A NEW METHODOLOGY TO MODEL AND ASSESS RELIABILITY OF LARGE DYNAMIC HYBRID SYSTEMS

PERRINE BROY*

HASSANE CHRAIBI, ROLAND DONAT

EDF R&D, 1 avenue du Général de Gaulle 92141 Clamart France
email: perrine.broy@edf.fr; hassane.chraibi@edf.fr ; roland.donat@edf.fr

CHRISTOPHE BERENGUER

Gipsa-lab - Grenoble Institute of Technology, France
email: christophe.berenguer@grenoble-inp.fr

YANN DIJOUX

Université de Technologie de Troyes, Institut Charles Delaunay, France
email: yann.dijoux@utt.fr

This paper describes a new methodology to model and assess the behavior of industrial-sized dynamic hybrid systems. It consists of a tool platform PyCATSHOO, based on PDMP framework, stochastic hybrid automata representation and Monte Carlo simulation.

Introduction

In the field of large systems reliability, few studies take into account the evolution of a deterministic continuous variable. In other words, studied systems are rarely both dynamic hybrid and industrial-sized. The evolution of a hybrid system is a combination between discrete stochastic events on the one hand and continuous deterministic phenomena on the other hand. These systems are often referenced as falling under the dynamic reliability framework [1,2] and are present in the field of energy, hydropower generation domain in particular. We choose use Stochastic Hybrid Automata to describe and Monte Carlo simulation to quantify the behavior of our large hybrid dynamic system. Section 1 introduces the mathematical framework. Studied system and PyCATSHOO are briefly described in Section 2. Finally, some results and discussions are provided in Section 3.

*Corresponding Author, Tel.: +33 1 47 65 46 38

1. Mathematical framework

1.1. *Piecewise Deterministic Markov Processes (PDMP)*

Mathematically, hybrid systems are generally represented by Piecewise Deterministic Markov Processes (PDMP) [3]. A PDMP couples a discrete random variable $(I_t)_{t\geq 0}$ which represents the system configuration, on the one hand, and a continuous deterministic vector $(\mathbf{X}_t)_{t\geq 0}$ which characterizes a physical phenomenon that affects the system on the other hand. The evolution of $\mathbf{X}_t$ is governed by a differential equation depending on the system configuration i : $\frac{\mathrm{d}\mathbf{X}_t}{\mathrm{d}t} = \mathbf{v}(i, \mathbf{X}_t)$ where $\mathbf{v}$ is the velocity of $\mathbf{X}_t$ when the system configuration is i. The characterization of the system at time t, namely the marginal distribution of random vector $(I_t, \mathbf{X}_t)$, is a solution measure of the Chapman-Kolmogorov equations associated to the PDMP.

1.2. *Stochastic Hybrid Automaton (SHA)*

A SHA is a finite-state automaton controlling a set of deterministic continuous equations. The transitions between states are associated with probability distributions. The SHA have been used to describe hybrid systems in a nuclear context [4]. From the definition given in [5,6], a SHA is described by a 6-tuple $A = (M, X, (m_0, x_0), f, inv, T)$ where

- M is a set of modes (discretes states);
- $X \subseteq \mathbb{R}^n$ is a continuous state space;
- $m_0 \in M$ is the initial mode and x_0 is the initial continuous state;
- $f \ : \ M \to \mathcal{C}(\mathbb{R}, X)$ is a function which associates every mode m with the function f_m. $f_m = \frac{d\phi_m}{dt}$ characterizes the continuous dynamics for the mode m;
- inv is a function that associates each mode $m \in M$ to a subset $Inv(m)$ of X. $\{\exists t \in \mathbb{R}^+,\ \phi_m(t) \notin Inv(m)\}$ leads to the output at time t from mode m via one of the transitions $\tau \in T$;
- T is the set of transitions. Each $\tau \in T$ is characterized, among others, by its source m and its destination m', a jump rate function and a set of condition.

2. Describing and modeling system

2.1. *System description*

The studied hydraulic system is inspired by gated spillways with regard to some features. This concerns the nature and the number of its components as well as its complexity. The undesired event is the overflow of a maximum threshold by the dam level. This type of system, which includes about 75 components, is divided into four sub-systems, namely power supply, the Instrumentation and Control (I&C), the hydromechanic part and the basin. Each component is characterized by a failure distribution. Failures may occur on demand and/or in operation. Finally, the basin is characterized by the dam level. This level is a deterministic continuous variable whose evolution is governed by a differential equation $\frac{dh(t)}{dt} = q_{IN}(t) - q_{OUT}(t)$. This equation depends on the opening height of the valves. This opening height depends on the state of upstream components, transmission especially, which are discrete stochastic systems. Then the behavior of the spillway and each of its components can be represented by a PDMP and by a SHA respectively.

Let $Q = (q_0, q_1, \dots, q_n)$ be the vector of inflow values. Q is called the flood hydrograph. During the flood, the inflow into the basin is $\forall t \in [t_i, t_{i+1}[,\ q_{IN}(t) = q_i + \frac{[(t-t_0)-i\delta t](q_{i+1}-q_i)}{\delta t}$ where $q_i = q_{IN}(t_i)$, t_0 is the flood start time and δt is the time interval between two measurements of the inflow.

Let $h(t)$ be the level at time t. The conveyance of a valve is given by $q_{OUT}(t) = f(h(t)) \times op(t)$ where f is a function whose parameters depend on each type of valve and $op(t)$ is the height of the valve opening. For a flapgate, $f(h(t)) = mL\sqrt{2g \cdot \max(0, h(t) - lt)}$ where m is a coefficient, L is the valve width, g is the gravitational constant and lt is the lower threshold of the flapgate. $op(t)$ is the valve opening height at time t. It is a percentage of its total height, which depends on the start opening time, on an eventual failure time ($t_{fail} = \infty$ by default) and the duration of the opening operation.

