Benchmark for the identification of cyber-physical systems to hybrid automata

G. Lembrez, A. Calame, G. Faraut

***Abstract*— Cyber-physical systems are systems in which software elements are in control of physical entities. On the occurrence of discrete events, these systems can switch between states in which their behaviour is continuous. Most industrial systems fall under this characterisation, rendering critical the ability to identify them to manageable models. However, the development of an identification method requires a collection of training examples. The hybrid automaton formalism allows for a systematic approach to the design of cyber-physical systems. The aim of this study is to provide a benchmarking library of automata for the identification techniques of cyber-physical systems to such models. Various examples from literature have been collected and simulated in the Stateflow environment. To facilitate the choice of an example in the library, a set of criteria is created to separate the models by representative characteristics.**

***Index Terms*— Cyber-physical systems, Hybrid automata, Identification, Stateflow**

I. Introduction

THE term ”cyber-physical systems” refers to systems in which computational elements are embedded in physical processes. As such, the software elements and the physical variables are mutually affected by one another. This coupling, along with the diversity of the physics covered by this definition renders challenging the modelling of such systems. However, most modern industrial systems are cyber-physical systems. Yet, the modelling of a system requires a high level in expertise which negates the interest of this broad concept.

Systematic identification techniques allow to shortcut expert intervention. The core idea is to automatically create a model from the data collected on the physical system. However, designing a method for the identification of cyber-physical system requires a formalism able to model any system that falls under this definition. In addition, a training database is required to assess the performances of the method.

The hybrid automata formalism allows for the modelling of cyber-physical systems. The theory of hybrid automata is a generalization of the theory of finite state automata with the addition of continuous variables. Those variables combined with the discrete transitions provide a model for the combination of physical entities and software elements.

The aim of this study is to create a database of hybrid automata that can be used to evaluate identification methods of cyber-physical systems.

A. Objectives

Although the theory of hybrid automata has been used multiple time in previous works, no compilation of examples exists. We aim to address the lack of an usable library of hybrid automata. Thus, we intend to meet the three following objectives

- Gathering examples from literature. In fact, various past studies concerning hybrid automata models are relevant from an identification perspective. In addition, working with real systems rather than purely theoretical ones is preferable from an industrial point of view.
- Creating a compilation of hybrid automata based on the synthesis of the collected examples. All automata from the database will be simulated using the stateflow environment and the stateflow model will be made available.
- Making the benchmark accessible. To meet this objective, we will design and provide a set of criteria with which to assess the relevance of an example with respect to the aim of the identification.

II. Preliminaries

A. The simulation environment

The development of the automatas has been carried out using the stateflow environment from MATLAB-simulink. The aim of this section is to describe the hypothesis and features of this environment.

Stateflow semantic is based on Harel’s statecharts model [2]. Although, statecharts offer a rich description of timed automaton, they do not allow the modelization of hybrid systems. Hybrid automata theory can provide this description. Several formal definition of the theory exist [3], [7] which share common concepts. Stateflow models are an enrichment over statecharts which allows to take into account the concepts of location and flow from hybrid automata theory.

Figure 1 provides an illustration of a simple stateflow model of a thermostat. The basic concepts of stateflow semantics are as follow :

- The system is in a succession of states called *locations*. In 1, the thermostat can be in two states : Heat on or Heat off.
- While in a location, the behaviour of the system is dictated by the associated differential equation, called *flow*. In the example, the temperature satisfies a first order differential equation.

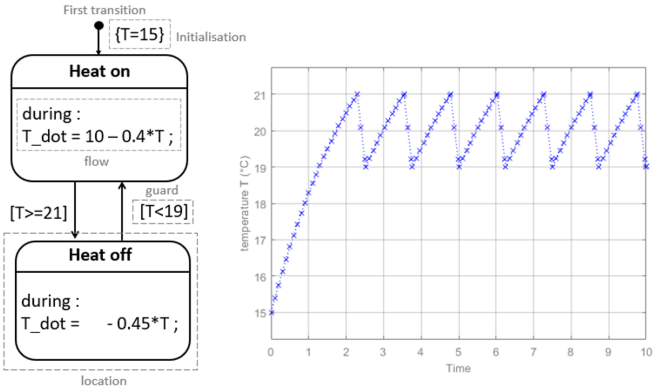

Fig. 1. Stateflow model of a thermostat. This academic model is commonly used to introduce the theory of hybrid automata as it is a simple cyber-physical system.

