Complex System Prognostics : a New Systemic Approach

Flavien Peysson [1], Mustapha Ouladsine [1], and Rachid Outbib [1]

[1] *LSIS Laboratory of System and Information Sciences, UMR CNRS 6168*
University of Paul Cézanne, Aix-Marseille III - 13397 Marseille Cedex 20, France
flavien.peysson@lsis.org, mustapha.ouladsine@lsis.org, rachid.outbib@lsis.org

ABSTRACT

Profitability and rentability are two key features for industrial companies that exploit complex engineered systems. One way to improve these features is the maintenance. Indeed, companies need to keep and improve equipments availability while reducing the maintenance costs. The maintenance optimization is now more than ever an industrial concern. The goal is to avoid failure and to have the right equipment with the right person at the right moment, at the right place. In the Prognostics and Health Management cycle, a prognostic function is used to predict the future system damage states in order to improve the maintenance plan. This paper addresses the prognostic domain by presenting a generic framework for prognostic. This framework allows to make a prediction of the system damage state by taking into account how and where the system will be used. The framework is described by a specific formalism and methodology to analyze the system damage dynamic of elementary resources and to trace the subsystem and system damage state according to the system structure. The framework is based on the system decomposition according to three levels: Environment, Mission, Process. This paper introduces the maintenance plan and a systemic view in the framework.

1 INTRODUCTION

Maintenance optimization consists to find the right balance between preventive and corrective maintenance while respecting an objective set in term of productivity and profitability. Maintenance action dates are then computed in order to optimize one criterion that can be the maintenance costs, the equipment availability, the safety or a compromise between the three.

Figure 1 depicts the induced costs by the maintenance and the failure of systems. The green line is the global maintenance costs according to the observed number of failure occurrence on the system. This means that if equipments are often maintained, there will be few failures but lot of money is needed. To the contrary, if equipments are never maintained few financial resources are needed but a lot of failures will be observed. It seems clear that the failure costs, represented by the red line, are inversely proportional to the maintenance costs. Indeed the unspent money will be used for the restoration actions on the system. Moreover, the system will be unavailable. The sum of the maintenance costs, given by the blue line, represents the total costs to maintain a system in operation. The optimal maintenance is a maintenance that minimizes the routine maintenance costs and costs associated to restoration actions after failure. One way to have an optimal maintenance policy is to use an automated aid system for the maintenance in order to identify the equipments to maintain and to know when the maintenance needs to be do.

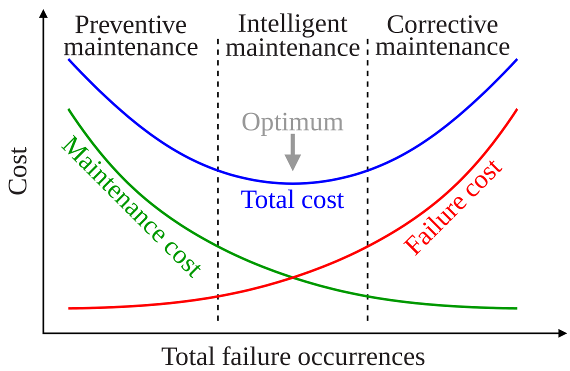


Figure 1: The maintenance costs

From this first analysis, it is clear that there are a growing interest in the intelligent maintenance where the monitoring has a fundamental part (Racoceanu, 2006). Condition Based Maintenance (CBM) uses real-time information to evaluate the damage state of a system and to know if there needs to a maintenance action. To extend CBM, Prognostics and Health Management (PHM) techniques have emerged to predict the evolution of the system damage state (Vachtsevanos *et al.*, 2006). PHM is a system engineering dis-

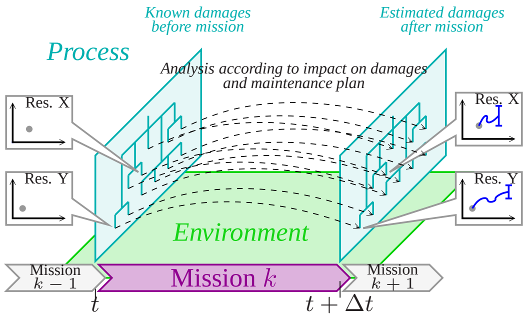


Figure 2: Prognostic process

cipline focusing on detection, prediction, health management of complex system.

