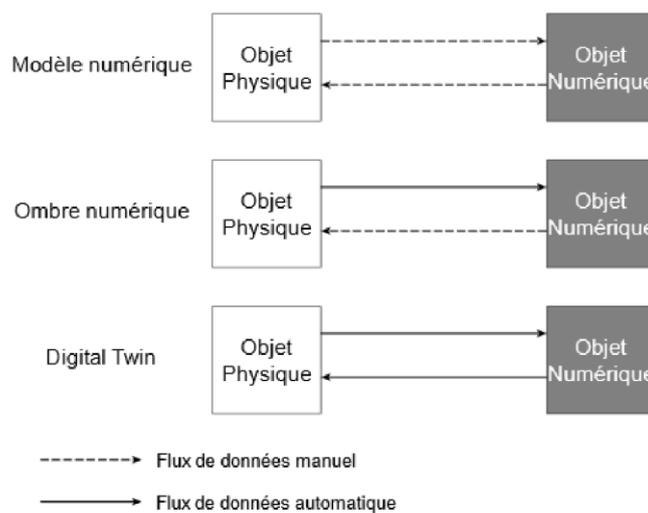


Fig. 1. Types de communication entre un objet physique et un objet numérique

Il faut donc penser les outils comme devant être capables de gérer des requêtes automatiques pouvant être nombreuses, et entraînant des problématiques en termes de temps de calcul et de ressources de calculs. De plus, ce projet rassemblant plusieurs partenaires ayant chacun leurs domaines d'expertise, leurs savoir-faire mais aussi leurs propres codes de calculs, de nouvelles problématiques liées aux couplages de ces différents codes doivent aussi être pris en compte (EDF, 2023).

Les travaux présentés dans ce papier se concentrent majoritairement sur la mission d'entraînement du « Réacteur Numérique ». Ainsi le moteur de calcul développé ne sera pas en communication automatique avec le système réel mais devra être capable de communiquer avec un simulateur d'entraînement qui pourra fournir le temps de simulation en temps réel ainsi que les différentes sollicitations pour les composants.

Dans les travaux précédents (Hequet, 2021 ; Hequet, 2022), il a été choisi d'utiliser les Automates Stochastiques Hybrides (ASH) (Casteñeda et al., 2011 ; Chiacchio et al., 2016) et de les modifier pour y ajouter une composante d'évolution de dégradation et de comportement selon des profils d'utilisations (Hequet et al., 2021). Ceci nous a amené à étendre les ASH et définir les Automates Stochastiques Hybrides Basés sur des Profils (ASHBP). Enfin, une seconde modification, ajoutant des fonctions appelables en interne, rend les ASHBP réactifs à certains types de commandes, donnant l'automate suivant (figure 2) :

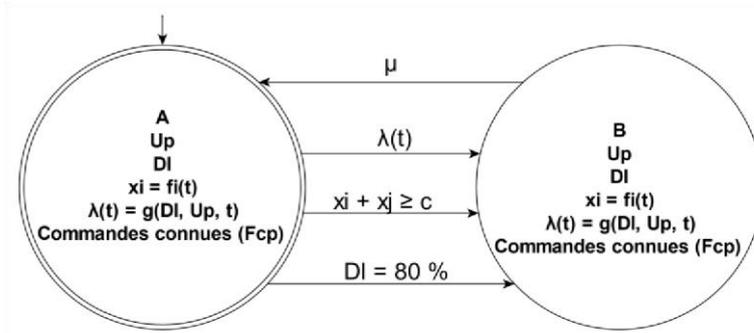


Fig. 2. Exemple d'ASHBP interactif avec 2 états et 4 transitions avec pour état initial, l'état « A »

Il est possible de décrire formellement cet automate (de la Fig. 2.) par un tuple à 18 composantes :

$$ASHBP_{interactif} = (\chi, E, Ar, X, A, H, F, p, U_p, D_l, D_{lf}, f_\lambda, f_{cp}, \chi_0, x_0, p_0, U_{p0}, D_{l0}) \quad (1)$$

La signification des variables d'un ASHBP décrit par l'équation (1) est la suivante : χ - ensemble des états discrets ; E - ensemble des événements ; Ar - ensemble des arcs ; X - ensemble des variables continues ; A - une fonction "activité" associant des variables à des état (association d'un élément de X à un élément de χ) ; H - ensemble des horloges ; F - application reliant chaque horloge à une fonction de distribution ; p - matrice de distribution de probabilité des transitions d'état ; U_p - ensemble + des profils d'utilisation du système ; D_l - niveau de dégradation ; D_{lf} - ensemble de fonctions discontinues influencées par U_p + et impactant D_l . f_λ représente un ensemble de fonctions liées au taux de défaillance le faisant varier en fonction du temps, des variables et des conditions. f_{cp} , représente un ensemble fini de fonctions pouvant être appelées et qui peuvent représenter, par exemple, des sollicitations propres au composant modélisé ou des changements de profils d'utilisation. Enfin, $\chi_0, x_0, p_0, U_{p0}, D_{l0}$ représentent les conditions initiales de l'automate.

Dans ces automates, l'ensemble D_{lf} permet de lier le profil d'utilisation avec le niveau de dégradation. Cet ensemble est constitué de fonctions discontinues, chacune reliées à un profil d'utilisation différent. Ainsi cet ensemble a pour dimension le nombre de profils d'utilisations (U_p) définis pour le système modélisé. Pour un profil dont le nom est par exemple "nominal", la valeur de D_{lf} sera alors :

$$D_{lf}(U_p) = \begin{cases} D_{TauxUp(nominal)} & \text{si } U_p = \text{"nominal"} \\ 0 & \text{sinon} \end{cases} \quad (2)$$

Le résultat de cette fonction sera alors utilisé pour définir un nouveau niveau de dégradation. Avec, par exemple ce modèle utilisé dans (Hequet et al., 2022) :

$$D_l(t_n) = (t_n - t_{n-1}) * D_{lf} + D_l(t_{n-1}) \quad (3)$$

avec D_l étant le niveau de dégradation tel que $\{D_l \in \mathbb{R}^+ \mid 0 \leq D_l \leq 100\}$, t_n l'instant courant et t_{n-1} l'instant précédent. Enfin, avec ce niveau de dégradation, nous estimons un taux de défaillance propre à la dégradation avec, par exemple, le modèle suivant :

D

$$\lambda_{degradation} = e^{(10l-10)} \quad (4)$$

Avec ce modèle, nous représentons l'impact grandissant de la dégradation sur le taux de défaillance du système modélisé. Enfin, ce taux de défaillance de dégradation est ajouté au taux de défaillance propre au profil d'utilisation, donnant un taux de défaillance « global » du système modélisé :

$$\lambda_{global} = \lambda_{profil} + \lambda_{degradation} \quad (5)$$

Ainsi, l'évolution de la dégradation du système modélisé a un impact sur son taux de défaillance. Cet outil de modélisation est la base des travaux qui seront présentés dans cette communication.