- The transition between two locations occurs as soon as the invariant associated with the transition is verified. This invariant is called guard and noted $[G]$.

B. Choices and hypothesis

We chose to limit the base definition of hybrid automata to the simple concepts listed above. In fact, most automata found in literature are limited to a set of continuous variables, a set of locations, a set of transitions associated with a set of guards thus motivating the limitation of the definition. In the following we will examine the ways in which this definition can be expanded upon. To diversify the automata we propose, we will use in priority Raskin's formalism [7]. Here we list the specifics of simulation of hybrid automata in stateflow.

- Raskin's theory enables the utilisation of events, which are essentially rising edges with no duration triggering transitions. However, due to the solver used, this type of variable is not allowed in the stateflow environment.
- Variable affectation is enabled at the end of transitions for all type of variables. However, only discrete variables can be updated within a location. The update occurs when the location is left.
- Stateflow allows the use of orthogonal product between automata [2]. This operation consists in having the automata work in parallel while sharing variables. This feature is especially useful to simplify the representation of heavy models.

III. Criteria to motivate the selection of an automaton

In this section, we list the criteria that are relevant for the comparison of automata. The method used to isolate those criteria consists in the adoption of two points of view. First, from the simple definition of the formalism provided in part II-B we considered the different enrichment allowed by Raskin's theory. Each generalization of the formalism allows to simulate more complex behaviours, thus making a relevant criterion. Then, we examined the feature used in previous work of identification to further compare automata modeled with the same formal rules.

- **Coupled variables (C1) :** this criterion is true if the flow dictating the evolution of a continuous variable depends on another continuous variable. Coupled systems require different methods of identification, which justifies the use of this feature as a criterion.
- **Autonomy (C2) :** an autonomous model is a model which does not observe input data during simulation. Non-autonomous model can be fed data, allowing them to simulate more complex behaviour and rendering the identification procedure dependant on the input as well as the identification method.
- **Discontinuities (C3) :** although the variables are supposed continuous, Raskin's theory allows to take into accounts jumps. Jumps are the affectation of a value to a continuous variable at the end of a transition. Automata that use this criterion comprises piece-wise continuous variables instead of purely continuous variables.
- **Flow classification (C4) :** this criterion regroups the mathematical properties of the flows of the automaton. First we consider the O.D.E. 5Ordinary Differential Equation) classification, then we examine some specific behaviours. Among the specifics are the first order equations and the flows of the $\dot{y} = C$ type. The latter is highlighted as linear hybrid automata constitutes another branch of the theory of hybrid automata.
- **Variable type (C5) :** by default, the variables dictating the evolution of the automata are continuous. However, the theory of hybrid automata allows the utilisation of discrete variables. Discrete variables are sampled over time in a similar fashion as continuous variables, the key difference being that they can only have a finite number of evaluation. The most interesting use of this addition is in the clustering of the behaviour of the system. Finally, as stated before, the use of events is not enabled in stateflow.
- **Representation of time (C6) :** The last enrichment we considered adding is the definition of timed hybrid automata. The definition of clocks permits the utilisation of temporal guards. In other words, with this extension we are able to add delay in the transitions. In stateflow, this action is done using the $after()$ function.

IV. Presentation of examples

Here we present some of the automata from the final benchmark. We will focus first on the modelling of the physical system, then its representation as an hybrid automata.

A. Predator-prey model

1) Physical system: In Trieste (Italy), at the end of the First World War, the fishing catches had decreased. The fishery office had noted that the proportion of shark type fishes, not very interesting for consumption, had increased considerably compared to the interesting sardine type fishes. The fishery office asked for help from Volterra who modeled the shark-sardine system.

The system model is based on two differential equations where $x(t)$ and $y(t)$ represent respectively the quantity of

sardines and the quantity of sharks :

$$\begin{cases} \dot{x}(t) &= a.x(t) - b.x(t).y(t) \\ \dot{y}(t) &= c.x(t).y(t) - d.y(t) \end{cases}$$

with $a, b, c, d > 0$

This model, also called the Lokta-Volterra system, means that in the absence of sharks, sardines proliferate $\dot{x} = ax$, in the absence of sardines, sharks disappear $\dot{y} = -dy$. The term with $x(t).y(t)$ represents the meeting of sharks and sardines which increases the number of sharks and decreases the number of sardines.

