UNCERTAINTY IN HYBRID SYSTEMS AND ITS APPLICATION TO AIRCRAFT SYSTEMS

Emília Villani
Instituto Tecnonlógico de Aeronáutica, 12228-900 São José dos Campos, SP, Brazil
evillani@ita.br

Paulo Eigi Miyagi
University of São Paulo, Escola Politécnica, Av. Prof. Mello Moraes, 2231, São Paulo, SP, Brazil
pemiyagi@usp.br

Abstract. *This paper discusses how hybrid system modeling and analysis techniques can be used for estimating the reliability of mechatronics systems under the occurrence of failures. It introduces the modeling of uncertain and probabilistic events for the OO-DPT net, which combines Petri net, differential equation systems and object-oriented paradigm. It then presents its application for the case of aircraft systems, where safety issues are a major concern. The landing system of a military aircraft is used as a case-study aiming to estimate how sensor redundancy improve system safety.*

Keywords*: Petri nets, hybrid systems, fault analysis, reliability, safety systems*

1. Introduction

Mechatronics systems usually integrate components, equipment and technologies of different nature. Due to this diversity, they are frequently considered as *hybrid*, mainly in the systems theory point of view. A hybrid system combines characteristics of both Discrete Event Dynamic Systems and Continuous Variables Systems (Alla; David, 2004).

This paper discusses how hybrid system modeling and analysis techniques can be used for estimating the reliability of mechatronics systems under the occurrence of failures. The problem of failure modeling and analysis has already been approached by a number of works (Miyagi; Riascos, 2005). However, most of them consider the problem either from a discrete or continuous point of view.

The modeling formalism considered in this work is the Object-Oriented Differential Predicate Transition nets (OO-DPT net). It combines Petri net for the discrete part and differential equation systems for the continuous one. The object-oriented paradigm is incorporated in order to achieve modularity. The problem of design and validating control systems using OO-DPT nets has already been presented before (Villani; Miyagi; Valette, 2005). However, previous works have assumed that the system is operating under nominal conditions. No failure or uncertainty behavior is taken into consideration during the system analysis.

In this paper, the authors introduce the modeling of uncertain and probabilistic events for the OO-DPT nets and analyze its application for the case of aircraft systems, where safety issues are a major concern. It uses as a case-study the landing system of a military aircraft. This case-study has already been presented at previous work (Villani; Miyagi; Valette, 2003). By that time, OO-DPT net was used to model the system behavior without considering failures, uncertain behavior and component redundancies. In that case, the purpose was to formally verify a set of behavior properties for the nominal conditions of operation, based on the research of dangerous scenarios. In the case of this paper, the purpose is to analyze the system safety and reliability in the case of failure.

This paper is organized as following. Section 2 introduces the problem of safety analysis for aircraft system and discusses how a hybrid approach and the OO-DPT net can contribute to it. Section 3 describes the proposed approach and the modeling formalism. Section 4 describes the case-study and Section 5 presents some conclusion.

2. The Problem of Safety Analysis in Aircraft Systems

In aeronautics, one of the main concerns during the design of a system is safeness. For this purposes, most of the system components are provided with redundancy. The degree of redundancy of each *component* depends on a number of factors such as the kind of failure that can occur in a component and its probability, per hour of flight, and how the component failure affects the system operation and functionality.

Based on the component analysis, *system* failures should be considered. A system failure can be the result of the failure of a single component or a set of component. The maximum allowed probability of failure at a system level depends on how it deteriorates the level of flight quality and how it affects the aircraft operation. As an example, Table 1 presents the classification of aircraft systems into 3 groups according to the maximum probability of failure (Stevens; Lewis, 1992).

Due to the system complexity, the safety analysis is usually done without considering the dynamical behavior of the aircraft systems. However, this is an important aspect and can significantly influence the results of the safety analysis. Among the issues to be considered are:

- How the failure of a component affects the system behavior and how it can influence the probability of failure of other components.
- How to estimate the probability of critical scenarios that combine a set of component failures.
- How redundancy affects the system behavior. How failures are detected by control systems and how they are treated. How to estimate the probability of a wrong diagnostic and what are the consequences.

In order to answer these questions, a model of the aircraft system behavior should be built and analyzed. For this purposes, a number of aircraft systems are typically classified as hybrid. They incorporate continuous dynamics such as the continuous positioning of surfaces or the pressure evolution in a hydraulic system, as well as discrete sequence of events, such as switching between components in the case of failure, or executing the command sequences for landing and take-off. The OO-DPT nets can be then used to modeling the aircraft system and analyze the system reliability. The proposed approach is presented in the next section.

Table 1 – Maximum probability of failure in aircraft systems (Stevens; Lewis, 1992).