III. L'INTEGRATION DANS LE JUMEAU NUMERIQUE

Afin de représenter l'impact de la dégradation et l'intégrer dans l'environnement du simulateur d'entraînement du « Réacteur Numérique », d'autres modifications ont dû être apportées et sont présentées ici. Nos travaux se concentrant sur l'environnement d'entraînement, les composants modélisés devront répondre à des sollicitations afin de remplir leurs missions. C'est lors de ces sollicitations que les effets des dégradations pourront être observés. La dégradation dans les versions précédentes des ASHBP impactait uniquement le taux de défaillance, le faisant varier en utilisant un taux de défaillance propre à la dégradation. Cependant, la dégradation ayant aussi un impact sur les performances des systèmes, nous avons décidé de la représenter au travers de dérives sur le résultat de sollicitations en utilisant un scénario vraisemblable, s'inspirant de retours d'expériences d'opérateurs.

A. Le fonctionnement du moteur de calcul

Afin de permettre aux utilisateurs de s'entraîner sur divers scénarios mais aussi observer l'impact de la dégradation, il faut pouvoir modifier les données initiales lors de la simulation des composants. Il peut aussi être intéressant de voir directement l'impact du vieillissement sur les composants. Dans cette optique, nous avons développé un moteur de calcul permettant de faire fonctionner en parallèle deux modules de calculs. Le premier module, le module de simulation d'entraînement, est utilisé pour la modélisation des composants en « temps réel », ce qui signifie que ce module sera en communication avec les composants du simulateur d'entraînement du « Réacteur Numérique ». Le second module est le module de vieillissement : il permet de récupérer l'état du système à l'instant courant et l'utiliser comme condition initiale dans le module d'entraînement. Ensuite, ce second module simulera le vieillissement du système sur une durée donnée en entrée. Les résultats de cette

simulation sont enregistrés dans un fichier csv qui peut ensuite être chargé dans le module de simulation d'entraînement au travers d'une fonction dans l'interface. Enfin, un dernier module, le module d'interface graphique a été développé pour faciliter les échanges humain/machine. L'architecture du système est présentée en Fig. 3.

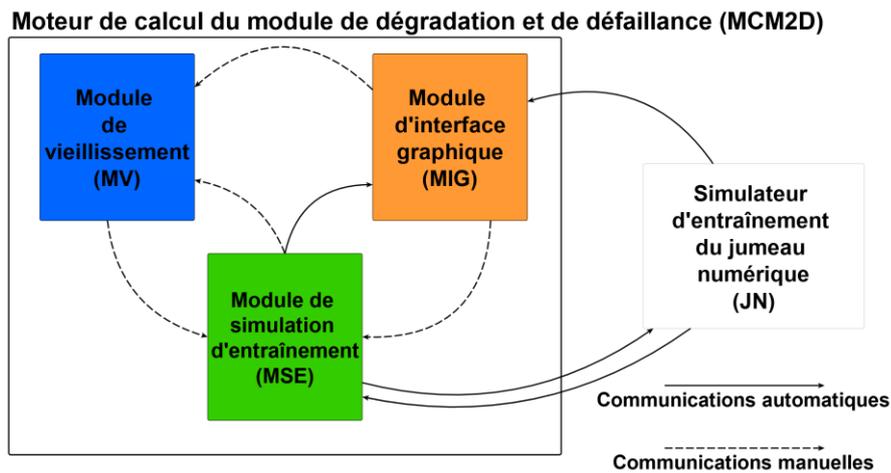


Fig. 3. Schéma de l'architecture de la communication entre le Moteur de Calcul du Module de Dégradation et de Défaillance (MCM2D) avec le Module de Vieillessement (MV), le Module de Simulation d'Entraînement (MSE), le Module d'Interface Graphique (MIG) et le simulateur d'entraînement du jumeau numérique (JN).

Les données échangées entre les différents modules sont présentées dans le tableau 1.

TABLE I. DONNEES ECHANGEES SUR LES CONNEXIONS ENTRE LES MODULES.

Connexion	Données	Connexion	Données
Simulateur d'entraînement du JN / MIG	<ul style="list-style-type: none"> - Temps de simulation - Pression - Commande Réelle du RCM - Ouverture Réelle de la vanne 	MIG / MSE	<ul style="list-style-type: none"> - Commande de lancement du module - Sollicitation du RCM - Paramétrage du RCM - Chargement des données du MV pour le MSE - Arrêt du module
Simulateur d'entraînement du JN / MSE	<ul style="list-style-type: none"> - Temps de simulation 	MIG / MV	<ul style="list-style-type: none"> - Lancement du module - Arrêt du module
MSE / Simulateur d'entraînement du JN	<ul style="list-style-type: none"> - Commande Réelle du RCM 	MSE / MV	<ul style="list-style-type: none"> - Envoi des conditions initiales de la simulation de vieillissement
MSE / MIG	<ul style="list-style-type: none"> - Commande Théorique du RCM - Ouverture théorique de la vanne 	MV / MSE	<ul style="list-style-type: none"> - Chargement des résultats de vieillissement dans le RCM

Ainsi, comme on peut voir en Fig. 4., au démarrage du moteur de calcul, il est possible de paramétrer le mode de fonctionnement entre un mode hors connexion et un mode de connexion au simulateur du jumeau numérique. Ici, nous nous intéresserons uniquement au mode avec connexion au simulateur. Une fois le mode choisi, le moteur de calcul utilise le protocole MQTT (HiveMQ, 2020), qui est un protocole permettant la communication dans un réseau informatique par un système d'abonnement et de publication, pour se connecter au simulateur d'entraînement du jumeau numérique. Si la connexion est réussie, alors le moteur sera paramétré pour un fonctionnement en mode « Jumeau Numérique », c'est-à-dire en connexion avec le simulateur du jumeau numérique. Il va lancer alors un processus propre à ce mode de fonctionnement présenté en Fig. 5.

Une fois la connexion effectuée, le moteur lance le module d'interface graphique puis le module de simulation d'entraînement pour les connecter à leur tour au simulateur d'entraînement du jumeau numérique. Le module d'interface graphique permet de récupérer les données utiles pour observer l'impact des choix de l'opérateur humain et de lancer les modules du Moteur de Calcul du Module de Dégradation et de Défaillance (MCM2D). Cette interface permet d'effectuer la communication entre un utilisateur humain et le moteur de calcul (et ses modules) afin d'effectuer par exemple des sollicitations de composant, des modifications de paramètres ou le lancement du module de vieillissement. Enfin, si la commande reçue est différente de la commande « EXIT », alors le moteur lancera cette commande et reviendra dans un état d'attente de nouvelle commande.