2) Automaton:

a) Variables: The automaton does not have input variable. The output of the system are the continuous variables represented in the array X :

$$X = \{x, y\}$$

b) Parameters: The different coefficients shown above represent the parameters. Another coefficient noted f has been added in order to represent the influence of human fishery.

c) Locations: The system has two locations, the first one represent the evolution of the sardine and shark populations, the second one add the influence of human fishery.

B. Drone failure

1) Physical system: This automaton is a model of a 6 propeller drone represented figure 2. The automaton simulates the behaviour of the hovering drone as it attempts to remain stable after some of its motors shutdown. Thus, each location corresponds to a failure scenario and the associated flow comprises the equation of the system's dynamic. Failure scenarios and equations of dynamic originate from previous work on the control of hovering hexacopter during failure [8] . The drone is located by the position and orientation of its center of mass. To describe the system's dynamic, both position and velocities have to be known. The state of the system is described in the vector :

$$S = (x\ y\ z\ \phi\ \theta\ \psi\ u\ v\ w\ p\ q\ s)^T$$

Where (ϕ, θ, ψ) are the Euler angle, (u, v, w) the acceleration and (p, q, r) the angular acceleration. In addition, the variables of the system are the rotation speed of each propeller Ω_i. In the original paper, the speed of each controller is supposed to be positive, therefore the change of variable $\Omega_i = \Omega_{r,i}^2$ has been made. Each rotor speed can be individually controlled.

$$\Omega = (\Omega_{r,1}, \Omega_{r,2}, \Omega_{r,3}, \Omega_{r,i4}, \Omega_{r,5}, \Omega_{r,6})^T$$

The forces exerted on the system are the thrust, the rotor drag, the air resistance and the weight. The moments exerted on the system are the torques due to rotational rate difference between rotors, gyroscopic effect and yaw counter torque.

2) The automaton:

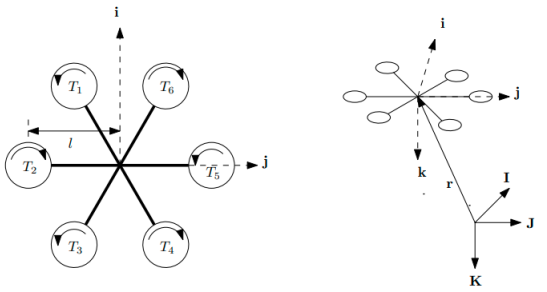

Fig. 2. Geometry of studied system (source : original paper). Note that the vertical axis is oriented in the direction of the weight force.

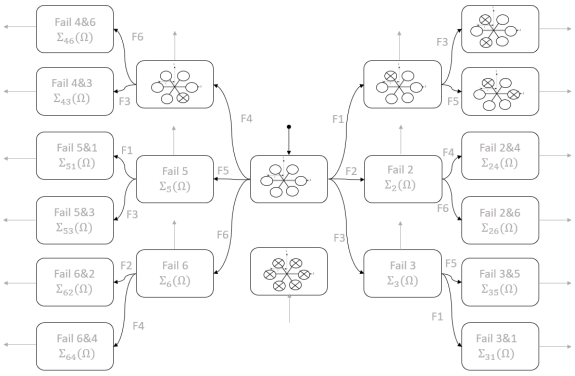

Fig. 3. simplified view of the automaton. A rotor is crossed if it is not working. The gray transitions are triggered by any motor failure that cannot be stabilized.

a) The transitions: The automaton corresponds to the decomposition of the behaviour of the drone in modes, see the simplified view figure 8. Each mode corresponds to a failure scenario, meaning a set of rotors not working. The transition between modes is regulated by 6 discrete variables

$$F_i \in \{0, 1\} \quad i \in \{1, 2, .., 6\}.$$

Where $F_i = 1$ as soon as motor i encounters a failure. The modelling of motor failure with discrete variables is a substitute for events. In this configuration, there is no difference between events and our work around.

b) The locations: The key difference between each state of the system is the way in which the action of gravity is compensated by the thrust force. This action has to be distributed across motors so that the drone can remain in the air while not experiencing torque along its pitch and roll axis. Let $F_t = b\sum\Omega_i^2$ the thrust force. Figure 4 represents the failure states associated with the following rotor speed :

