

Amélioration de l'outil DyRelA pour une évaluation de la Sûreté de fonctionnement des systèmes dynamiques

Gabriel Antonio Pérez Castañeda

Département et Laboratoire de Métal-Mécanique
Instituto Tecnológico de Tehuacán
Tehuacán, Puebla, Mexique
perezc76@ensem.inpl-nancy.fr, gapercas@yahoo.com

Nicolae Brinzei et Jean-François Aubry

Centre de Recherche en Automatique de Nancy – CNRS
UMR 7039, Nancy – Université, INPL
Vandœuvre-lès-Nancy, France
{Nicolae.Brinzei, Jean-Francois.Aubry}@ensem.inpl-nancy.fr

Résumé—La recherche de solutions analytiques pour l'évaluation de la fiabilité en contexte dynamique n'est pas résolue dans le cas général. Des approches partielles existent dans le cas d'hypothèses particulières. La simulation de Monte Carlo serait le seul recours, s'il existait des outils performants permettant la simulation simultanée de l'évolution événementielle discrète du système et de son évolution continue, tout en prenant en compte les aspects probabilistes. Dans ce contexte, nous avons d'abord, introduit le concept formel d'automate stochastique hybride (ASH) afin de prendre en compte tous les problèmes posés par la fiabilité dynamique. En plus, nous l'avons implémenté dans l'environnement Scicos de Scilab. Finalement, nous avons défini l'outil informatique DyRelA avec le but d'accéder à l'évaluation des grandeurs de la sûreté de fonctionnement (SdF) par la simulation de Monte Carlo d'un système modélisé par l'ASH. Le but de cet article est de présenter une version améliorée de DyRelA qui permet de réduire significativement la durée de simulation menant à l'évaluation de la SdF d'un système en contexte dynamique.

Abstract—The research of analytical solutions for reliability assessment in dynamic context is not solved in the general case. Partial approaches exist in the case of particular hypothesis. The Monte Carlo simulation would be the only recourse, but there were no efficient tools allowing the simultaneous simulation of the discrete event evolution of the system and its continuous evolution, taking into account the probabilistic aspects. In this context, we have formally introduced the concept of stochastic hybrid automaton (SHA) with the goal of consider all the problems defining the concept of the dynamic reliability. The SHA has been implemented in the Scicos toolbox of the Scilab environment. Finally, we have defined the DyRelA tool with the aim to access to the assessment of the dependability parameters by a Monte Carlo simulation of a system modeled by a SHA. The goal of this article is to present an improved version of the DyRelA tool which will significantly reduce the simulation duration of the dependability assessment of a system in dynamic context.

Mots clés : *Fiabilité Dynamique, Sûreté de Fonctionnement, Automate Stochastique Hybride, Simulation de Monte Carlo.*

Keywords: *Dynamic reliability, dependability assessment, hybrid stochastic automaton, Monte Carlo simulation*

I. INTRODUCTION

La fiabilité dynamique de systèmes hybrides met en évidence la présence de deux constituants de nature différente : un état continu du système dépendant des variables physiques continues (fonctionnelles ou dysfonctionnelles) décrivant la dynamique et d'un état discret dépendant des états discrets de ses constituants. La fiabilité dynamique accentue aussi le fait que tels systèmes évoluent dans le temps et que les défaillances (et réparations) peuvent influencer les dynamiques et, réciproquement, que les dynamiques (et les variables d'état associées) peuvent affecter les taux de défaillance et de réparation.

La complexité mathématique de l'évaluation analytique de la sûreté de fonctionnement d'un système dynamique nous amène à recourir à la simulation. Pour cela, il nous faut disposer d'un modèle adapté de la fonction de structure du système. Nous considérons que la fonction de structure d'un système en contexte dynamique est un Automate Stochastique Hybride (ASH). C'est un automate parce qu'il est composé d'un ensemble d'états discrets. Il est hybride parce que chaque état discret est caractérisé par un système d'équations continues et par un sous-ensemble de transitions de sortie définies par des seuils sur ces variables continues. Il est stochastique parce que chaque état discret est caractérisé par un ensemble de variables aléatoires et un sous-ensemble de transitions de sortie définies par des seuils sur les variables aléatoires. L'automate nous permet d'accéder aux grandeurs de la Sûreté de Fonctionnement (SdF), lesquelles sont obtenues par statistique sur un grand nombre de simulations (méthode de Monte Carlo). L'intérêt du formalisme est bien entendu vis-à-vis de l'impossibilité à trouver une solution analytique pour les grandeurs de la SdF dans de nombreux cas. L'ASH a été défini formellement et présenté dans [1], [2]. Nous avons implémenté l'ASH dans l'environnement Scicos de Scilab. En conséquence, nous avons développé l'outil DyRelA (Dynamic Reliability and Assessment) [3]. Cet outil est capable d'effectuer, par une simulation de Monte Carlo, l'évaluation des paramètres de la sûreté de fonctionnement de systèmes dans le contexte dynamique. Le système à étudier est modélisé par l'ASH en prenant en compte tous les aspects de la fiabilité