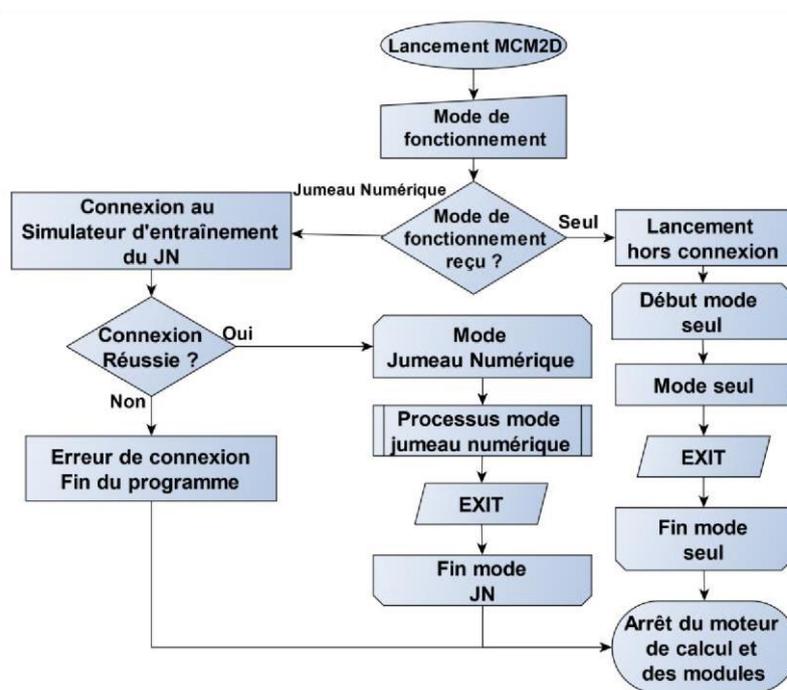


Fig. 4. Logigramme du fonctionnement du moteur de calcul de dégradation et de défaillances

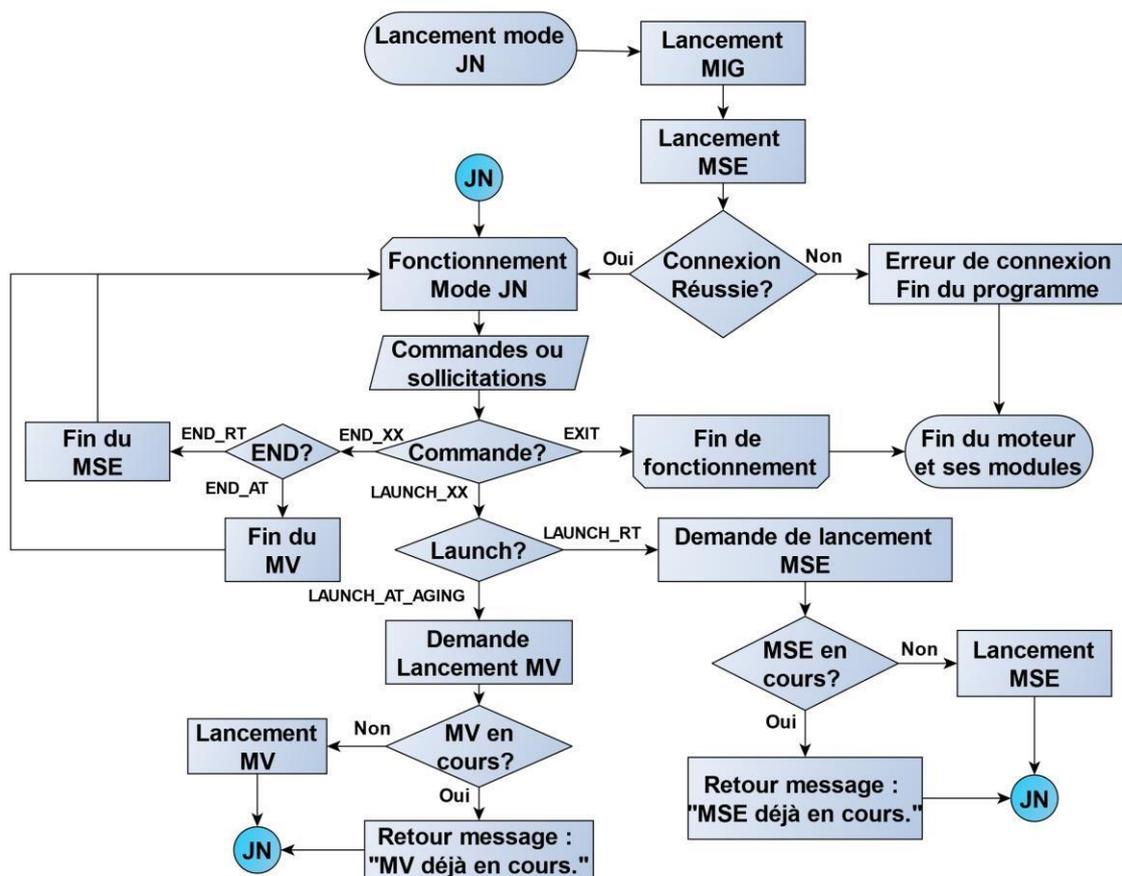


Fig. 5. Logigramme du fonctionnement du MCM2D en mode « Jumeau Numérique ».

B. Représenter l'impact de la dégradation

L'objectif de ces travaux étant de représenter l'impact de la dégradation sur un système, il a fallu étendre le concept des ASHBP interactifs présentés dans la section précédente et construits à partir de l'équation (1). Tout d'abord, une partie stochastique pour la dégradation a été ajoutée. Il existe donc, comme pour les ASHBP initiaux, une dégradation monotone mais à celle-ci s'ajoute une valeur de dégradation aléatoire, présentée en (6). Avec D_{sf} un facteur d'amplitude des dégradations

aléatoires. Cette partie aléatoire peut être positive ou négative, avec X une variable aléatoire suivant une loi uniforme comprise entre -1 et 1, $U(-1,1)$, faisant augmenter ou diminuer la dégradation.

$$D_l(t_n) = (t_n - t_{n-1}) * (D_{lf} + X * D_{lf} * D_{sf}) + D_l(t_{n-1}) \quad (6)$$

Ensuite, le concept de dérive a été intégré au travers d'un coefficient affectant le retour d'une fonction de sollicitation. Par exemple, dans le cadre d'un capteur, avec la dégradation, une dérive dans les valeurs retournées par le capteur peut être observée. Pour représenter ce phénomène, cinq grandeurs supplémentaires au constructeur des ASHBP sont intégrées, donnant un 23tuple comme décrit par l'équation (7).