- case 1 :

$$\Omega_i = \sqrt{\frac{mg}{6b}} \quad \forall i$$

- case 2 :

$$\begin{aligned} \Omega_6 &= \sqrt{\frac{2mg}{5b}} \\ \Omega_i &= \sqrt{\frac{mg}{5b}} \quad \forall i \in \{2, 3, 4\} \end{aligned}$$

Finally, this model allows for the use of different control strategy. For now, only a simple feedback loop with proportional correction is implemented and the coefficient have only been

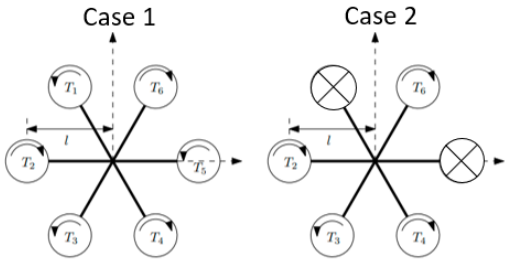

Fig. 4. Two cases of failure corresponding to two weight compensation scenarios

tuned via trial and error. In addition, control is limited to three DOF : pitch, roll and translation along K axis. Equation 1 presents the control equation for rotor i. The only difference between rotors is the sign of K_θ and K_ϕ which depends on the contribution of the propeller to pitch and roll.

$$\dot{\Omega}_i = K_w w + K_z(z_0 - z) + K^i_\theta \theta + K^i_\phi \phi \tag{1}$$

To summarize, each flow will be comprised of the fundamental principle of dynamic to update S and the equations of control to update Ω.

C. Automatic transmission

1) The physical system: This model is a simplified 3-gear automatic transmission system [5]. The efficiency of a system which delivers mechanical power varies with respect to load and rotation speed. To guarantee an efficient use of the transmission, the gears are switched to change the system's dynamic. In an automatic transmission, the occurrence of this commutation between states is automatically computed. Thus, an hybrid automaton is an appropriated model. The values of rotation speed used as threshold for switching between gears have been tuned with respect to transmission efficiency and speed [4].

The effects of the load and dynamic of the vehicle are supposed known with respect to the rotation speed of the engine shaft. This effect is represented by the function η named efficiency in gear i in the original paper (see equation 2 and figure 5). a_i represents the gear ratio. Here, the objective is no longer to find a set of guards. The simulation provides the behaviour of the transmission under the driver's solicitation.

$$\eta_i = 0.01 + 0.99\exp((\omega - a_i)^2/64) \tag{2}$$

2) The automaton:

a) Variables. : The load is imposed by the vehicle dynamic. Therefore, the state of the system is given by the rotation speed of the output shaft ω as well as the distance travelled by the vehicle. The latter is proportional to the angular position of the shaft θ.

b) Locations. : Each location corresponds to the combination of a gear ratio and the dynamic behaviour of the vehicle (whether it is accelerating or decelerating). The flow F in Gear i is :

$$(F) \begin{cases} \dot{\theta} = \omega \\ \dot{\omega} = \alpha\eta_i \end{cases} \quad \text{where } \alpha \in \{-1; 1\}$$

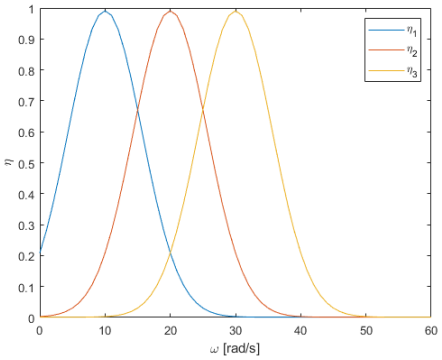

Fig. 5. Efficiency function. In a real system, this function would depend on the dynamic of the system, especially the input torque. In addition, this function is usually tabulated, whereas here it is defined by an analytic expression

The guards are calculated so that the efficiency of the transmission remains optimal. A delay of 5 seconds has been added for each transition to prevent the transmission to immediately switch between gears, which would be an unrealistic behaviour. Finally, the behaviour of the conductor is modelled by another automaton that switches between being accelerating and decelerating.