dynamique. La simulation de Monte Carlo est effectuée par Scilab en mode Batch.

Le but de cet article est de présenter une version améliorée de DyRelA. Cette amélioration a pour objectif de réduire la durée de simulation de l'évaluation de la sûreté de fonctionnement des systèmes en contexte dynamique. Pour cela, nous présentons, d'abord, la définition formelle de l'ASH. Deuxièmement, nous présentons des approches que nous avons développées antérieurement pour accéder à cette évaluation. Troisièmement, nous présentons les aspects que nous avons traités pour améliorer l'outil DyRelA. Finalement, nous présentons l'application de DyRelA à un cas test. Les résultats obtenus de l'évaluation des paramètres de la SdF à travers de DyRelA ont été comparés avec les approches markovienne et semi-markovienne. Des conclusions et des perspectives sont présentées.

II. AUTOMATE STOCHASTIQUE HYBRIDE

Nous avons formellement défini l'automate stochastique hybride (ASH) sur la base de l'automate hybride proposé par [4] et [5].

Un automate stochastique hybride est défini comme un 11-tuple :

$$ASH = (\mathcal{X}, \mathcal{E}, \mathcal{A}, X, A, \mathcal{H}, \mathcal{F}, p, x_0, x_0', p_0) \quad (1)$$

dans lequel :

- \mathcal{X} est un ensemble fini d'états discrets,
- \mathcal{E} est un ensemble fini d'événements,
- \mathcal{A} est un ensemble fini d'arcs de la forme (x_e, G, R, x') où :
- x et x' sont les états origine et but de l'arc, e l'événement associé à l'arc, G la condition de garde et R est la fonction de réinitialisation. Sur occurrence de e et si la condition de garde G est vérifiée, le système bascule de l'état x à l'état x' dans lequel R définit les valeurs initiales des variables continues du système,
- X est un ensemble fini des variables réelles,
 $A : \mathcal{X} \times \mathcal{X} \rightarrow (\mathfrak{R}^+ \rightarrow \mathfrak{R})$ est une fonction des « activités », qui associe à un élément de $\mathcal{X} \times \mathcal{X}$ une fonction définie sur \mathfrak{R}^+ et à valeur dans \mathfrak{R} ,
- \mathcal{H} est un ensemble fini d'horloges,
- $\mathcal{F} : \mathcal{H} \rightarrow (\mathfrak{R} \rightarrow [0,1])$ est une application qui associe à chaque horloge une fonction de répartition de probabilité,
- p est une distribution de probabilités de transition d'état $p(x' | x_e)$. Par exemple, si on a le même événement e définissant les transitions de l'état discret x vers les états discrets x' et x'' (l'automate à états sous jacent n'est pas déterministe), on peut définir la probabilité p de passer de l'état x à l'état x' et la probabilité $(1-p)$ de passer de l'état x à l'état x'' ,
- x_0 , x_0' et p_0 correspondent respectivement à l'état discret initial, à la valeur initiale du vecteur d'état continu et à la distribution initiale des probabilités de transition.

Les éléments \mathcal{X} , \mathcal{E} et \mathcal{A} de l'automate stochastique hybride correspondent à l'automate à états finis définissant sa partie événementielle. En revanche, X et A définissent sa partie continue. Finalement, \mathcal{H} et p expriment son aspect temporel et stochastique.

L'ASH ne fait explicitement référence au temps qu'à travers la dynamique des variables continues dont l'évolution est décrite par des équations différentielles du temps. Certains changements d'états ne dépendent que de cette évolution définie par certaines conditions (ce que certains appellent des événements endogènes). D'autres changements d'état peuvent aussi être dus à des événements aléatoires donc non prévisibles (on en connaît seulement qu'une distribution des probabilités d'occurrence). Le temps doit donc apparaître explicitement dans la définition de l'automate comme une variable partagée par les deux dynamiques continue et discrète. Il faut noter également que les lois de distribution de probabilités de défaillance des composants sont susceptibles de voir leurs paramètres dépendre de certaines variables continues (température, pression, etc.) mais aussi de la trajectoire suivie par le système (chemin suivi dans l'automate depuis l'état initial) pour décrire certains phénomènes de vieillissement (exemple nombre de commutations d'une vanne...).