System group	Probability of failure, 1/h	Type of system	Principle of design
I	$<10^{-8}…10^{-9}$	Full authority, flight critical control systems	practically failure-free systems
II	$<10^{-5}…10^{-6}$	Important, no flight critical control systems	failure-safe systems
III	$<10^{-4}$	Auxiliary and comfort systems	failure-safe systems

3. The proposed approach

The analysis of the reliability of a mechatronics system can be organized in the following steps:

Step 1. Modeling of the system behavior under nominal conditions.
Step 2. System validation by verifying formal properties of the system model.
Step 3. Introduction of uncertain behavior and failure into the model.
Step 4. Determination of critical scenarios considering the uncertain behavior and failures.
Step 5. Estimation of probabilities using Monte Carlo simulation.

As presented before, Steps 1 and 2 has already been approached in previous works, they are briefly discussed in the next section. Step 3 is detailed in Section 3.2 and Steps 4 and 5 are discussed in Section 3.3.

3.1 Hybrid System Modeling based on Petri Net, Differential Equation System and OO- Paradigm

The modeling formalism has been introduced in (Villani, Miyagi, Valette, 2005). It is based on the incorporation of object-oriented concepts to the Differential Predicate-Transition Petri nets, proposed in (Champagnat et al, 1998).

Briefly, the model of a system is composed of the a set of 'n' classes (C_1, C_2, …, C_n). Each class C_i is modeled by a DPT net, which defines an interface between differential equation systems and Petri net elements. Its main features are:

- Each object of the class C_i is represented by a token in the DPT net of C_i.
- A set of variables (x_i) is associated with each token of the class C_i: they correspond to the attributes of the class.
- A differential equation system (F_{j_i}) is associated with each place (p_{j_i}): it defines the dynamic of the x_i associated with the tokens in p_{j_i}, according to the time (θ).
- An enabling function (e_{j_i}) is associated with each transition (t_{j_i}): it triggers the *firing* of the enabled transitions according to the value of the x_i associated with the tokens of the input places of t_{j_i}.
- A junction function (j_{j_i}) is associated with each transition (t_{j_i}): it defines the value x_i associated with the tokens of the output places of t_{j_i} after the transition firing.

The communication among objects can be discrete or continuous. The continuous interactions are modeled by sharing continuous variables among objects. The value of the shared variables is determined by one object and can be used in the junction function, the equation systems or the enabling function of other objects.

The discrete interactions are method calls. Each class offers methods that are associated with its transitions and that can be requested by other classes. A method call is modeled as the fusion of two transitions: the transition t_{j_i} of the class C_i that offers the method and the transition t_{w_v} of the class C_v that calls the method. The method call happens when both transitions are enabled in their classes.

As an example, Figure 1 presents the models of the classes *C_1 – Actuating Cylinder* and *C_2 – Discrete Sensor*. On the left side of each model, the Petri net describes the discrete behavior. The information associated with the continuous dynamics is on the right side. The time is represented as 'θ'.

In the case of class *C_1 – Actuating Cylinder*, the position of the hydraulic cylinder (variable 'x') varies from 0 to K_x, according to the pressure (external variable 'p_d') on the hydraulic circuit. When the method 'extend cylinder' is called the cylinder extends. The speed of the movement (variable 'v') depends on the pressure, the cylinder area (K_A) and the force against or in favor of the movement, which is considered as constant (K_F) and is associated with the load. Two auxiliary variable 'R' and 'E' indicates when the cylinder is completely retracted or extended.

In the case of the class *C_2 - Discrete Sensor*, it switches from ON (p_{1_2}) to OFF (p_{2_2}) according to the value of the external variable 'S_{in}'. If the an object of class C_2 is used to inform the position of the actuating cylinder, the variable 'S_{in}' could be associated either with 'E' or 'R'.

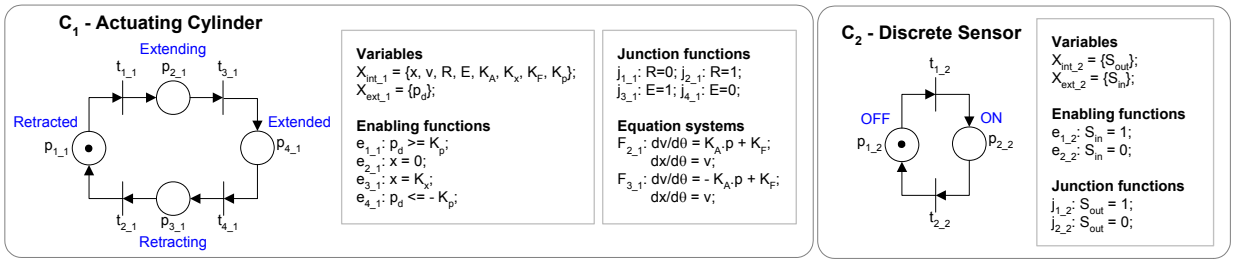

Figure 1. Model of classes *C_1 – Actuating Cylinder* and *C_2 – Discrete Sensor*.

