RESILIENCE ASSESSMENT OF INTERCONNECTED CRITICAL INFRASTRUCTURES WITH PYCATSHOO

HASSANE CHRAIBI

EDF Lab Paris Sacly, 7, Bd Gaspard Monge, 91120 Palaiseau, France
email: hassane.chraibi@edf.fr

Great importance is attached to improving the resilience of Interdependent Critical Infrastructures (ICI). In particular, recent advances in Supervisory Control And Data Acquisition technology have allowed the implementation of optimization mechanisms that minimize the impact of disruptions and improve recoveries. In order to give a fair assessment of the resilience of a system, it is now necessary to take these optimization mechanisms into account. However, few resiliency assessment frameworks do. This paper presents PyCATSHOO —a platform recently introduced by EDF— capable of modeling complex dynamics in a system. One such dynamic will be expanded upon in this paper: reconfiguration capabilities that are based on optimization algorithms such as integer and mixed linear programming. This will be illustrated through the resilience assessment of a simplified gas and electricity transmission network, in which some consumers have a higher priority.

Keywords: Resilience, critical infrastructure, complex systems, piecewise deterministic Markov processes.

1. Introduction

Much effort is directed towards maintaining a standard of resilience in Interdependent Critical Infrastructures (ICI) such as the transmission and distribution of electricity, water, gas and telecommunication. There are many variants in the definition of the resilience of a system [1], which in turn give different mathematical objectives.

In general terms, a system's resilience is its ability to minimize the gap between its nominal state and states taken by the system after a disruptive event. Resilience assessment frameworks should be flexible and adapt to different assessment objectives, and to systems with different missions and different types of disruptions. They should also be able to model complex systems without excessive simplification.

Classical approaches which rely on boolean representation and structural aspects of systems, do not meet these requirements [2].

These approaches are especially not adapted to the presence of internal interdependences in ICI systems. ICI systems may additionally be subject to two kind of phenomena that renders classical methods ineffective: 1. Discrete stochastic disruptions that induce system reconfigurations (which often rely on advanced optimization algorithms [3] [4]). 2. Continuous deterministic physical phenomena.

This paper presents PyCATSHOO [5] —a platform recently introduced by EDF— which satisfies all the above mentioned requirements. This will be illustrated through the resilience assessment of a simplified gas and electricity transmission network, in which some consumers have a higher priority.

The next section introduces the model of this system and the formulation of the resilience indicators. Section 3 explains the main concepts of PyCATSHOO. Finally, section 4 presents the assessment results and shows PyCATSHOO's ability to factor in complex optimization objectives.

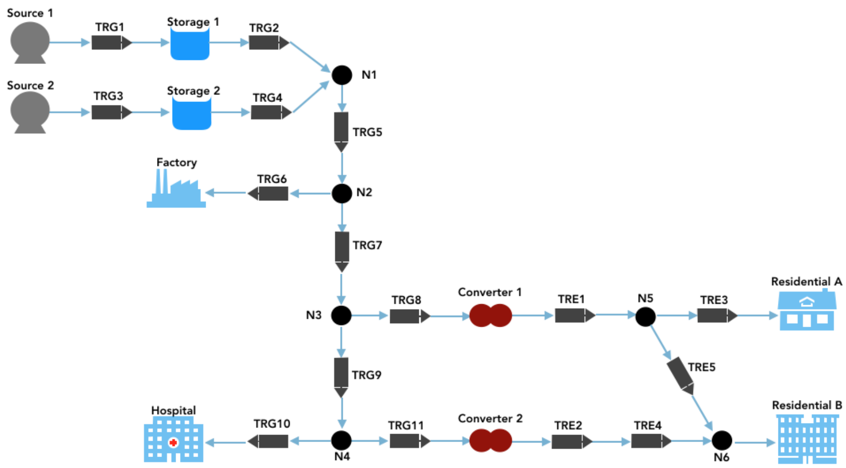

Figure 1. Gas-Electricity distribution network

2. Test-case: Gas-Electricity transmission system

Consider a gas and electricity network which supplies consumers with different levels of priority. As shown in figure 1, this system comprises two gas sources, "Source 1" and "Source 2". The gas is transported through several pipes, called "TRG i", to two consumers: "Factory" and "Hospital". The gas also supplies two converters, "Converter 1" and "Converter 2", which produce electricity. This electricity is transported through lines called “TRE i” to two other consumers, “Residential A” and “Residential B”. The network includes several connexion nodes, “Ni”, which stand for convergence or bifurcation points.