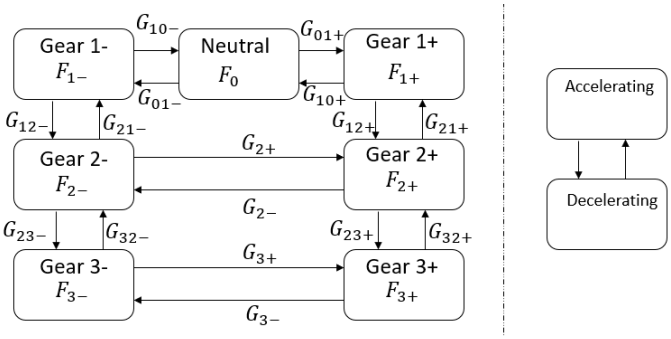

Fig. 6. simplified view of the automaton. The dot line symbolises the orthogonal product of the two subsystems

D. Inverted pendulum

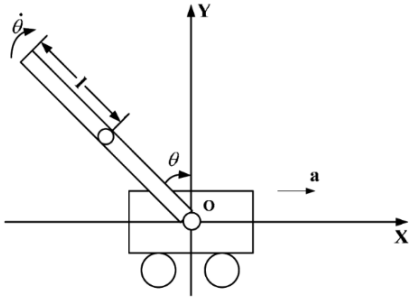

Fig. 7. Physical system (source : original paper). The x acceleration is the only controllable DOF.

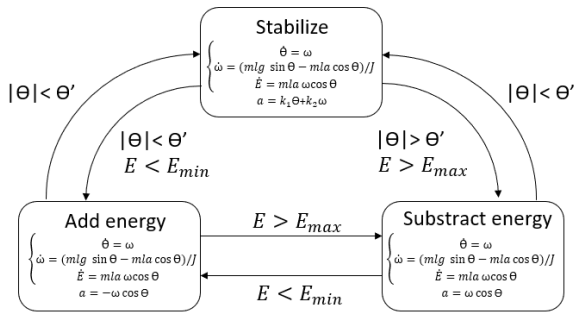

Fig. 8. simplified view of the automaton

1) Physical system: This system (see figure 7) comprises an inverted pendulum mounted on a chariot able to translate on one axis. The automaton simulates the control of the moving part so that the pendulum remains in an unstable state. From an energetic perspective, the pendulum has to follow three behaviour, depending on its need to add, subtract or maintain its mechanical energy []. The core idea in the original paper is that the sign of the derivative of mechanical energy can be controlled by choosing the acceleration of the platform a (see equation 3).

$$\begin{cases} a = -\omega \cos\theta & \text{will increase } E \\ a = \omega \cos\theta & \text{will decrease } E \\ a = k_1\theta + k_2\omega & \text{correction to remain in } \theta = 0 \text{ state} \end{cases} \tag{3}$$

2) The automaton:

a) Variables. : The state of the system is described by the set of three variables (θ, ω, E). θ is the angle between the pendulum and the vertical axis, ω is the angular velocity of the pendulum and E is the mechanical energy of the system.

b) Locations. : The system can be in three states. If the pendulum is near the expected angular position, the system is accelerated to remain in this position. However, if the system is not in the expected angular position, it can be in two states. In that case, the platform is accelerated to compensate for the lack or excess in energy.

E. Barrier

1) Physical system: The model is based on the communication between three different systems : the train, the controller and the barrier [3]. The train communicates its position to the controller and when it is estimated that the train is near enough to the intersection, the controller will command the barrier to close. When the controller estimate that the train has passed the intersection, it will command the barrier to reopen. Then the controller will communicate with another train and the cycle is repeated.

When a train is near enough of the intersection, a sensor measures the position of the train. When the train is far enough past the intersection, the sensor stop measuring its position. When the following train is in the vicinity of the intersection, its position is in turn measured by the sensor. The apparition of a train before the intersection is randomized and its speed

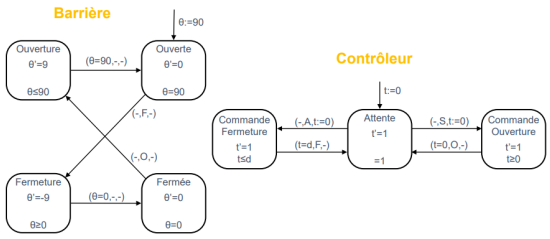

Fig. 9. Hybrid automaton modeling the barrier and the controller

is considered to be constant in each state. The closing speed of the barrier is also considered to be constant.