3.2 Introduction of Uncertaint Behavior

Failures are uncertain events and are usually associated with probabilities. In order to analyze the system behavior under the occurrence of failures, probabilistic behavior should be introduced in the OO-DPT nets. The problem of modeling uncertainty in hybrid system has already been approached in many works of the literature (e.g. Pola et al. (2003)).

These works can be classified according to how the uncertainty is introduced into the models. Most of them consider one or more of the following cases:

- The continuous dynamic is modeled by using stochastic differential equations.
- The occurrence of discrete events is set according to probabilistic laws.
- After the occurrence of an event, the new state of the system is set according to probabilistic laws.

Another important point is the formalism used as a background. Most of the works already published are based on hybrid automata. Examples are (Bujorianu, Lygeros, 2003) and (Hespanha, 2004). Among the formalisms that model the discrete dynamic using Petri nets, there is the Fluid Stochastic Petri nets. It starts from the definition of Generalized Stochastic Petri nets and incorporate elements for the modeling of continuous dynamics, such as continuous places (Horton et al, 1996) and (Wolter, 2000).

The introduction of uncertainty into the models that merge Petri net and differential equation system is briefly approached in (Khalfaoui, 2003) and is also based on Generalized Stochastic Petri net. It considers that the dates of transition firings can be set according to stochastic distributions and, in the case of conflict between two or more transition, the decision can be made by associating a fixed probability to each transition.

This paper adopts a slightly modified definition, which aims to augment the modeling flexibility provided by the formalism. Instead of associating stochastic distributions to the dates of transition firings, we introduce *probabilistic junction functions* that set the value of the continuous variables after a transition firing according to probabilistic distributions (PD). No restriction is made on the kind of distribution that can be used. After the firing, these variables can then be used in enabling functions or equation systems, influencing both the discrete and continuous dynamics. An example is presented in Figure 2, where the probabilistic distribution PD_1 is used to determine the date of firing of transition t_2.

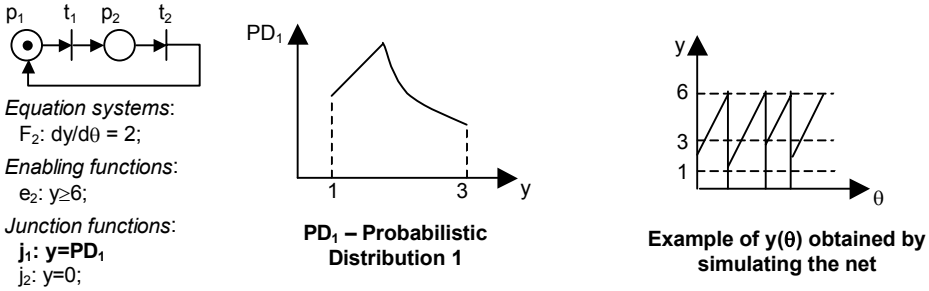

Figure 2. Example of probabilistic junction function.

In the case of conflict between transitions, probabilistic junction functions can be used to generate randon numbers that are used in the enabling function of the transitions under conflict, in order to choose the one that should fires. An example is presented in Figure 3. The probability of firing t_2 is 80%, in the remaining 20% of the cases, t_3 will be fired.

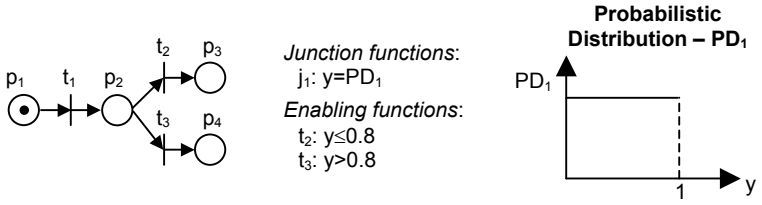

Figure 3. Example of probabilistic junction function for solving conflict.