$$ASHBP_{interactif} = (\chi, E, Ar, X, A, H, F, p, U_p, D_l, D_{lf}, D_{sf}, f_\lambda, f_{CP}, C_d, D_{id}, D_{dr}, \chi_0, \alpha_0, p_0, U_{p0}, D_{l0}, C_{d0}) \quad (7)$$

C_d est un coefficient de dérive représentant la dérive relative à une performance ou une sollicitation, C_{d0} étant sa valeur initiale. Ce coefficient peut soit être défini de manière manuelle par un formateur, mais peut aussi évoluer avec le temps, plus ou moins rapidement selon la dégradation :

$$C_d(t_n) = C_d(t_{n-1}) + D_{dr} \quad (8)$$

Dans ce but, D_{id} , est une grandeur, constante, définie au début de la simulation permettant de représenter le taux de l'impact de la dégradation sur la dérive. Pour modéliser cet impact, nous utilisons D_{dr} , un taux de dérive qui fait augmenter le coefficient de dérive au cours du temps. Ce taux de dérive augmente en utilisant la constante représentant l'impact de la dégradation sur la dérive donnant le modèle suivant (9).

$$D_{dr} = D_l(t_n) * D_{id} \quad (9)$$

Avec ces modifications, il sera possible d'utiliser les ASHBP interactifs pour représenter l'impact de la dégradation sur les performances des composants modélisés. Dans la suite, le terme ASHBP assimilera la dernière version des ASHBP_{interactif}. IV. EXPERIENCE

A. Plan d'expérience

Afin de présenter nos travaux au sein du projet « Réacteur Numérique », un démonstrateur a été développé. Pour construire ce démonstrateur, un scénario issu de données d'expérience a été déterminé. Ce papier présentera une partie des fonctionnalités du démonstrateur présenté en tant que preuve de faisabilité de l'utilisation de ce moteur de calcul pour remplir les missions demandées par le projet « Réacteur Numérique ». 1) Contexte

Le scénario retenu pour le démonstrateur est celui d'un opérateur devant gérer manuellement la pression dans une cuve (Fig. 6). La valeur de cette pression est donnée par un capteur et doit être gardée entre 26,5 et 27 bars et est influencée par la valeur d'ouverture d'une vanne. L'opérateur peut jouer sur cette valeur d'ouverture au travers de commandes envoyées par un Relais de Commande à Main (RCM). Le RCM enverra alors une commande d'ouverture à la vanne dont l'ouverture influencera la pression du système. Enfin, l'opérateur devra utiliser cette valeur de pression pour conduire la vanne avec le RCM.

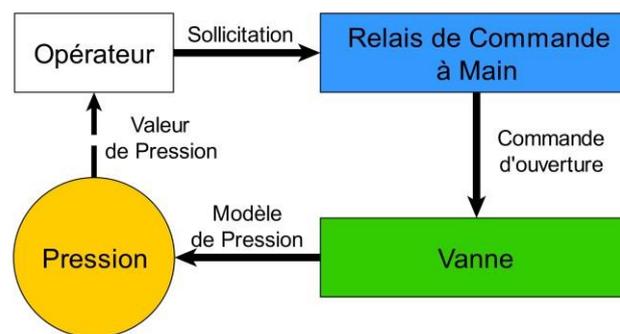


Fig. 6. Schéma des interactions du système entre les éléments pour la conduite d'une vanne.

Ce RCM peut être sujet à de la dégradation, ce qui va influencer, par dérive, la valeur de la commande envoyée à la vanne. En effet, plus le RCM sera dégradé, plus une dérive importante pourra être observée sur les commandes envoyées par le RCM. L'occurrence de défaillances ne sera, ici, pas prise en compte pour montrer uniquement l'impact de la dégradation.

Au cours du scénario trois phases seront présentées. Une première phase sera la conduite avec un RCM ayant une dégradation négligeable et donc n'ayant pas de dérive significative. La seconde phase sera l'utilisation du Module de Vieillesse (MV) qui permettra de faire évoluer la dégradation du composant en le vieillissant virtuellement de 15 ans. Durant cette phase, nous ne considérerons aucune action de maintenance ni de recalibrage de l'appareil. Enfin, dans une troisième et dernière phase, l'opérateur effectuera la même mission que dans la première phase mais avec le RCM vieilli de 15 ans. Ces trois phases auront lieu dans la même session d'entraînement.

2) Modèles

Le système considéré est constitué d'un RCM commandant une vanne. L'ouverture de celle-ci influence la pression au sein d'une cuve.

a) Module de simulation d'entraînement

Dans cette modélisation pour le module de simulation d'entraînement, seules les dégradations et leurs impacts du RCM sont modélisées au sein du Moteur de Calcul du Module de dégradation et de défaillance à l'aide d'un ASHBP comme présenté en figure 7. Les dégradations de la vanne ne sont pas représentées. Les aspects fonctionnels du RCM, de la vanne ainsi que la pression sont modélisées au sein du simulateur d'entraînement du jumeau numérique et sont donc considérés comme des modèles boîte noire.

Le module de simulation d'entraînement du moteur récupère la valeur d'horloge du simulateur d'entraînement afin de se synchroniser et de modéliser le RCM. Le module d'interface graphique permet de récupérer les données issues du module de simulation d'entraînement et d'afficher les données à l'utilisateur.

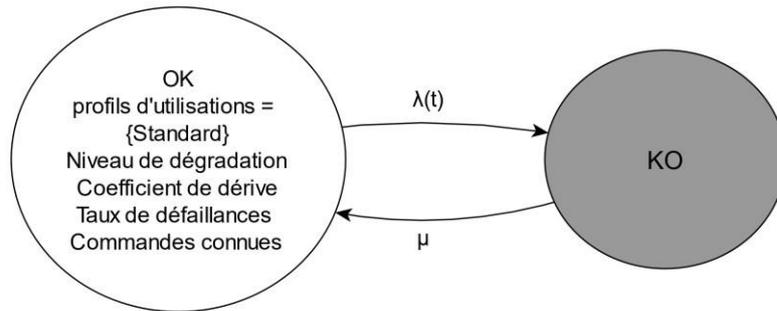


Fig. 7. Automate Stochastique Hybride Basé sur des Profils interactif représentant un RCM avec un seul profil d'utilisation et l'ajout d'un coefficient de dérive applicable au résultat de la sollicitation.