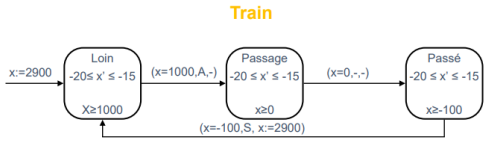

Fig. 10. Hybrid automaton modeling the trains

2) Automaton:

a) Variables: The automaton does not have input variable. The output of the system are the continuous variables represented in the array X :

$$X = \{x, y\}$$

x is the position of the train in the vicinity of the intersection, y represents the opening of the barrier.

b) Locations: The system is modeled by three automatons who are running in parallel. The automaton "Train" will generate a train whose speed is randomly generated. The locations of the automaton depend on the position of the train in comparison with the intersection. The locations of the automaton "Contrôleur" define the order to give to the barrier which will open or close depending on the position of the train. When the train is near the intersection, the automaton "Contrôleur" will be in the location "Commande Fermeture" which will impose the automaton "Barrière" to be in the state where the barrier will be undergoing closing. After the train is far enough, the automaton will generate a new train and reiterate the cycle. The automaton "Contrôleur" will be in the location "Commande Ouverture" which will impose the automaton "Barrière" to be in the state where the barrier is being opened.

F. Pump

1) Physical system: The system is constituted of two tanks connected through two canals named C_2 and C_3. The objective of the pump is to maintain the water level of second tank within a range [1]. The pump will supply or not the first tank depending on the water level in the second tank. After a while, the valve V4 is activated for a few minutes. The first tank will supply or retrieve water to the second tank depending on its water level position.

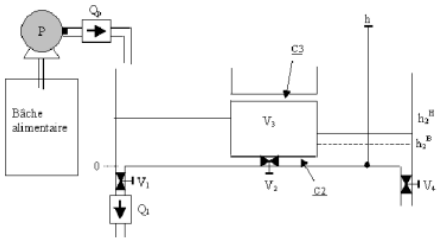

Fig. 11. Schema of the system (source : original paper)

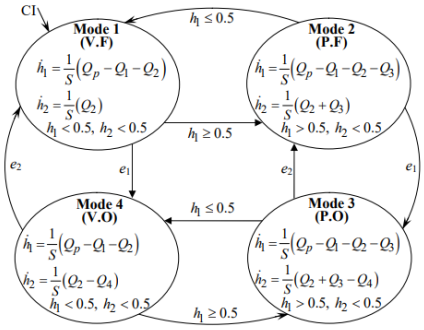

Fig. 12. Schema of the hybrid system model (source : original paper)

2) Automaton:

a) Variables: The automaton does not have input variable. The outputs of the system are the continuous variables represented in the array X :

$$X = \{h_1, h_2\}$$

h_1 is the water level in the first tank, h_2 is the water level in the second one.
In the model, the opening and the closing of the valve V_4 are modeled respectively by the events e_1 and e_2. However Stateflow have difficulties when it comes to generating and reading rising edges. Thus, in our model, e_1 and e_2 are modeled by a Boolean whose value will be 1 if the valve is open, 0 otherwise.

b) Parameters: The parameters are the dimensions of the tank and the characteristic of the liquid.

c) Locations: At first, the water level in the first and second tank is between the height of the canal C_3 and C_2. The locations depend on two factors : the state of the valve V_4 and the water level in the first tank in comparison with the height of the canal C_3.

G. Temperature regulation (complex)

1) Physical system:

At first the furnace will work in a similar way as the example of the . However this time, the possibility of the dysfunction of the PI controller is considered in the model [6]. During which, the temperature will begin to rise without any control. Then the dysfunction will be detected once the temperature exceeds a value fixed by the operator. The TOR controller will be thus activated imposing the temperature to be bounded between two values. However the TOR controller can present a dysfunction too, the furnace will then stop functioning. Each dysfunction can be repaired, the system will thus return to its nominal operating.