In order to illustrate the application of probabilistic junction function to the modeling of failures in aircraft systems, a new probabilistic model of classes C_1 – *Actuating Cylinder* and C_2 – *Discrete Sensor* is presented in Figure 4. In the case of the actuating cylinder, the variable Δ_L, defined according to the probabilistic distribution PD_1, models the eventual leakage that retards the cylinder movement. In the case of the discrete sensor, two different kinds of failure are considered. The first one is when the sensor remains blocked in one of the positions even when the input signal (S_{in}) changes. It corresponds to the firing of t_{4_2} or t_{5_2}, and happens with a probability of $(1\text{-}P_3)$ and $(1\text{-}P_6)$, respectively. The second kind of failure occurs when the sensor looses the connection with the control system (firing of t_{7_2}). In this case, from an external point of view, the sensor is blocked at OFF. The date for the firing of t_{7_2} (θ_f) is set according to the probabilistic distribution PD_3.

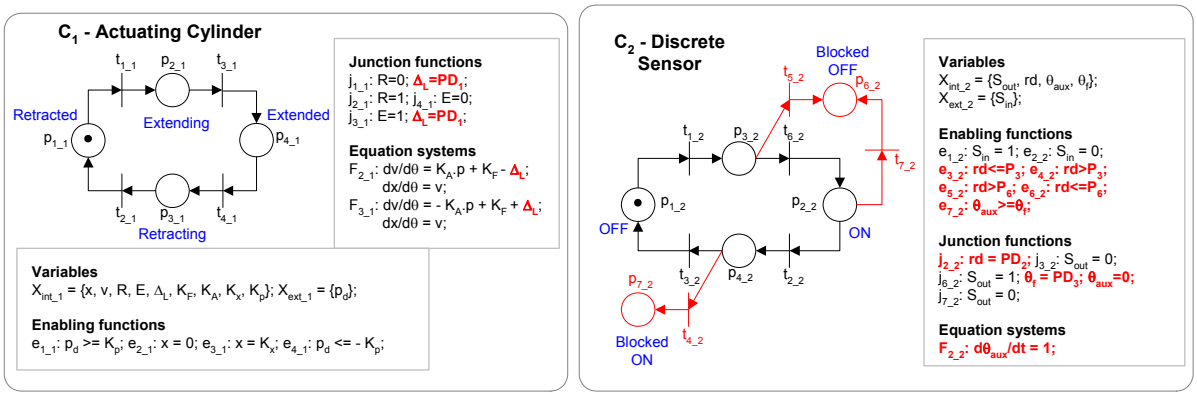

Figure 4. *Probabilistic* model of classes C_1 – *Actuating Cylinder* and C_2 – *Discrete Sensor*.

3.3. Model Analysis

Formal analysis techniques can be used for verifying behavior properties such as that dangerous states will never be reached or that certain sequence of events will always be executed under certain conditions. However, most of the time, these properties are only reasonable if we consider the system under nominal conditions of operation.

When failures are included into the model, the system will probably not obey the same set of properties and dangerous states may be reachable. An example is the case of component redundancy. No matter how many copies of a component we include into the system, if all them are subject to failure, then, from a formal point of view, the state where all the copies fail is a reachable state. The problem in this case is how to determine the scenarios that lead to dangerous states and how estimate their probability.

According to the model characteristic, formal techniques used for the verification of behavior properties may be applied for the research of dangerous scenarios. In this case, the probabilistic behavior is ignored and enabling function of the transitions associated with a probabilistic decision (such as t_{3_2}, t_{4_2}, t_{5_2} and t_{6_2} in the model of class C_2 - Figure 4) are considered constantly enabled. Once the dangerous scenarios are identified, their probability can be estimate by Monte Carlo simulation. As no restriction is imposed about the kind of probabilistic distributions that can be used, the application of formal techniques for the probability estimation of dangerous scenarios is impossible.

4. The case -study

The case-study considered in this paper is the landing-system of Rafale, a military airplane made by Dassault Aviation. It is composed of 3 landing sets (named as A, B and C) containing each one a door and a landing-gear. A simplified schema of a landing set is presented in Figure 5. The sequence that must be performed at landing consists of

opening the doors of the 3 landing-gear compartments, extending the landing-gears and closing the doors. A similar sequence must be performed at take-off.

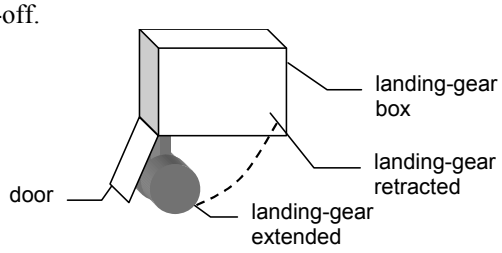

Figure 5. Landing set.

The landing-gear and door movement is performed by a set of actuating hydraulic cylinders. For each door, a hydraulic cylinder opens and closes the door. For each landing gear, a hydraulic cylinder extends and retracts the landing gear. The hydraulic cylinders are moved by a set of electro-valves. Furthermore, discrete sensors inform the control system about the positions of the actuating cylinders, the pressure in the hydraulic system, among others.