Lorsqu'il reçoit une sollicitation, l'ASHBP du RCM calcule une commande réelle et une commande théorique. La commande théorique est la commande parfaite, sans dérive. La commande réelle applique la valeur de la dérive selon la dégradation du composant. L'ASHBP utilise le principe de dégradation et de dérive présenté dans la section précédente et modélisées par les équations (7), (8) et (9). Pour cela, lorsqu'une commande est entrée par l'opérateur au travers du RCM, celle-ci est comparée à l'ouverture réelle de la vanne pour en déduire la différence (diff) en (10). À cette différence est appliquée le coefficient C_d , représentant la dérive appliquée par cette nouvelle commande en (11). La dérive est donc appliquée sur la différence entre la nouvelle commande théorique, celle sans aucune dérive, et l'ancienne valeur d'ouverture, donnant la commande réelle.

$$diff = commande_{cible} - ouverture_{réelle} \quad (10) \quad commande_{réelle} = ouverture_{réelle} + diff * C_d \quad (11)$$

Pour la commande théorique, le modèle reste le même avec le calcul de la différence comme dans (10) mais C_d n'est pas appliqué pour la commande (12) :

$$commande_{théorique} = ouverture_{réelle} + diff \quad (12)$$

Ces commandes sont ensuite envoyées au modèle de RCM du simulateur d'entraînement du jumeau numérique qui enverra sa commande à la vanne. L'opérateur peut ainsi observer deux interfaces, celle du simulateur d'entraînement du jumeau numérique et celle du module d'interface graphique qui seront présentées dans la section « Résultats » de cet article. b) *Module de vieillissement*

Le module de vieillissement permet d'effectuer la seconde phase du scénario. Dans cette phase, l'utilisateur fait appel au MV en donnant en entrée la durée du vieillissement. Ensuite, le MV récupère les dernières valeurs du MSE sur l'état du RCM. Les données du RCM récupérées sont sa durée de fonctionnement depuis sa mise en marche, son taux de dégradation, son niveau de dégradation (D_i), son taux d'impact de la dégradation sur la dérive (D_{id}), son taux de dérive (D_{dr}) et son coefficient de dérive (C_d). Une fois les données récupérées, le module va réutiliser le modèle d'ASHBP du MSE (fig. 7) et l'intégrer à une simulation de Monte-Carlo qui va lancer 100 séquences pour obtenir des données moyennes sur la durée de vieillissement donnée par l'utilisateur en entrée.

Enfin, ces données de vieillissement sont sauvegardées dans un fichier csv. Si l'utilisateur le souhaite, les valeurs moyennes du dernier instant simulé par le MV peuvent être chargées dans l'ASHBP du MSE. Ainsi, l'utilisateur pourra interagir avec le composant RCM vieilli. 3) *Valeurs initiales*

Dans cette section sont présentées les conditions initiales des valeurs du MCM2D au lancement de la session d'entraînement. Dans le cadre d'utilisation de variables aléatoires, le générateur utilisé est yam5 (Livinskii et al., 2021). Le simulateur d'entraînement du jumeau numérique est considéré comme une boîte noire, les valeurs de pression y sont calculées et mises à jour. Les valeurs d'ouverture et de commandes sont forcées par le MCM2D. Les valeurs d'évolution de la dégradation sont

basées sur une durée de 40 ans pour atteindre que le niveau de dégradation atteigne 100%. Les valeurs ici sont théoriques et ne sont pas des données issues de systèmes réels.

TABLE II. VALEURS INITIALES DE LA SESSIONS D'ENTRAÎNEMENT.

Variable	Valeur	Variable	Valeur
Graine pour le générateur de nombres aléatoire utilisant le générateur yarn 5	55	C_d	1
T_life	0.0 (secondes)	Pression	26.887 (bars)
Niveau de dégradation (D_i)	0.0 (%)	Ouverture réelle	0 (%)
Taux de dégradation (D_{taux})	$2.85e^{-4}$ (%/h)	Ouverture théorique	0 (%)
Facteur d'amplitude des dégradations aléatoires (D_{sf})	0.1	Commande réelle	0 (%)
Impact de la dégradation sur la dérive (D_{id})	$5e^{-5}$	Commande théorique	0 (%)
Taux de dérive (D_{dr})	$D_i * D_{id}$	Echelle temporelle du simulateur d'entraînement	0.1 (seconde)

B. *Expérience et discussions* Lors d'une session d'entraînement, l'opérateur peut observer les interfaces du MCM2D et du simulateur d'entraînement du jumeau numérique. Afin d'envoyer des sollicitations, l'opérateur devra utiliser l'interface du MIG (Fig. 8.), choisir la fonction « SOLICITATION », entrer la valeur de la nouvelle commande pour le RCM et cliquer sur le bouton « Send » afin d'envoyer la commande au MCM2D qui exploitera la commande et enverra le résultat de la sollicitation selon l'état du RCM au simulateur d'entraînement du jumeau numérique qui affichera la commande du RCM et l'ouverture de la vanne dans le champ « Pos_dem » (Fig. 9.).

L'opérateur peut observer l'évolution sur des fenêtres de temps de maximum 200 secondes. Afin de respecter un temps de réponse inférieur à 100ms entre chaque pas de temps et éviter une désynchronisation du MIG, le rafraîchissement des courbes se fait manuellement grâce au bouton « Refresh ». Le bouton « Save » permet de sauvegarder les données du MIG et du MSE dans un fichier csv. Enfin, le bouton « Quit » permet d'arrêter le MCM2D et ses modules.

Les résultats de la session d'entraînement sont affichés sur les figures 10 à 13. Les données en lien avec la dégradation sont affichées avec une échelle temporelle de l'ordre de l'année afin d'observer les changements issus du MV.

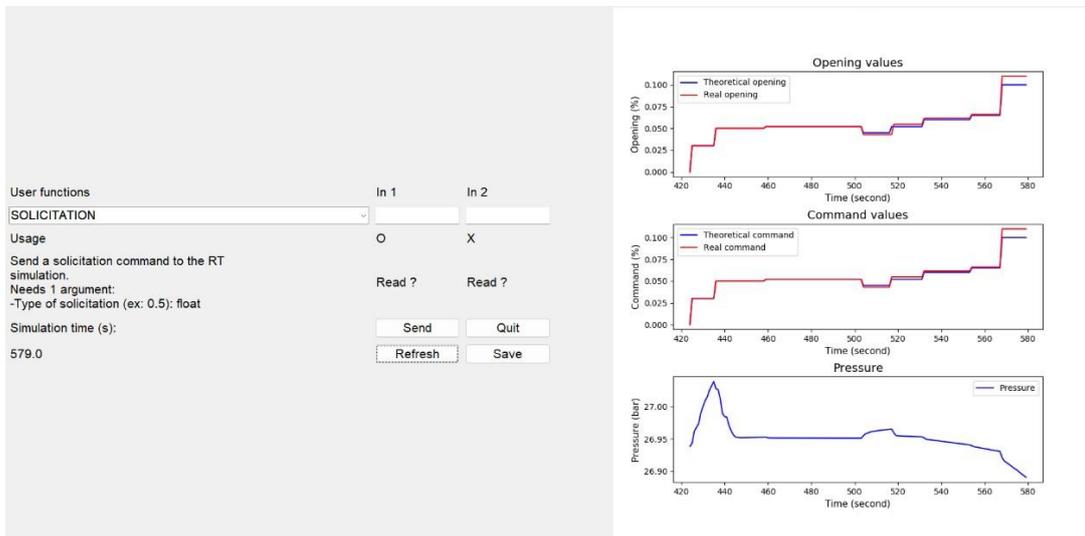


Fig.