The two sources are assumed inexhaustible and deliver constant capacities: SC_i. “Storage 1” and “Storage 2” behave like storage buffers. The volume of their content VS_i evolves according to the following set of ordinary differential equations (1):

$$\frac{dVS_i}{dt} = q_i^{in} - q_i^{out} \tag{1}$$

where q_i^{in} is the incoming flow-rate to “Storage i” and q_i^{out} is the outgoing flow-rate. In addition, these two values are constrained by the following set of equalities (2):

$$\begin{cases} q_i^{out} = 0 & if\ VS_i < lVS_i \\ q_i^{in} = 0 & if\ VS_i > hVS_i \end{cases} \tag{2}$$

where lVS_i is the volume threshold under which the buffer is considered empty and hVS_i is the volume threshold above which there is an overflow. Each one of the pipes “TRG i” and lines “TRE i” in the network has a maximal transportation capacity TC_i that the carried flow-rate F_i cannot exceed. This can be expressed by the following set of inequalities (3):

$$F_i \leq TC_i \tag{3}$$

Each consumer in the network has a gas or electricity demand CD_i. The network should provide each consumer with a flow CF_i^{in} that is as close as possible to the demand and that doesn’t exceed it. The network should additionally take into account the priorities CP_i of each consumer (in decreasing order, *Hospital, Factory, Residential B and Residential A*). All these constraints can be formulated by inequalities (4) and by the maximization objective (5).

$$CF_i \leq CD_i \tag{4}$$

$$\max\left(\sum_{i\in\mathbb{C}S} CP_i \times CF_i^{in}\right) \tag{5}$$

where $\mathbb{C}S$ is the set of consumers in the network.

In addition, we have to take into account other constraints deduced from the material balances around every connexion node.

The mission of this network is to provide consumers with gas or electricity in order to meet their needs. We consider that there exist 3 kinds of disruptions to this mission:

1. The failure of a source, which makes its supply drop to a value drawn uniformly between 0 and its nominal supply rate.
2. The failure of a transportation portion “TRG i” or “TRE i”, which makes its capacity drop to a value drawn uniformly between 0 and its nominal capacity.
3. The failure of a converter, which makes its capacity drop to 0.

This kind of system is widespread and can be critical when it supplies high priority customers, industrial or dense residential areas.

System resilience can here be measured with two indicators. The first one has to do with the system robustness and, the second one, with the recovery efficiency.

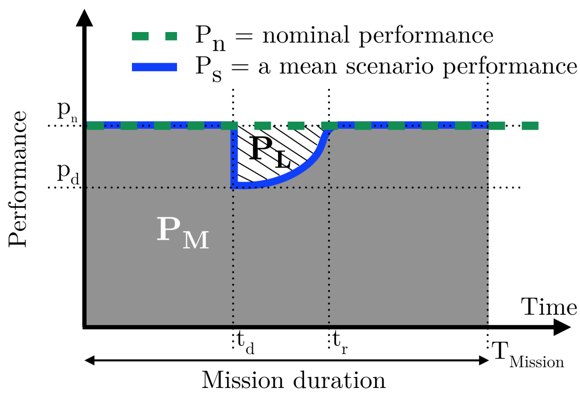

Figure 2. Evolution over time of a scenario performance

Figure 2 gives the evolution of performance in a specific scenario. In this scenario we observe an instantaneous performance drop which corresponds to a disruption. The magnitude of this drop gives an idea of the system’s robustness. We then adopted the mathematical expression given in (6) as a robustness performance indicator:

$$I_{robustness} = \frac{p_d}{p_n} \tag{6}$$

where p_n is the process performance, just before the disruption occurrence and p_d the minimal value reached after such an event.