The system components are provided with different degree of redundancy. The computer that performs the landing sequence and processes the pilot commands, is provided with redundancy 2 (there are 2 computer concurrently processing signals and commands of the landing systems). Each discrete sensor is provided with redundancy 3.

Sensors signal are used for coordinating the landing system movements during landing and take-off. Furthermore, during cruise and other flight phases, they are constantly read and processed with a certain frequency in order to monitor the landing system and detect problems. In the case of failure the pilot is notified.

The focus of this case-study is on analyzing how the sensor redundancy augments the system safety. Because of the limited space, only the sensors of the landing-gear actuating cylinder are considered in this paper.

Each landing gear has two different kinds of sensor, each of them with redundancy 3:

- Landing-Gear Extended – LGE.
- Landing-Gear Retracted – LGR.

In this text, in order to unambiguously refer to a sensor, the notation LGE_{xy} is used to refer to the LGE sensor 'x' (1, 2 or 3) of the landing-gear 'y' (A, B or C). Similarly, LGR_{xy} is used to identify the LGR sensors.

The signal from the landing-gear and door sensors are processed in the following way:

Level 1) For each set of three redundant sensors, the output signals are compared among them and a combined sensor output is generated. It could be on, off or under failure.

Level 2) For each landing-gear the combined sensor output of LGEs is compared with the combined sensor output of LGRs and a landing-gear output is provided. It indicates the current state of the landing-gear, which can be extended, retracted, moving or under failure.

Level 3) The same approach is performed for each door. A door output is generated and indicates the current state of each door, which can be open, closed, moving or under failure.

Level 4) For each landing set, the landing-gear output is compared with the door output and a landing-set output is generated. Failures are detected in situations such as if the door is closed and the landing-gear is moving.

Level 5) The three landing-set output are compared among them and a landing-system output is generated. As the same set of valve simultaneously affect all the three doors (or landing-gears), the movement of doors and landing-gears is synchronized in some points. An example is the command to extend the landing-gear is emitted only when the three doors are completely open. If the system detects a situation when the landing-gear and door are moving, a failure is detected.

Once that this example presented in this paper does not encompass the door sensors, only Levels 1 and 2 are considered. The strategy adopted for processing the set of three redundant sensors at Level 1 is:

- If all the three output signals are 'ON', the combined sensor output is 'ON'.
- If all the three output signals are 'OFF', the combined sensor output is 'OFF'.
- If two output signals are 'OFF', and one is 'ON', the combined sensor output is 'OFF' and the identity of the sensor with output 'ON' is memorized. If on the next time the sensors are read, the output of this sensor is still different from the other two, this sensor is considered as fault, and from this moment on it is ignored by the control system.
- A similar approach is executed when two output signals are 'ON', and one is 'OFF'.
- If one sensor has been eliminated and the other two are 'OFF', the combined sensor output is 'OFF'.
- If one sensor has been eliminated and the other two are 'ON', the combined sensor output is 'ON'.
- If one sensor has been eliminated and the other two sensor signals are different from each other, the combined sensor output remains unchanged and an error is memorized. If on the next time the sensors are read, the two outputs are still different, the combined sensor output is 'under failure'.

At Level 2 the strategy is:

- If the LGE combined sensor output is ‘ON’ and the LGR is ‘OFF’ the landing-gear output is ‘extended’.
- If the LGR output is ‘ON’ and the LGE is ‘OFF’ the landing-gear output is ‘retracted’.
- If the both LGR and LGE output are ‘OFF’ the landing-gear output is ‘moving’.
- If the both LGR and LGE output are ‘ON’ the landing-gear output is ‘under failure’.
- If the LGR output is ‘ON’ and LGE output is ‘under failure’, the landing-gear output is ‘retracted’.
- If the LGE output is ‘ON’ and LGR output is ‘under failure’, the landing-gear output is ‘extended’.
- If the LGR output is ‘OFF’ and LGE output is ‘under failure’, the landing-gear output is ‘under failure’.
- If the LGE output is ‘OFF’ and LGR output is ‘under failure’, the landing-gear output is ‘under failure’.

The modeling of the landing system is presented in the next section.

4.1. System Modelling

The system model is composed of a set of 6 classes: *C_1 - Actuating Cylinder*, *C_2 - Discrete Sensor*, *C_3 - Hydraulic System*, *C_4 – Sensor Level_1*, *C_5 – Sensor Level_2*, *C_6 - Landing-gear Controller*. The first 3 classes model the behavior of the physical components, while classes C_4, C_5 and C_6 models the landing-gear control system.