8. Interface du MIG avec à gauche la zone pour l'envoi de commande et à droite la zone d'affichage des données.

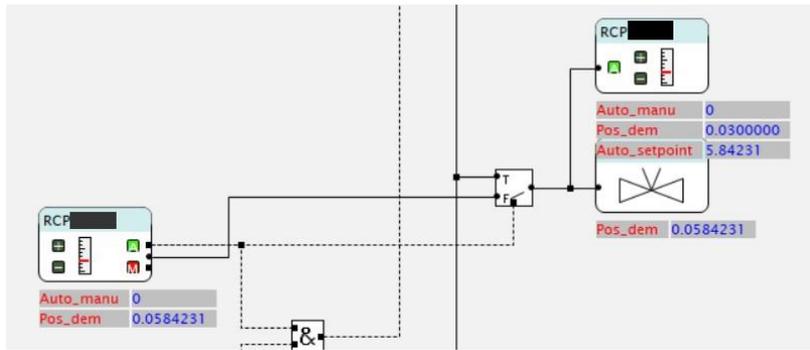


Fig. 9. Interface du MIG avec à gauche la zone pour l'envoi de commande et à droite la zone d'affichage des données.

Les données de commande, d'ouverture et de pression sont présentées sur une échelle de temps de l'ordre de la seconde car ce sont les grandeurs affichées et utilisées par l'opérateur pour conduire la vanne avec le RCM en cours de session.

À la fin de la session les données de simulation peuvent être récupérées dans un fichier csv en rassemblant les données des trois phases. Durant la première phase, celle de simulation en temps réel, la dégradation ne varie que très peu car le temps évolue avec un pas de temps de 100 ms.

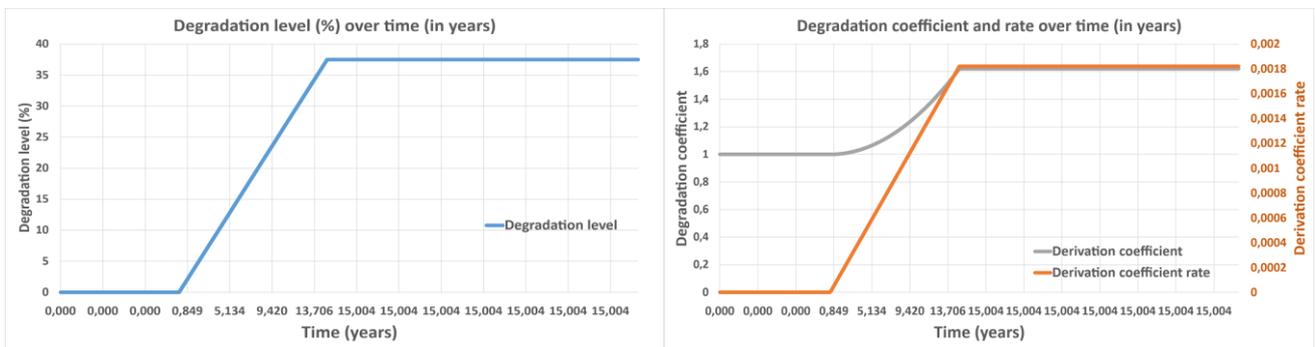


Fig. 10. À gauche : Niveau de dégradation du RCM (%) en fonction du temps (en années).

À droite : Coefficient de dégradation (en gris, axe vertical à gauche) et taux de dérive (en orange, axe vertical à droite) en fonction du temps (en années).

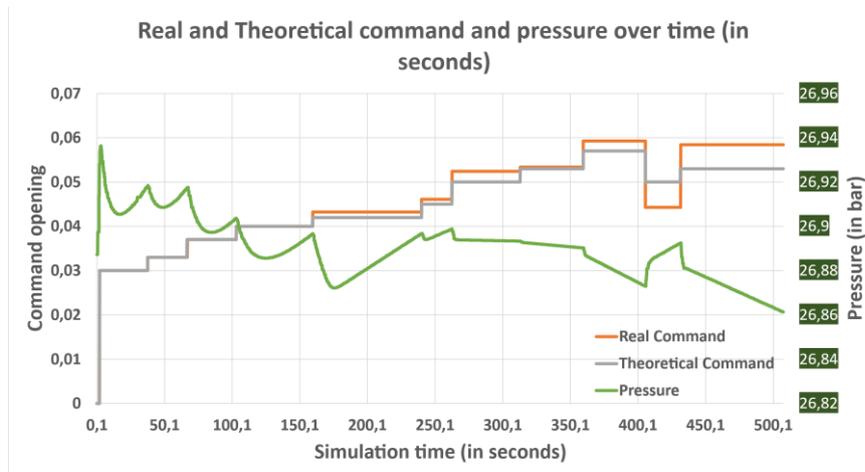


Fig. 11. Commandes réelles (en orange) et théoriques (en gris) du RCM (% d'ouverture) sur l'axe de gauche ainsi que la pression (bar, en vert) sur l'axe de droite en fonction du temps (en secondes).

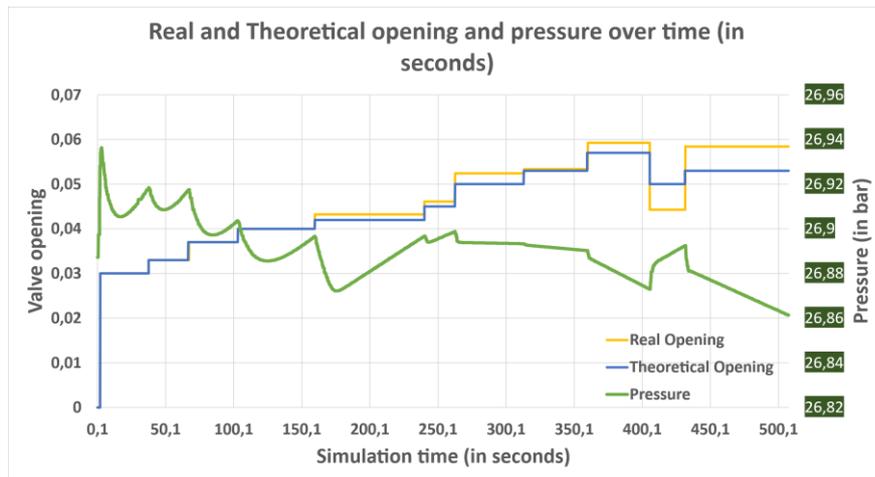


Fig. 12. Ouvertures réelles (en jaune) et théoriques (en bleu) de la vanne (% d'ouverture) ainsi que la pression (bar, en vert) sur l'axe de droite en fonction temps (en secondes).