III. EVALUATION DE LA SDF D'UN SYSTÈME DYNAMIQUE EN UTILISANT LA BOÎTE A OUTILS SCICOS DE SCILAB

Le modèle à simuler doit prendre en compte les éléments suivants :

- Le système est hybride : équations d'état continu et automates à états finis se combinent.
- Les équations d'état continu se reconfigurent sur occurrence des événements entraînant éventuellement une modification de la structure fiabiliste.
- Les variables et les événements peuvent revêtir un caractère déterministe ou stochastique.
- Il doit permettre « l'injection » des défaillances.
- Il doit inclure le diagnostic des défaillances et la réaction à ces défaillances en temps réel.
- Il doit pouvoir prendre en compte des lois de probabilités quelconques (notamment continues en fonction du temps, mais aussi discrètes) et l'interaction entre ces lois de probabilités et l'état continu du système.

A l'heure actuelle, il existe plusieurs approches développées pour évaluer la fiabilité dynamique. Mais, elles ne prennent pas en compte tous les aspects mentionnés ci-dessus. Dans [6] nous présentons un état de l'art de la fiabilité dynamique et les approches qui ont été développées pour traiter les problèmes posés par la fiabilité dynamique.

A. ASH implémenté avec des blocs existants dans Scicos

Avant que le concept d'automate hybride n'ait été introduit par l'INRIA dans l'environnement Scicos [5], nous avons développé une première approche pour coupler le

comportement des variables physiques d'un système dynamique avec le fonctionnement ou dysfonctionnement de ses composants en utilisant les blocs déjà existants dans Scicos [7]. L'intérêt du formalisme était bien entendu de simplifier l'aspect mathématique en évitant le recours à des expressions analytiques complexes. Nous avons pu prendre en compte les interactions entre fonctionnement et dysfonctionnement pour une évaluation fine de ces paramètres de SdF. Cependant, le modèle Scicos obtenu (fig. 1) était d'une complexité excessive par rapport à la simplicité de l'exemple choisi, ceci étant dû au fait que l'automate des états du système n'existe pas en tant qu'élément du modèle, mais il est diffusé dans le modèle au moyen des boîtes de commutation. Par ailleurs, l'approche souffrait du manque de caractère systématique et de réutilisation. Nous avons constaté qu'il n'est ni facile, ni pratique, ni efficace de modéliser ce petit système de cette manière. C'est pour cela que nous avons cherché ensuite un autre moyen ou manière d'implémenter l'ASH.

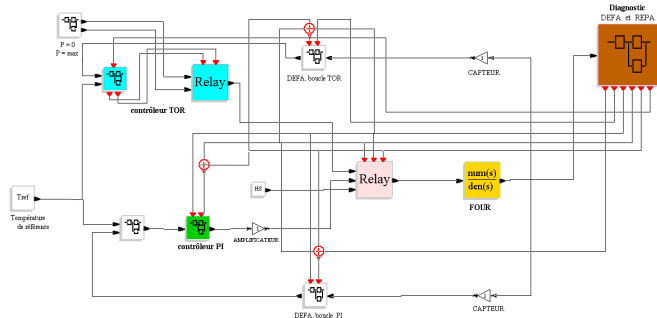


Figure 1. Implémentation de l'ASH dans l'environnement Scicos avec les blocs existants

B. Implémentation de l'outil DyRela

En prenant comme base l'implémentation dans Scicos de l'ASH que nous avons faite, l'outil DyRela a été conçu. La fig. 2 montre l'implémentation de l'ASH dans l'environnement Scicos de Scilab.

L'implémentation de l'ASH dans Scicos [8] est constituée de trois composants : un automate, un descripteur de modes et un générateur aléatoire (figure 2a). La figure 2b illustre une partie de l'implémentation dans l'environnement Scicos d'un système dynamique modélisé par l'ASH.

L'automate hybride est un bloc Scicos. Il est constitué (figure 1) de i ports d'entrée (à gauche du bloc) et de deux ports de sortie (à droite du bloc). La sortie notée « x, \dot{x} » fournit les valeurs des variables d'état continu x ainsi que leurs dérivées \dot{x} . La sortie notée « x_0, x_{i-1} » fournit l'état discret courant du système ainsi que l'état précédent. La sortie notée « e » produit un événement lorsque toute transition d'état discret est effectuée.