The model of class C_1 and C_2 has already been presented in Figure 4. There are three objects of class C_1: $O_{1.1}$ - Landing-Gear A, $O_{2.1}$ - Landing-Gear B, $O_{3.1}$ - Landing-Gear C, and 18 objects of class C_2: $O_{1.2}$ – LGE_{1A}, $O_{2.2}$ – LGE_{2A}, $O_{3.2}$ – LGE_{3A}, $O_{4.2}$ – LGE_{1B}, $O_{5.2}$ – LGE_{2B}, $O_{6.2}$ – LGE_{3B}, $O_{7.2}$ – LGE_{1C}, $O_{8.2}$ – LGE_{2C}, $O_{9.2}$ – LGE_{3C}, $O_{10.2}$ – LGR_{1A}, $O_{11.2}$ – LGR_{2A}, $O_{12.2}$ – LGR_{3A}, $O_{13.2}$ – LGR_{1B}, $O_{14.2}$ – LGR_{2B}, $O_{15.2}$ – LGR_{3B}, $O_{16.2}$ – LGR_{1C}, $O_{17.2}$ – LGR_{2C}, $O_{18.2}$ – LGR_{3C}.

Model of Class C_3 – Hydraulic System

This class models the dynamics of the hydraulic pressure in the landing-gear model. The OO-DPT net of the class is presented in Figure 6. According to the position of the electro-valves the pressure in the hydraulic circuit (p_d) increases or decreases. The time necessary for increasing and decreasing the pressure is subjected to uncertainties that are represented by variable Δ, specified according to a probabilistic distribution PD_3. There is only one object of this class: $O_{1.3}$ – Hydraulic System.

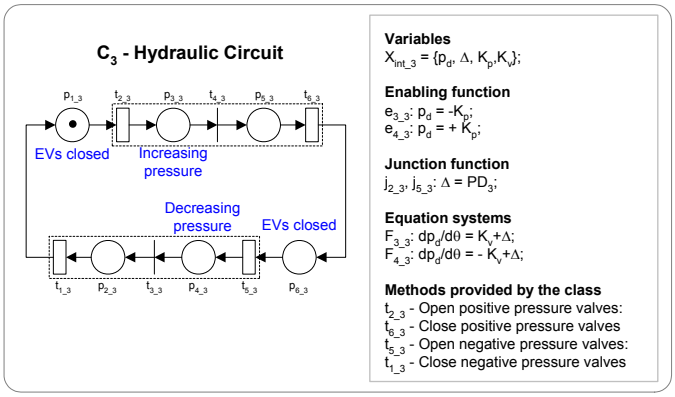

Figure 6. Model of class *C_3 – Hydraulic System*.

Model of Class C_4 – Sensor Level_1

This class models the processing of sensor outputs at Level 1. The OO-DPT net of the class is partially presented in Figure 7. Part of the model has been omitted in order to simplify this presentation. Basically, the current state of the set of sensors is represented by places p_{1_4} to p_{8_4}, they indicate if the set is under failure, if any error has been memorized and if any sensor has been discharged. In the full model of the class, these places are connected with transitions that have been omitted from Figure 7. The sensors input are the external variables S_{in_1}, S_{in_2} and S_{in_3}. For each object of class C_4, S_{in_1}, S_{in_2} and S_{in_3} are associated with the variable S_{out} of three objects of class *C_2 – Discrete Sensor*.

Considering for example the case when the last state of the set was ‘under failure’ ($p_{1\_4} = 1$). In this case, when the object is requested to process the sensor signals (firing of t_{1_4}), only transition t_{2_4} will be enabled and the new output will also be ‘under failure’ (firing of t_{4_4}). If the previous state was operation with 2 sensors (firing of t_{3_4}), with Sensor 1 under failure (firing of t_{5_4}), and the current input signals from Sensors 2 and 3 are different (enabling function of e_{6_4} is true), then the new output of the set of sensors will be ‘under failure’ (firing of f_{6_4}). In a similar way, other transitions not include in Figure 7 process all the other combinations of previous state and current sensor signals.

There are 6 objects of this class: $O_{1.4}$ – LGE Output_A, $O_{2.4}$ – LGE Output_B, $O_{3.4}$ – LGE Output_C, $O_{4.4}$ – LGR Output_A, $O_{5.4}$ – LGR Output_B, $O_{6.4}$ – LGR Output_C.