Durant la seconde phase, lancée à un temps de simulation de 120 secondes, le système est modélisé avec le module de vieillissement, un formateur a fait vieillir le RCM sur 15 ans, ce qui a augmenté le niveau de dégradation à environ 37% (Fig. 10.). Ce composant vieilli est réutilisé par l'opérateur dans la troisième phase. L'impact de cette dégradation s'observe sur la dérive appliquée aux commandes et donc ensuite à l'ouverture de la vanne (Fig. 12-13). La première commande après le lancement du MV est envoyée vers 155 secondes environ et une différence entre les valeurs de la commande théorique et celle réelle est observable. La valeur de la dérive peut être observée sur la courbe en gris (Fig. 11) où l'évolution du coefficient de dérive est affichée. A l'issue des 15 ans de vieillissement, le coefficient de dérive peut être observé sur la Fig. 11 sur la courbe grise et est d'environ 1,6. Ceci signifie que les commandes envoyées par le RCM subiront une dérive de l'ordre de 60% sur la différence entre l'ancienne commande réelle et la nouvelle théorique. Ainsi, avec le temps, plus les performances sont dégradées, plus les opérateurs sont impactés pour piloter et comprendre le phénomène pour trouver des solutions.

L'évolution du coefficient de dérive visible sur la courbe grise en Fig. 11 devient exponentielle au fur et à mesure que la dégradation augmente. Cela s'explique avec l'augmentation du taux de dérive avec la dégradation. Plus le taux est grand, plus ce coefficient augmentera rapidement afin de représenter le phénomène que plus un équipement est dégradé plus ses performances seront dégradées, et donc plus il faudrait effectuer d'opérations de maintenance sur celui-ci pour diminuer cet impact.

Cette dérive peut être minimisée sur de petites commandes. Cependant, lorsque la situation exige une commande plus grande, comme celle observée à environ 400 secondes (Fig. 12-13), l'impact de la dérive devient plus visible et peut entraîner des changements importants du comportement du système comme il est possible d'observer dans les figures précédentes (les sauts de la valeur de pression).

Dans cette session d'entraînement, les valeurs d'ouverture de la vanne sont très faibles. Les effets de la dérive sont donc minimisés. Cependant, même si la dérive est faible, plus il y a de commandes et donc d'occurrence de dérives, plus l'impact sera visible. Ce phénomène doit donc pouvoir être pris en compte lors de l'entraînement des opérateurs afin de pouvoir reconnaître ce type de phénomène et trouver une solution si un tel scénario arrivait sur le système réel.

V. CONCLUSION

Au travers de cet exemple de session d'entraînement, il est montré que les outils présentés dans de précédents travaux (Hequet et al., 2022) ont pu être intégrés dans le jumeau numérique du « Réacteur Numérique ». Cette intégration s'est effectuée au travers du développement d'un moteur de calcul communiquant avec le simulateur d'entraînement du jumeau numérique, développé par CORYS. Dans ce moteur, plusieurs modules peuvent être lancés permettant une simulation d'entraînement mais aussi un vieillissement des composants modélisés. Ce vieillissement faisant augmenter la dégradation, considérée aléatoire dans ce papier, aura un impact sur la performance du composant modélisé au travers d'un coefficient de dérive, variable selon le niveau de dégradation.

Ainsi, il sera possible pour un opérateur de s'entraîner sur un système à différentes périodes de son exploitation pour observer de scénarios possibles (déterministes ou stochastiques) et ainsi entraîner sa capacité à détecter l'origine de problèmes pour en déduire les opérations à effectuer.

Cependant, il reste encore des problématiques à résoudre pour le futur. En effet, la construction d'un modèle de composant nécessite une bonne connaissance de celui-ci et donc une étude approfondie des retours d'expériences. Il faut ensuite étudier le processus pouvant causer les défaillances ou dysfonctionnements pour pouvoir les représenter et estimer les évolutions futures avec des modèles permettant de tenir la synchronisation avec les logiciels en communication. En effet, pour un simulateur d'entraînement, le pas de temps donné ici étant de 100 ms, il faut que les calculs soient terminés dans ce laps de temps. Il faut donc trouver des optimisations et des algorithmes pour que la synchronisation temporelle ne soit pas perdue.

Enfin, dans un contexte d'entraînement, il serait intéressant de penser à une interface séparée entre un formateur construisant un scénario et un opérateur faisant tourner le scénario. Selon une utilisation en tant que formateur ou qu'opérateur, les fonctions nécessaires seront différentes. Un formateur souhaitera être capable de paramétrer les différentes grandeurs des composants (temps de vie, niveau de dégradation, taux de défaillance, probabilités d'événements, ...) là où un opérateur souhaitera surveiller certaines grandeurs et effectuer des sollicitations sur des composants. Actuellement il n'existe qu'une seule interface graphique rassemblant les deux statuts.

En perspectives, des optimisations techniques pourront être apportées sur le moteur de calcul et les différents modules. De plus, pour une connexion de ce moteur à un jumeau numérique, en communication automatique et continue avec un système réel, il serait intéressant de pouvoir déterminer un niveau de dégradation d'un composant selon les retours du jumeau. En utilisant des données de capteurs pour évaluer des baisses de performances ou d'autres phénomènes physiques, il serait peut-être possible d'estimer le niveau de dégradation des composants et utiliser ces estimations afin de les utiliser dans un module d'entraînement ou de pronostics. Ceci pouvant être utilisé pour de l'aide à la décision pour l'exploitant pour la conduite de la tranche et pour la maintenance.

REMERCIEMENTS

Ces travaux s'inscrivent dans le projet « Réacteur Numérique » no. 2019-PSPC-15, financé par la BPIFRANCE, dans le cadre de l'appel à « Projets Structurants pour la Compétitivité » (PSPC) du Programme d'Investissements d'Avenir (PIA).