Le *descripteur de modes* du modèle de l'automate stochastique hybride correspond aux différentes dynamiques continues du système. Il y a autant de dynamiques continues que d'états discrets. Il a deux entrées : la première correspond aux variables d'état continues et à leurs dérivées provenant de l'automate hybride. La deuxième reçoit les valeurs fournies par le générateur aléatoire. Par ailleurs, le descripteur de modes a i ports de sortie chacun étant défini par le vecteur $[A, R, G]^T$.

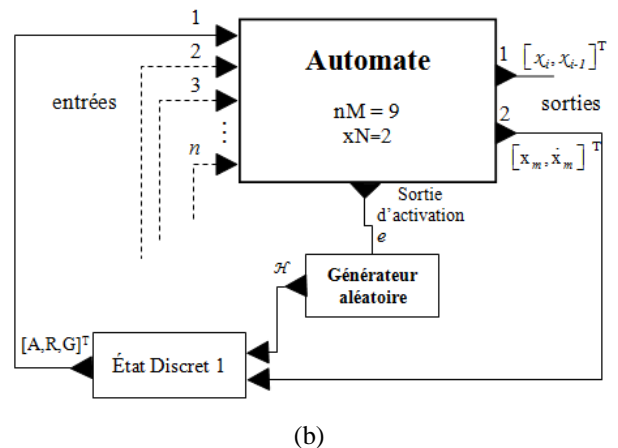
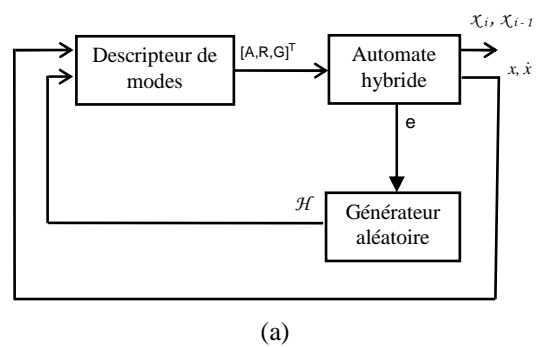


Figure 2. Implémentation de l'ASH dans l'environnement Scicos

Le *générateur aléatoire* correspond à la structure temporisée stochastique \mathcal{H} de l'équation (1). Le générateur aléatoire réalise des tirages aléatoires correspondant aux transitions aléatoires vers les états concernés à travers sa sortie. Chaque fois qu'une transition d'état discret se produit, la sortie « e » du bloc automate hybride génère un événement activant le bloc générateur aléatoire à travers son entrée (au dessus du bloc). A ce moment a lieu le tir des valeurs aléatoires.

De cette façon DyRela permet d'évaluer les paramètres de la SdF des systèmes dynamiques dont l'ASH modélise le système à traiter. La simulation de Monte Carlo est faite dans Scilab en mode Batch, c'est-à-dire, les simulations peuvent non seulement s'effectuer directement sur Scicos, mais aussi être relancées à partir de Scilab.

C. Amélioration de l'outil DyRela

Nous nous sommes proposé de réduire la durée de temps de simulation que prend DyRela pour évaluer les paramètres de la SdF. Il y a deux raisons principales pour le faire. D'abord, l'aspect graphique de Scicos sur lequel est implémenté l'ASH : le descripteur de modes est constitué de trois différents types de blocs Scicos. Le premier type correspond aux blocs qui modélisent chacune des dynamiques A du système. Le deuxième type de blocs fournit les valeurs de réinitialisation R des variables continues. Finalement, le troisième type de blocs correspond à la fonction de garde G pour chaque état discret. Un deuxième aspect est lié au solveur numérique : pendant la simulation tous les blocs Scicos du modèle du système sont actifs. Cela entraîne que le solveur numérique consacre du

temps pour évaluer toutes les fonctions définies par les blocs en provoquant un ralentissement conséquent de la simulation.

Pour optimiser DyRelA nous avons modifié l'implémentation de l'ASH dans Scicos en intégrant dans un seul bloc Scicos le descripteur de modes, l'automate et le générateur aléatoire dans le but de réduire la durée des simulations. Ce nouveau bloc nous l'avons nommé « SHA » (Stochastic Hybrid Automaton).