As for the recovery efficiency, the indicator that we have adopted in this study is given by equation (7).

$$I_{recovery} = \frac{\int_0^{T_{Mission}} P_s dt}{\int_0^{T_{Mission}} P_n dt} = \frac{P_M}{P_M + P_L} \tag{7}$$

where P_n is the system's performance in a nominal scenario and P_s is the system's performance of a mean scenario with disruptions.

3. PyCATSHOO basics

PyCATSHOO is a tool recently developed at EDF. It is currently released as freeware and can be downloaded from pycatshoo.org. PyCATSHOO is dedicated to the probabilistic performance assessments of complex systems. In particular, it is able to address hybrid models *i.e.* models which mix deterministic and continuous physical phenomena on the one hand, and, on the other hand, discrete and stochastic behavior. It is based on the theoretical framework of Piecewise Deterministic Markov Processes (PDMP) [6] formulated as distributed hybrid stochastic automata. A PyCATSHOO model is a collection $\mathbb{M} = (\mathbb{O}, \mathbb{P}, \mathbb{L}, \mathbb{I})$ where:

- $\mathbb{O}$ is a set of system components. Each component of this set contains:
  - $\mathbb{V}^c$, $\mathbb{V}^d$ and $\mathbb{V}^r$ are sets of respectively continuous variables, discrete variables and variable references. Note that while the elements of $\mathbb{V}^c$ and $\mathbb{V}^d$ are intrinsic variables, the element of $\mathbb{V}^r$ are references to variables from other components.
  - $\mathbb{B}$ is a set of message boxes. A message box contains incoming and outgoing channels. An outgoing channel is linked to an intrinsic variable (element of $\mathbb{V}^c$ or $\mathbb{V}^d$). This allows the variable to be read by other components. An incoming channel is linked to a variable reference (element of $\mathbb{V}^r$). This allows the variable to be written by the outgoing channel of another component.
  - $\mathbb{A}$ is a set of automata. An automaton contains a set $\mathbb{S}^a$ of states and a set $\mathbb{T}^a$ of transitions. A transition is characterized by its source (element of $\mathbb{S}^a$) and by a set of targets (subset of $\mathbb{S}^a$). It is also characterized by a condition and a probability distribution. The parameters of this probability distribution are defined as functions of the component variables. As for the condition, it is a boolean expression of the component variables.

    Note that a transition without probability distribution is deterministic. It is called *forced transition* and is triggered when its condition becomes true. When such condition involves continuous variables, it defines a boundary crossing in the component PDMP. A transition

with a probability distribution is called *spontaneous transition*. It provokes a spontaneous jump to a different component state.
  - $\mathbb{F}$ is a set of actions that the system should trigger after specific events occur. There are four kinds of events: entering or leaving a state, a transition triggering, an automaton state change and a writing of a new value in a reference variable.

- $\mathbb{P}$ is a set of PDMPs. Each PDMP contains:
  - References to the continuous variables managed by the PDMP. Their values are computed by the PDMP by solving ordinary differential equations or by using explicit expressions.
  - References to different component methods. These methods implement explicit expressions or ordinary differential equations for continuous variable calculations.
  - Forced transitions whose conditions involve continuous variables. The conditions of such transitions define the boundaries of the PDMP's modes.
- $\mathbb{I}$ is a set of Mixed-Integer Linear Programming systems (MILP). Each element of this set contains:
  - References to variables managed by the MILP.
  - References to different component methods which implement inequalities that belong to MILP.
  - The objective function to maximize, which is constructed piece by piece in different elements of the sets $\mathbb{F}$ which belong to different components (elements of $\mathbb{O}$).
- $\mathbb{L}$ is a set of linear equation systems. A linear equation system is almost identical to an element of $\mathbb{I}$, except for the fact that inequalities are replaced by equalities and that it defines no associated objective function.

4. Resilience assessment

By using PyCATSHOO to model the system presented in section 2, we can simulate response to failures and assess the resilience of the system (by calculating the two indicators introduced in section 3: robustness and recovery efficiency).