REFERENCES

- Abdel-Hameed, M. (2013). Gamma degradation process. *Levy Processes and Their Applications In Reliability and Storage*, Springer, 2013.
- Aghaei, M., Fairbrother, A., Gok, A., Ahmad, S., Kazim, S., Lobato, K., Oreski, G., Reinders, A., Schmitz, J., Theelen, M., Yilmaz, P., & Kettle, J. (2022). Review of degradation and failure phenomena in photovoltaic modules. *Renewable and Sustainable Energy Reviews*, 159, 112160. <https://doi.org/10.1016/j.rser.2022.112160>
- Bian, L., Gebraeel, N., & Kharoufeh, J. P. (2015). Degradation modeling for real-time estimation of residual lifetimes in dynamic environments. *IEEE Transactions*, 47(5), 471-486. <https://doi.org/10.1080/0740817X.2014.955153>
- Black, M., Brint, A. T., & Brailsford, J. R. (2005). A semi-Markov approach for modelling asset deterioration. *Journal of the Operational Research Society*, 56(11), 1241-1249. <https://doi.org/10.1057/palgrave.jors.2601967>
- Broo, D. G., & Schooling, J. (2021). Digital twins in infrastructure : Definitions, current practices, challenges and strategies. *International Journal of Construction Management*, 0(0), 1-10. <https://doi.org/10.1080/15623599.2021.1966980>
- Castañeda, G. A. P., Aubry, J.-F., & Brinzei, N. (2011). Stochastic hybrid automata model for dynamic reliability assessment. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 225(1), 28-41. <https://doi.org/10.1177/1748006XJRR312>
- Chen, N., Ye, Z.-S., Xiang, Y., & Zhang, L. (2015). Condition-based maintenance using the inverse Gaussian degradation model. *European Journal of Operational Research*, 243(1), 190-199. <https://doi.org/10.1016/j.ejor.2014.11.029>
- Chiacchio, F., D'Urso, D., Manno, G., & Compagno, L. (2016). Stochastic hybrid automaton model of a multi-state system with aging : Reliability assessment and design consequences. *Reliability Engineering & System Safety*, 149, 1-13. <https://doi.org/10.1016/j.ress.2015.12.007>
- Chookah, M., Nuh, M., & Modarres, M. (2011). A probabilistic physics-of-failure model for prognostic health management of structures subject to pitting and corrosion-fatigue. *Reliability Engineering & System Safety*, 96(12), 1601-1610. <https://doi.org/10.1016/j.ress.2011.07.007>
- Desgeorges, L., Piriou, P.-Y., Lematre, T., & Chraïbi, H. (2021). Formalism and semantics of PyCATSHOO : A simulator of distributed stochastic hybrid automata. *Reliability Engineering & System Safety*, 208, 107384. <https://doi.org/10.1016/j.ress.2020.107384>
- EDF. (2022, février 9). *R&D : Le numérique pour transformer le nucléaire* | EDF FR. <https://www.edf.fr/groupe-edf/inventer-l-avenir-de-l-energie/r-d-unsavoir-faire-mondial/pepites-r-d/reacteur-numerique/ambition-du-projet>
- EDF. (2023, juin 15). *R&D : Le Réacteur Numérique, une vision collective* | EDF FR. <https://www.edf.fr/groupe-edf/inventer-l-avenir-de-l-energie/r-d-unsavoir-faire-mondial/pepites-r-d/reacteur-numerique/projet-r-d>
- Gebraeel, N., Elwany, A., & Pan, J. (2009). Residual Life Predictions in the Absence of Prior Degradation Knowledge. *IEEE Transactions on Reliability*, 58(1), 106-117. <https://doi.org/10.1109/TR.2008.2011659>
- Glaessgen, E., & Stargel, D. (2012, juin 14). *The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles* / Structures, Structural Dynamics, and Materials and Co-located Conferences. <https://arc.aiaa.org/doi/abs/10.2514/6.2012-1818>
- Hequet, G., Brinzei, N., & Pétrin, J.-F. (2021). Usage profile in physical systems modeled with stochastic hybrid automata. *International Conference on Information and Digital Technologies, IDT 2021*, 220-229. <https://doi.org/10.1109/IDT52577.2021.9497617>
- Hequet, G., Brinzei, N., & Pétrin, J.-F. (2022, octobre 10). *Automates stochastiques hybrides basés sur des profils pour la modélisation de défaillances dans un système complexe*. 23e Congrès de Maîtrise des Risques et de Sécurité de Fonctionnement $\lambda\mu 23$. <https://hal.science/hal-03878447>
- HiveMQ. (2020). *HiveMQ – MQTT & MQTT 5 Essentials : A comprehensive overview of MQTT facts and features for beginners and experts alike*. <https://www.hivemq.com/>
- Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital Twin in manufacturing : A categorical literature review and classification. *IFAC PapersOnLine*, 51(11), 1016-1022. <https://doi.org/10.1016/j.ifacol.2018.08.474>

- Kurz, R., Brun, K., & Wollie, M. (2009). Degradation Effects on Industrial Gas Turbines. *Journal of Engineering for Gas Turbines and Power*, 131(062401). <https://doi.org/10.1115/1.3097135>
- Lawless, J., & Crowder, M. (2004). Covariates and random effects in a gamma process model with application to degradation and failure. *Lifetime Data Analysis*, 10(3), 213-227. <https://doi.org/10.1023/b:lida.0000036389.14073.dd>
- Levesque, B., & Gouyon, D. (2021, mars). Projet “ Réacteur Numérique ” : Vers un Jumeau Numérique de Réacteur Nucléaire. *S-mart2021 : 17ème colloque national S-mart*. <https://hal.archives-ouvertes.fr/hal-03225080>
- Livinskii, V., Babokin, D., & Regehr, J. (2021). *Intel/yarpgen* [C++]. Intel Corporation. <https://github.com/intel/yarpgen> (Edition originale 2016)
- Lu, C. J., & Meeker, W. Q. (1993). Using Degradation Measures to Estimate a Time-to-Failure Distribution. *Technometrics*, 35(2), 161-174. <https://doi.org/10.2307/1269661>
- Pratviel, E. (2022). *Les Français et le nucléaire : Adhésion et traits d’image*. <https://www.ifop.com/publication/les-francais-et-le-nucleaire-adhesion-et-traitsdimage/>
- RTE, rte. (2022). *Bilan électrique 2022 : Production d’électricité en France | RTE*. <https://analysesetdonnees.rte-france.com/bilan-electriqueproduction#Nucleaire>
- VanDerHorn, E., & Mahadevan, S. (2021). Digital Twin : Generalization, characterization and implementation. *Decision Support Systems*, 145, 113524. <https://doi.org/10.1016/j.dss.2021.113524>
- Ye, Z.-S., & Chen, N. (2014). The Inverse Gaussian Process as a Degradation Model. *Technometrics*, 56(3), 302-311. <https://doi.org/10.1080/00401706.2013.830074>
- Ye, Z.-S., & Xie, M. (2015). Stochastic modelling and analysis of degradation for highly reliable products. *Applied Stochastic Models in Business and Industry*, 31(1), 16-32. <https://doi.org/10.1002/asmb.2063>
- Zio, E. (2016). Some Challenges and Opportunities in Reliability Engineering. *IEEE Transactions on Reliability*, 65(4), 1769-1782. <https://doi.org/10.1109/TR.2016.2591504>