Scilab est un logiciel de calcul scientifique ouvert et libre. Scicos est une boîte à outils importante de Scilab et fournit un éditeur graphique pour la construction et la simulation des systèmes dynamiques [9]. En plus des blocs déjà existants, Scicos permet de construire de nouveaux blocs. Chaque bloc Scicos est défini par deux fonctions : la fonction d'interface et la fonction de calcul (fig. 3). La première est utilisée uniquement par l'éditeur de Scilab et la deuxième par le simulateur. La fonction d'interface doit être écrite en langage Scilab. La fonction de calcul est écrite normalement en langage C, mais elle peut être définie aussi en langage Scilab. La fonction d'interface traite les interactions avec l'éditeur, spécifie la géométrie du bloc, les entrées et les sorties du bloc, ainsi que le type de bloc. Elle traite aussi les paramètres et valeurs d'initialisation des états discrets et continus. La fonction de calcul définit le comportement du bloc pendant la simulation et elle est appelée plusieurs fois par le simulateur.

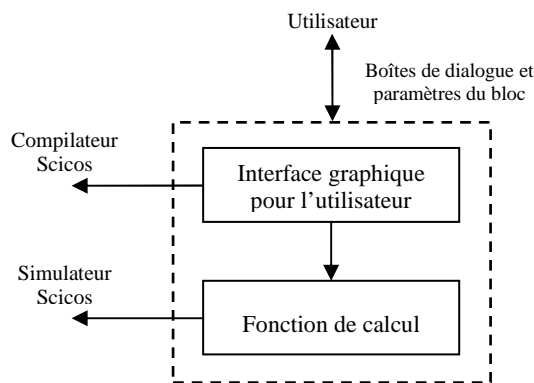


Figure 3. Bloc Scicos

La manière d'appel de la fonction de calcul est caractérisée par le type de la fonction d'interface : le type 4 est utilisé pour les programmes écrits en C et le type 5 pour les fonctions de calcul de Scilab. La fonction de calcul du bloc SHA a été écrite en C. Cette fonction reçoit deux arguments : la structure du bloc et un drapeau. Ce qui nous a permis d'intégrer les trois blocs de la première version de DyRelA sont la nature et les caractéristiques du drapeau. Le tableau 1 montre les différents drapeaux que nous avons utilisés et leur description.

Cette nouvelle implémentation de l'ASH prend en compte les différents modes continus de fonctionnement du système à travers des équations différentielles ordinaires et le passage de l'un à l'autre sur l'occurrence des événements déterministes et stochastiques. Les premiers événements sont produits par franchissement de seuils variables continues, les seconds sont produits par les défaillances des composants simulées par le générateur aléatoire, intégré au bloc, en fonction de leurs lois de probabilités.

TABLA 1. DRAPEAUX UTILISÉS DANS LA FONCTION DE CALCUL POUR DÉFINIR LE BLOC SHA

Drapeau	Description
0	Calcul la dérivée des états continus
1	Calcul les sorties du bloc
2	Actualise les états dus à une activation externe
3	Programme le temps d'activation
4	Initialise les états continus
5	Appel au bloc pour finir la simulation
9	Calcule le croisement par zéro

IV. LE CAS TEST

Le cas test que nous proposons a la prétention de permettre la prise en compte d'un grand nombre des problèmes relevant de la fiabilité dynamique de manière aussi réaliste que possible.

A. Description du cas test

Le cas test présente un four dont la température évolue en fonction de l'énergie qui lui est transmise (fig. 4). Dans notre exemple nous supposons que l'énergie est électrique, mais le modèle s'appliquerait également à la combustion par exemple. Le système de contrôle contient deux boucles de régulation. La première inclut un contrôleur proportionnel et intégral (PI) dont le rôle est de contrôler la température du four en fonction de la température de référence en fournissant à chaque instant la puissance juste nécessaire. Elle correspond au fonctionnement normal du système : on délivre l'énergie de manière continue. En cas de défaillance de cette boucle de régulation, une boucle de secours de type tout ou rien (TOR) permet de maintenir la température du four aux alentours de la température de référence $\pm \Delta T$ en chauffant à pleine puissance ou en ne chauffant pas. Les deux boucles ne doivent évidemment pas fonctionner en même temps. Pour cela, un relais bascule ses deux contacts permettant ainsi d'activer soit le régulateur PI, soit le régulateur TOR. L'ordre de basculement est donné par un système de détection dont le rôle est d'identifier les défaillances et les réparations et de réagir en commutant d'un régulateur à l'autre.