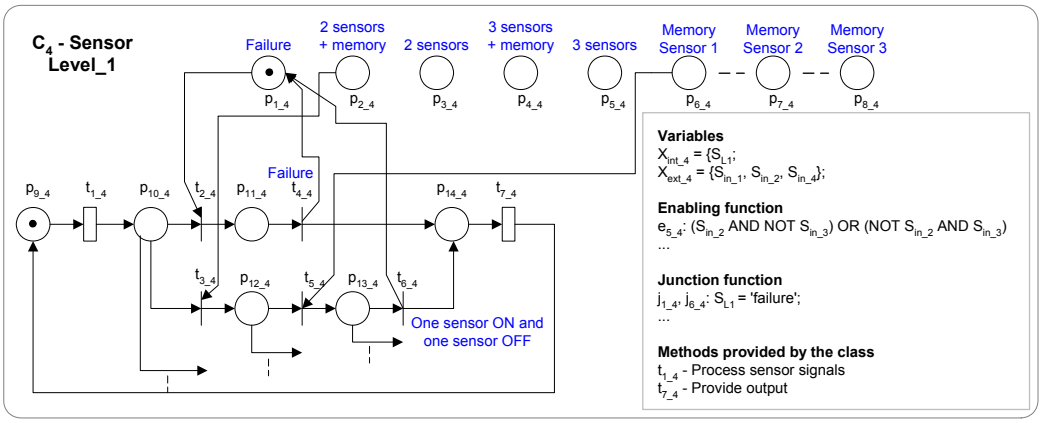

Figure 7. Model of class *C4 – Sensor Level_1*.

Model of Class C_5 – Sensor Level_2

The model of this class is similar to the model of class C_4, but instead of processing the sensor signals, this class processes the output of the objects of class C_5 (variable S_{L1}). This process is executed every time interval of K_F. It initiates by calling the method provided by the objects of class C_4 (associated with transition t_{1_4}) to update the value of S_{L1}. It then determines the current state of the landing-gear and stores the result in the variable S_{L2}, which is then used by the object of class C_6, among others. There are 3 objects of this class: $O_{1.5}$ – Landing-gear Output_A, $O_{2.5}$ – Landing-gear Output_B, $O_{3.5}$ – Landing-gear Output_C.

Model of Class C_6 - Landing-gear Controller

This class controls the operation of the electro-valve and the landing-gear extension and retraction according to commands emitted by the pilot. It is important to observe that it uses the output of the objects of class C_5 to detect if the landing-gear is extended or not. If after a time of K_{θ_max} the command has not been executed, a failure is detected (firing of t_{11_6} or t_{8_6}).

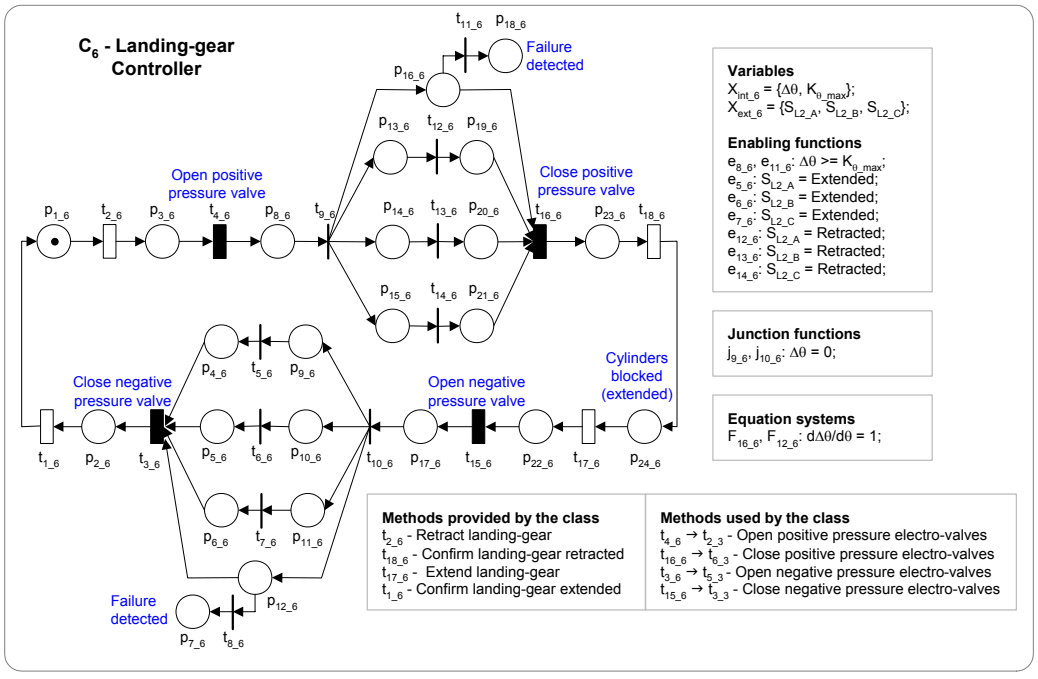

Figure 8. Model of class *C4 – Sensor Level_1*.