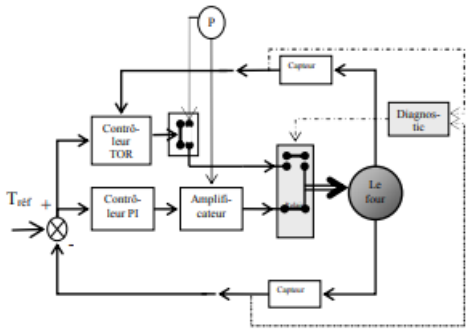

Fig. 13. Simplified schema of the system (Source : original paper)

2) The automaton:

a) Variables: The outputs of the system are the continuous variables represented in the array X :

$$X = \{T\}$$

T is the temperature of the furnace.
In the original model, the events r_C and r_T represents the reparation while d_C and d_T represents the dysfunctions. In our model, the dysfunctions are modeled by guards whose the validation is depends on variables which vary randomly. The reparation events r_C and r_T are modeled by the validation of a timed guard. The assumption here is that the detection and the reparation of the dysfunctions have always the same duration.

b) Parameters: The evolution of the temperature T is defined by the following equation :

$$K\dot{T} = T + U$$

U definite the final value of T, its value depends on the automaton state. K is a constant its value varies between locations.

c) Locations: The hybrid system has the five locations. Location 1 corresponds to the nominal operating. Location 2 corresponds to the state where a dysfunction occurred, but remains undetected. Location 3 and 4 correspond to the moment where the dysfunction is detected and the temperature evolution is supervised by the TOR controller. Location 5 corresponds to the dysfunction of the TOR controller.

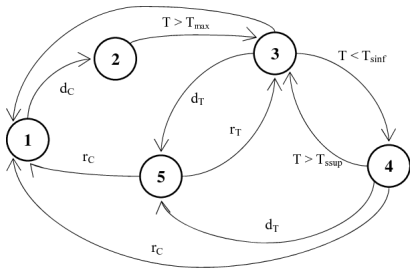

Fig. 14. Simplified state automaton of the furnace (Source : original paper)

V. Summary of examples

Table 15 is the compilation of all the collected automata classified with the criteria established previously. We used the following abbreviations :

- O.D.E. : Ordinary Differential Equation
- L : Linear
- NL : Non-Linear
- C : Continuous
- CD : Continuous and Discrete

Automaton	C1	C2	C3	C4	C5	C6
Bouncing ball	X	X		ODE L	C	
Billards	X	X		ODE L	C	
Lotka-Voltera	X	X		ODE L	C	
Automatic transmission				ODE NL	CD	X
Hexacopter	X			ODE NL	CDE	X
Temperature regulation (simple)		X		ODE L o1	C	
Temperature regulation (complex)				ODE L o1	CD	X
Water level regulation	X	X		ODE NL	CD	X
Inverted pendulum	X	X	X	ODE NL	C	
barrière de train			X	ODE L x'=c type	CD	X

Fig. 15. Classification of the created examples with respect to the criteria. The cell is crossed if the automaton satisfies the corresponding criterion. For the abbreviation, see part III

References

[1] Vincent Cocquempot. *Contribution à la surveillance des systèmes industriels complexes*. Habilitation à diriger des recherches, Université des Sciences et Technologie de Lille - Lille I, November 2004.

[2] David Harel. Statecharts: a visual formalism for complex systems. *Science of Computer Programming*, 8(3):231–274, June 1987.

[3] Thomas A. Henzinger. The Theory of Hybrid Automata. In M. Kemal Inan and Robert P. Kurshan, editors, *Verification of Digital and Hybrid Systems*, NATO ASI Series, pages 265–292. Springer, Berlin, Heidelberg, 2000.

[4] Susmit Jha, Sumit Gulwani, Sanjit A. Seshia, and Ashish Tiwari. Synthesizing switching logic for safety and dwell-time requirements. In *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, ICCPS '10, pages 22–31, New York, NY, USA, April 2010. Association for Computing Machinery.

[5] John Lygeros. Lecture notes on hybrid systems. In *Notes for an ENSIETA workshop*, 2004.

[6] Gabriel Antonio Perez Castaneda, Jean-François Aubry, and Nicolae Brinzei. Automate stochastique appliqué à l'évaluation de la fiabilité dynamique. In *7ème Conférence Internationale de Modélisation, Optimisation et Simulation des Systèmes, MOSIM 08*, page CDROM, Paris, France, March 2008.

[7] Jean-François Raskin. An Introduction to Hybrid Automata. In Dimitrios Hristu-Varsakelis and William S. Levine, editors, *Handbook of Networked and Embedded Control Systems*, Control Engineering, pages 491–517. Birkhäuser, Boston, MA, 2005.

[8] Fu-Hsuan Wen, Fu-Yuen Hsiao, and Jaw-Kuen Shiau. Analysis and Management of Motor Failures of Hexacopter in Hover. *Actuators*, 10(3):48, March 2021. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.