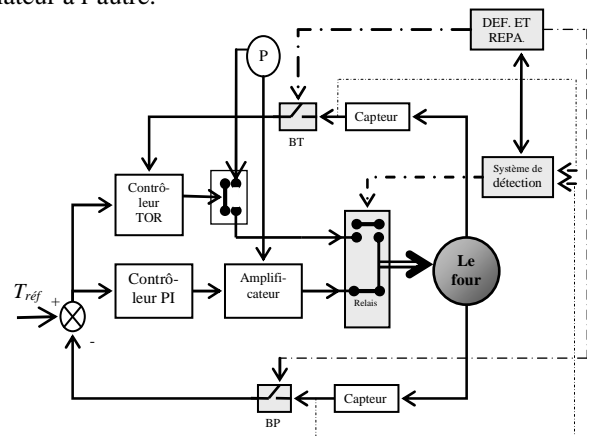


Figure 4. Diagramme structurel du système de contrôle de la température d'un four.

T_{ref} – Température de référence P – Alimentation de puissance BP – ouverture de la boucle du contrôleur PI. DEF. ET REPA. – défaillances et réparations BT – ouverture de la boucle du TOR

B. Description du comportement du cas test

Le système fonctionne de la manière suivante : au démarrage la température x du four est contrôlée par le contrôleur PI. Au bout d'un certain temps aléatoire, le contrôleur tombe en panne (caractérisé par un taux λ_{PI}) et la température du four augmente rapidement.

Le système de détection de franchissements de seuils détecte que la température du four a atteint une valeur dangereuse ($x \geq x_{max}$) déduisant ainsi que la température du four est hors contrôle. Il donne alors l'ordre au relais de basculer sur la boucle TOR. La boucle du contrôleur PI est maintenant ouverte et la boucle TOR fermée. La température du four est contrôlée maintenant par le TOR ($x_{inf TOR} \leq x \leq x_{sup TOR}$). Dès que le système de détection a repéré

que la température était hors contrôle, il a donné l'ordre de basculer au relais vers la boucle du TOR et enclenché le processus de réparation du contrôleur PI (on considère une réparation caractérisée par un taux μ_{PI}). Cependant, la possibilité de défaillance du régulateur TOR existe, après une durée également aléatoire (λ_{TOR}). Une fois que le contrôleur PI est réparé, le système de détection bascule le relais sur la boucle de celui-ci et ouvre la boucle du TOR. La température du four est maintenant à nouveau régulée par le contrôleur PI. On inclut également le processus de réparation du TOR (μ_{TOR}).

C. Résultats de la simulation

Nous avons appliqué l'ASH au cas test afin d'effectuer une simulation de Monte Carlo du système. L'ASH modélise le comportement déterministe et stochastique du système. Il est composé par 9 états discrets et une variable d'état continu : la température. L'ordinateur utilisé est un Pentium IV, CPU 3.2 GHz. et 0.99 Go de RAM. Les tableaux 2 - 4 montrent les récapitulatifs des résultats obtenus par simulation de la 1^{ère} version et de la version optimisée de DyRelA, mais aussi ceux obtenus par la solution analytique du processus de Markov et du processus semi-markovien correspondants. Concernant l'obtention du processus de Markov et du processus semi-markovien, il est à constater que le modèle ASH est isomorphe à un processus semi-markovien, car dans chaque état discret on a au maximum une transition déterministe due à un franchissement de seuil par la variable physique (température) qui est en compétition avec seulement des transitions stochastiques exponentielles (λ, μ constants). Le processus de Markov est obtenu par simplification de l'ASH en considérant que la durée du diagnostic des défaillances (par le franchissement de seuil de la température) est négligeable par rapport aux durées des états de fonctionnement et de réparation. On obtient ainsi un processus de Markov, car les seules transitions restantes dans le modèle sont des transitions stochastiques de taux constants (λ ou μ). Le processus de Markov et le processus semi-markoviens obtenus sont décrits en détails en [1]. Ces processus étant des processus stochastiques formels, une résolution analytique est possible et celle-ci est réalisée de manière quasi-instantanée en comparaison avec la durée des simulations de Monte-Carlo.

Les paramètres de la SdF évalués sont : le MTTF, le MTTR et la disponibilité du système dynamique proposé avec une

précision de 0.001 et une probabilité de 0.98. La simulation a été effectuée sur une plate-forme Linux.