2.2. *System modeling with PyCATSHOO*

PyCATSHOO [6] (PythoniC Object Oriented Hybrid Stochastic AuTomata) is a tools platform designed by EDF R&D. It is aimed at estimating reliability of dynamic hybrid systems. PyCATSHOO is based on PDMP formalism, multi-agent paradigm and object-oriented and functional programming.

PyCATSHOO architecture is structured into a set of Python classes. These classes allow to describe in a knowledge base the behaviour of the components involved in a range of target systems. In knowledge bases, each type of component is modeled by a PyCATSHOO class (PyC) which is a triplet (S, RMB, SMB). S is a stochastic discrete or hybrid automaton, RMB are Receiving Message Boxes and SMB are Sending Message Boxes. Specific systems are instanciated as a set of dependant components declared by means of PyCs. Monte Carlo simulation technique [7] is then used to compute relevant indicators on the considered system.

Note that some classes can be provided with a deterministic continuous variable. This variable will be calculated and updated as soon as the impacting transitions occur. In our system, the dam level calculation is hosted within the Basin class. To evaluate it, the basin uses information received from the flood (which characterizes hydrograph) and from valves (potential failures that prevent correct opening).

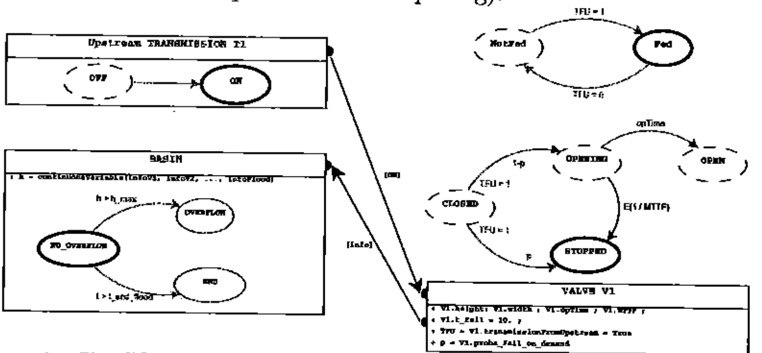

Figure 1. Simplified scheme of three PyCs. With active modes in bold, this scheme is a "picture" of few knowledge base instances at a given time $t > 10$. Attributes, modes, transitions and message boxes of class Valve are represented.

An important part of our task was to model each class of component by carefully defining their states, the conditions of transitions between different states, and the interactions with other components.

3. Results

PyCATSCHOO provides results such as events histories or indicators about variable of interest like means and confidence intervals. Figure 2 is the evolution of overflow probability where $t = 0$ is a century flood start time, for $N = 350000$ simulations. Most of overflow issues are due to on-duty operator (2 days out of 7). One part of the stories concerns the failure of one or more gates during opening. As can be seen in Figure 3, the stories issues depend on the time failure of two gates, given that two other gates are already out of order before the flood, and the last two ones will open correctly. Indeed, if a gate fails during its opening, then its conveyance will depends on the failure time.

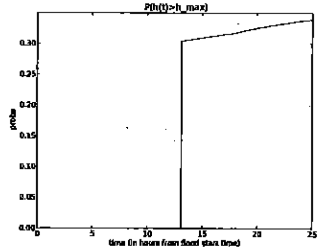

Figure 2. Evolution of overflow probability

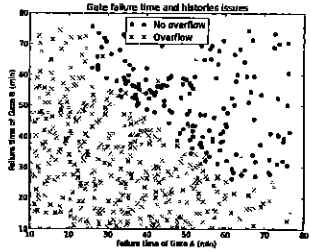

Figure 3. Gate failure time and histories issues

Conclusion

We have modeled an industrial-sized complex dynamic hybrid systems with tool PyCATSHOO. This approach will be soon improved by describing more precisely the operator, which could move on site when on-duty and repair some failed components. In future works we will also make the results more attractive than simple events histories. For exemple, by judiciously bringing together similar stories both from the perspective of components behavior and the global issue, we could identify the most probable faulty sequences that lead to the assessed system undesirable events.

References

[1] P.E. Labeau, C. Smidts and S. Swaminathan. Dynamic reliability: towards an integrated platform for probability risk assessment. *Reliability Engineering and System Safety 68*, 3, 219-254 (2000).

[2] P. Broy, H. Chraibi and R. Donat. Using Dynamic Bayesian Networks to solve a dynamic reliability problem. *Advances in Safety, Reliability and Risk Management*, 335-341 (2011).

[3] M. Davis. Piecewise-deterministic Markov processes: A general class of non-diffusion stochastic models. *Journal of Royal Statistical Society. Series B (Methodological) 46*, 3, 353-388 (1984).

[4] J.F. Aubry, G. Babykina, N. Brinzei et al. The APPRODYN project: dynamic reliability approaches to modeling critical systems. *Supervision and Safety of Complex Systems*, 181-222 (2012).

[5] A. A. Julius. Approximate abstraction of stochastic hybrid automata, in Hybrid Systems: Computation and Control. *LNCS, Springer Verlag*, vol. 3927, 318-332 (2006).

[6] H. Chraibi. Dynamic reliability ans assessment with PyCATSHOO: Application to a test case. *Proceedings of PSAM 2013, Tokyo, Japan.*

[7] http://simpy.sourceforge.net/