4.2. Model Analysis

The first activity is to determine the dangerous situations. Examples are situations that may result in a collision between the landing gear and the door. This is the case when the landing-gear is considered as retracted (firing of t_{1_6}) but is actually still moving (token in place p_{3_1}) or extended (token in p_{4_1}). In this situation a command will be emitted to close the doors and may damage the landing-gears.

Another example is when landing-gears are not retracted during cruise. This is the case when the doors are not considered as open because of sensor failures and, in order to avoid a potential shock, the landing-gear cannot be retract. The landing-gear extended during cruise augment the drag, deteriorating the aircraft aerodynamic. However, this situation is considered less dangerous than the previous one.

The second activity is the research of the scenarios that lead to these situations. For this purposes, techniques are current under development. In order to estimate the scenarios probabilities, Monte Carlo simulation must be used. For the moment there is no simulator available for the OO-DPT nets. In order to simulate, each class model is converted to a programming language, such as a MatLab subroutine, using a token-player approach. Once that the failure probability of the components are extremely small, the number of simulations that are necessary for obtaining reliable results is very large. Techniques are current under study to reduce this number.

Once this analysis has been completed, it will be possible to compare different strategies for processing the sensor signals. Weight can also be associated to each dangerous situation in order to provide more detailed results. A strategy with a probability P_1 of flying with the landing-gear extended may be considered better than a strategy that presents a probability P_2 of the collision between door and landing-gear even if P_1 is greater than P_2.

5. Conclusions

This paper presents the application of a hybrid approach for the reliability analysis of mechatronics systems. For this purposes, the modeling of uncertainty and probabilistic events is introduced to the OO-DPT net. Among the problems that motivate this work is the analysis of safety issues in aircraft systems. In this context, the approach is applied to the landing system of a military aircraft. As an example the problem of analyzing sensor redundancy and compare control strategies is detailed. Results are currently obtained by Monte Carlo simulation and translating the OO-DPT net to MatLab functions. The development of simulation tools is among the future works. Another important point is the proposal of techniques for determining critical scenarios and reducing the number of simulations.

6. Acknowledgements

This work received financial support of the governmental agencies FAPESP, CNPq and CAPES. Particularly, the authors would like to thank the Kyatera/TIDIA Program, under which the work is developed.

7. References

Alla, H.; David, R., 2004, *Discrete, Continuous, and Hybrid Petri Nets*. Springler Verlag.

Bujoriani, M.L.; Lygeros, J. et al., 2003, "Stochastic hybrid models: an overview". *12th Mediterranean Conference on Control and Automation (MED)*, Kusadasi.

Champagnat, R. et al, 1998, "Modelling and Simulation of a Hybrid System through Pr/Tr PN-DAE Model", *3rd International Conference on Automation of Mixed Processes*, Reims.

Hespanha, J.P., 2004, "Stochastic Hybrid Systems: Application to Communication Networks". *Hybrid Systems: Computational and Control (HSCC)*, Philadelphia.

Horton, G. et al, 1996, "Fluid Stochastic Petri Nets: Theory, Applications and Solution", *NASA CR-198274 ICASE Report No. 96-5*, Inst. for Computer Applications in Science and Eng., NASA Langley Research Center, Hampton.

Khalfaoui, S., 2003, "Méthode de Recherche des Scénarios Redoutés pour l'Evaluation de la Sûreté de Fonctionnement des Systèmes Mechatroniques du Monde Automobile", PhD Thesis, *Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS*, Toulouse.

Pola, G. et al., 2003, "Stochastic hybrid models: an overview". *IFAC Conference on Analysis and Design of Hybrid System (ADHS)*, St Malo.

Stevens, B.L.; Lewis, F.L., 1992, *Aircraft control and simulation*. Jonh Wiley & Son.

Miyagi, P.E.; Riascos, L.A., 2005, "Modeling and analysis of fault-tolerant systems for machining operations based on Petri nets", Control Engineering Practice, *In Press*, Elsevier.

Wolter, K., 2000, "Modelling Hybrid System with Fluid Stochastic Petri Net", *Proc. 4th Int. Conf. on Automation of Mixed Processes: Hybrid Dynamic Systems*, Dortmund.

Villani, E.; Miyagi, P.E.; Valette, R., 2003, "Petri Net and OO for the modular analysis of aN AirCraft landing system", 17th International Congress of Mechanical Engineering, São Paulo.

Villani, E.; Miyagi, P.E.; Valette, R., 2005, "A Petri-Net based Object-Oriented Approach for the Modelling of Hybrid Productive Systems", *Non-linear Analysis*: *Theory and Methods*, Elsevier, *Accepted to pubblication*.

8. Responsibility notice

The authors are the only responsible for the printed material included in this paper.