TABLEAU 2. RÉCAPITULATIF DES RÉSULTATS OBTENUS PAR SIMULATION PAR LA 1^{ÈRE} VERSION DE DYRELA

Paramètre	ASH – DyRelA 1 ^{ère} version	Nombre de simulations	Temps de simulation
A	99,999%	34.800	28 h
MTTF	$2,270 \cdot 10^6$ h	37.350	18 h
MTTR	28,34 h	37.550	6 min

TABLEAU 3. RÉCAPITULATIF DES RÉSULTATS OBTENUS PAR SIMULATION PAR LA VERSION OPTIMISÉE DE DYRELA

Paramètre	ASH – DyRelA version optimisée	Nombre de simulations	Temps de simulation
A	99,999%	37.779	2,30 h
MTTF	$2,256 \cdot 10^6$ h	36.050	1,40 h
MTTR	28,54 h	36.750	1,17 min

TABLEAU 4. RÉCAPITULATIF DES RÉSULTATS OBTENUS ANALYTIQUEMENT PAR LE PROCESSUS DE MARKOV ET PAR LE PROCESSUS SEMI-MARKOVNIEN

Paramètre	Processus de Markov	Processus semi-markovien
A	99,998%	99,998%
MTTF	$2,039 \cdot 10^6$ h	$2,279 \cdot 10^6$ h
MTTR	28,57 h	28,57 h

La figure 5 montre le temps moyen d'accès à l'état de défaillance du système. Cet état a été rendu absorbant. Nous avons donc approché le MTTF par la moyenne du temps d'accès à l'état absorbant (MoyTAEA) sur l'ensemble des histoires simulées (une histoire est le passage du système depuis l'état initial par une suite d'états de fonctionnement avant d'arriver à l'état de défaillance).

La figure 6 présente le temps moyen de séjour dans l'état d'indisponibilité du système. Pour approcher la disponibilité asymptotique, nous considérons comme mesure le temps moyen de séjour dans l'état d'indisponibilité (T_{moySEI}) et à partir de celle-ci on détermine la disponibilité par $\bar{A}_{\infty} = I - A_{\infty}$.

La figure 7 montre le temps moyen d'accès à l'état de fonctionnement. Nous avons donc approché le MTTR par la moyenne du temps d'accès aux états de fonctionnement (MoyTAEF) sur l'ensemble des histoires simulées (une histoire est le passage du système de l'état défaillant vers les états de bon fonctionnement).