Let us consider the response to a failure of the transportation portion “TRG 4” at time 200*h*. For simplicity, we inhibit failures in all other components.

In figure 3 and in table 1, we observe that the higher the consumer priority, the higher the resilience. In particular the hospital supply is not affected by the disruption. This behaviour cannot be explained by the structure of the system:

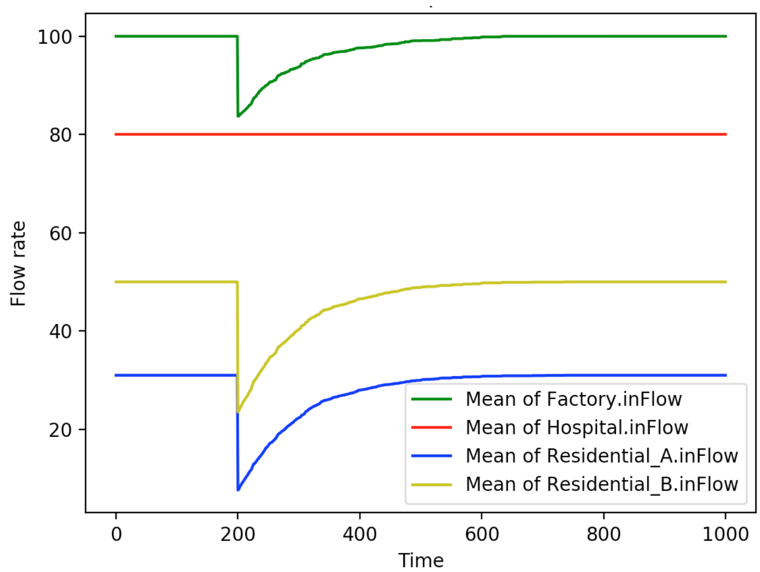

Figure 3. System response to a loss of gas transportation portion

Table 1. Indicators of the system Robustness and recovery efficiency

Indicators	Hospital	Factory	Residential B	Residential A
Demand	80	100	50	30
Priority	4	3	2	1
Robustness	100%	83.73%	47.09%	19.06%
Recovery efficiency	100%	98.36%	94.88%	71.81%

it is due only to operating rules, which rely on optimization algorithms. This is typically what most resilience assessment framework struggle to simulate, and what PyCATSHOO is good at.

5. Conclusion

This article gives a simple but compelling example of the limitations of classical resilience assessment frameworks. These limitations motivated the development of PyCATSHOO. This framework gives more accurate resilience assessments by making it possible to model —without excessive simplifying assumptions— complex optimization mechanisms in systems which mix deterministic and continuous physical phenomena, and discrete and stochastic behaviors.

Note that this does not come without a cost: PyCATSHOO requires more computation time. This is mitigated using parallel acceleration and will be even more so in the near future as current work at EDF is integrated into PyCATSHOO.

References

[1] A. Sharma. Resilience capacities assessment for critical infrastructures disruption. Ph.D. Thesis, Department of Civil and Environmental Engineering Politecnico di Milano, 2017. http://hdl.handle.net/10589/134640

[2] E. Zio. Challenges in the vulnerability and risk analysis of critical infrastructures. *Reliability Engineering and System Safety.* 152, 137–150, 2016

[3] Y. P. Fang, G. Sansavini. Optimum post-disruption restoration under uncertainty for enhancing critical infrastructure resilience. *Reliability Engineering and System Safety.* 185, 1–11, 2019

[4] C. Gomez, J. W. Baker. An optimization-based decision support framework for coupled pre- and post-earthquake infrastructure risk management. *Structural Safety.* 77, 1–9, 2019

[5] H. Chraibi, J.C. Houdebine, A. Sibler. PyCATSHOO: Toward a new platform dedicated to dynamic reliability assessments of hybrid systems. *13th International Conference on Probabilistic Safety Assessment and Management (PSAM 13).* 2-7 October, 2016. Seoul, Korea.

[6] M. H. A. Davis. Piecewise-deterministic Markov processes: a general class of non-diffusion. *Journal of the Royal Statistical Society.* vol. 46, no. 3, pp. 353–388, 1984.