Indisponibilité du système

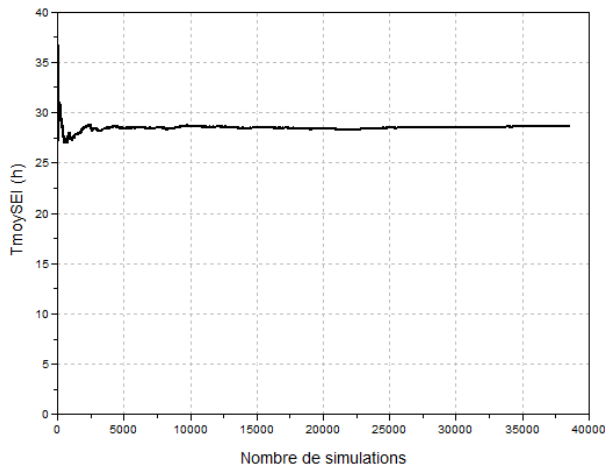


Figure 5. Temps moyen d'accès à l'état défaillance

MTTF du système

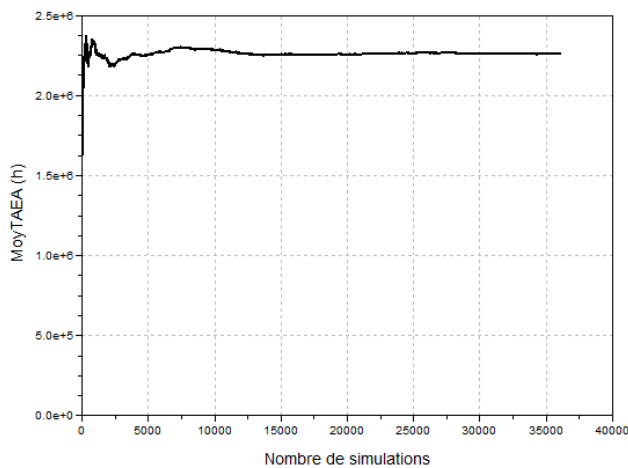


Figure 6. Temps moyen de séjour dans l'état d'indisponibilité du système

MTTR du système

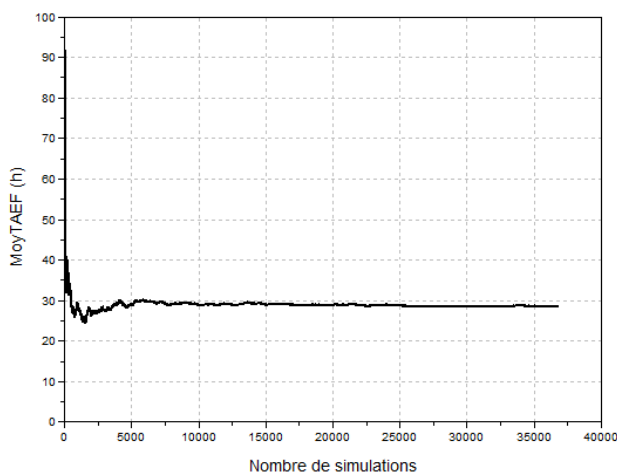


Figure 7. Temps moyen d'accès à l'état de fonctionnement

V. CONCLUSIONS ET PERSPECTIVES

Nous avons amélioré l'outil DyRelA en modifiant l'implémentation Scicos de l'ASH dans le but de réduire la durée des simulations nécessaires à l'évaluation de la SdF des systèmes en contexte dynamique.

Plutôt que de disposer de trois blocs Scicos différents et d'un ensemble de blocs définissant le descripteur de modes, nous avons construit un seul bloc Scicos. Cette nouvelle structure, nommée SHA, permet de modéliser le système dynamique à étudier. Nous avons réalisé une simulation de Monte Carlo du cas test et nous avons constaté la réduction considérable de la durée des simulations par rapport à la version précédente de DyRelA. Nous allons travailler pour rechercher d'autres modalités d'accélération de la simulation, notamment en mettant à profit les différentes échelles de temps présentes entre les aspects fonctionnels et dysfonctionnel et en parallélisant les calculs.

REFERENCES

- [1] G. A. Pérez Castaneda, "Evaluation par simulation de la sûreté de fonctionnement de systèmes en contexte dynamique hybride", Institut National Polytechnique de Lorraine – INPL, , Thèse doctorale, 2009.
- [2] G. A. Pérez Castaneda, J.-F. Aubry et N. Brinzei, "Modélisation d'un système par automate stochastique hybride pour l'évaluation de la fiabilité dynamique", Journal européen des systèmes automatisés, pp. 231-255, Volume 44 N° 2/2010.
- [3] G. A. Pérez Castaneda, J.-F. Aubry et N. Brinzei. "DyRelA (Dynamic Reliability and Assessment)"; dans 1st Workshop on DYnamic Aspects in DEpendability Models for Fault-Tolerant Systems, DYADEM-FTS 2010 in conjunction with European Dependable Computing Conference EDCC 8 - 1st Workshop on DYnamic Aspects in DEpendability Models for Fault-Tolerant Systems, DYADEM-FTS 2010 in conjunction with European Dependable Computing Conference EDCC 8, Espagne (2010).
- [4] T. A. Henzinger. The theory of hybrid automata. Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS), pp. 278 – 292, 1996.
- [5] M. Najafi et R. Nikoukhah, "Modeling Hybrid Automata in Scicos", Multi-conference on Systems and Control (MSC), Singapore, 1 – 3 October, 2007.
- [6] G. A. Pérez Castaneda, J.-F. Aubry et N. Brinzei, "État de l'art en fiabilité dynamique", 2e journées doctorales du GDR MACS JDMACS 2007, Reims, France.
- [7] G. A. Pérez Castaneda, J.-F. Aubry et N. Brinzei, "Modélisation et simulation d'un système dynamique hybride pour calculer sa fiabilité dynamique en utilisant le toolbox Scicos de Scilab", 7e édition du congrès international pluridisciplinaire Qualita 2007, Tanger, Maroc, 2007, p. 311-318.
- [8] G. A. Pérez Castaneda, J.-F. Aubry et N. Brinzei, "Simulation de Monte Carlo par Automate Stochastique Hybride, application à un cas-test pour la fiabilité dynamique", 8e édition du congrès international pluridisciplinaire Qualita 2009, Besançon, 2009.
- [9] S. L. Campbell, J.-P. Chancelier and R. Nikoukhah, Modeling and Simulation in Scilab/Scicos, Springer. 2006.