Various prognostic approaches have been developed ranging from a simple historical failure rate models to a complex physics-based model. (Byington *et al.*, 2003) and (Lebold and Thurston, 2001) have classified these approaches according to their applicability on complex systems and their economic viability. The three main classes are: model based, data driven and experienced based approaches. Most works in literature are on damage indicator evolution, where the damage indicator is an image of the health indicator of a system. More details and references on the review of prognostic approaches in the literature can be found in (Peysson *et al.*, 2008b).

This paper addresses the prognostic domain of the PHM discipline by presenting a generic framework for prognostics. Formalism and methodology for prognostics are detailed in section 2. Then a modeling for maintenance plan is presented in section 3. Finally how to estimate the damage of a complex system from basic equipment damage is discussed in section 4.

2 PROGNOSTIC FRAMEWORK

The implementation of an intelligent maintenance policy requires the formalization of a prognostic process that are able to predict the evolution of the damage state of a complex system according to the operational and environmental conditions to which the system is submit and to maintenance operation plan. Figure 2 depicts the proposed prognostic process over the Mission k of duration Δt. Prognostics of the damage variables is based on the study of their dynamics defined from damage behavioral model.

To provide a start of answer of the prognostic problematic, we have introduced in a first time a formalism to describe a complex system and in a second time we have designed a methodology to analyze and to predict the system damage dynamics i.e. damage trajectories. Our approach is based on a system description according to three levels (Peysson *et al.*, 2009b). Predictions are performed via a sequence of known mission parameters, and environmental conditions. This allows for mission and maintenance planning by taking into account the predicted system damages over time.

2.1 Formalism for prognostic

This formalism allows to describe a complex system $\mathcal{S}$ in order to analyze its temporal trajectories of its damage state over a mission and thus to predict the mission success. We defined the system $\mathcal{S}$ as:

$$\mathcal{S} = \langle\, \mathcal{P},\ \mathcal{E},\ \mathcal{M} \,\rangle \qquad (1)$$

where $\mathcal{P}$ is the *Process* level that gives means to accomplish a mission, $\mathcal{E}$ is the *Environment* level that represents areas where the mission is accomplished, and $\mathcal{M}$ is the *Mission* level that defines the use of the system during a time period. Figure 3 is an overview of the proposed generic prognostic framework. The three levels of description are depicted by the Venn diagram on the top of the figure.

One of the main goals of the proposed formalism is to model the influence of the mission and of the environment on the damage evolution of the process. Indeed, in the real world there are some exchange between these three levels like the pollution between the process and the environment. Even if these exchanges can be important, they do not interest us in term of damage prognostics. As our goal is to prognostic the system damage we kept in the formalism only the bidirectional exchange between mission and process because how the system is used impact its damage dynamic and system damage state is a determining factor for the mission progress. We also kept the unidirectional exchange between environment and process because environmental conditions where the system evolves also impact the damage dynamics.

Another main advantages of the formalism is the genericity. This formalism are completely independent of the system nature. In a prognostic goal, an electronic card will have the same model structure as an actuator or a Diesel engine. Universal models that will be used to analyze the damage dynamics.

Process level

The process $\mathcal{P}$ is decomposed by a hierarchical way in order to obtain basic equipments. These equipments are called resources and are deteriorated in use. Thus, resources are equipements for which the damage state must be predicted. Resources correspond to the leaves of the process tree, cf. section 4. Process $\mathcal{P}$ is defined by:

$$\mathcal{P} = \langle\, SP_r,\ \mathcal{R},\ \mathcal{P}_S,\ \mathcal{B} \,\rangle \qquad (2)$$

where SP_r is the root sub-process of the process tree, $\mathcal{R}$ is the set of the system resources, $\mathcal{P}_S$ is the system sub-process set and $\mathcal{B}$ is the set of structural relation between the element of $\mathcal{R}$ and $\mathcal{P}_S$.

Resources are identified from the functional description and maintenance actions of $\mathcal{S}$. Indeed, it is not useful to analyze the damage trajectory of a engine part, if in case of failure, the complete engine is replaced. A resource $r^i \in \mathcal{R}$ is characterized by the 7-tuple:

$$r^i = \langle\, \tau_{r^i},\ \delta_{r^i},\ \mathcal{U}_{r^i},\ \mathcal{X}_{r^i},\ \Phi_{r^i},\ \mathcal{D}_{r^i},\ \Psi_{r^i} \,\rangle \qquad (3)$$

where τ_{r^i} represents the operating time of r^i because all resources are not used in the same time on a complex system, δ_{r^i} corresponds to the damage state, this is a damage feature that evolves between 0 and 1 when it reaches one r^i is considered as unavailable. $\mathcal{U}_{r^i}$ is the operating profile set. An operating profile defines

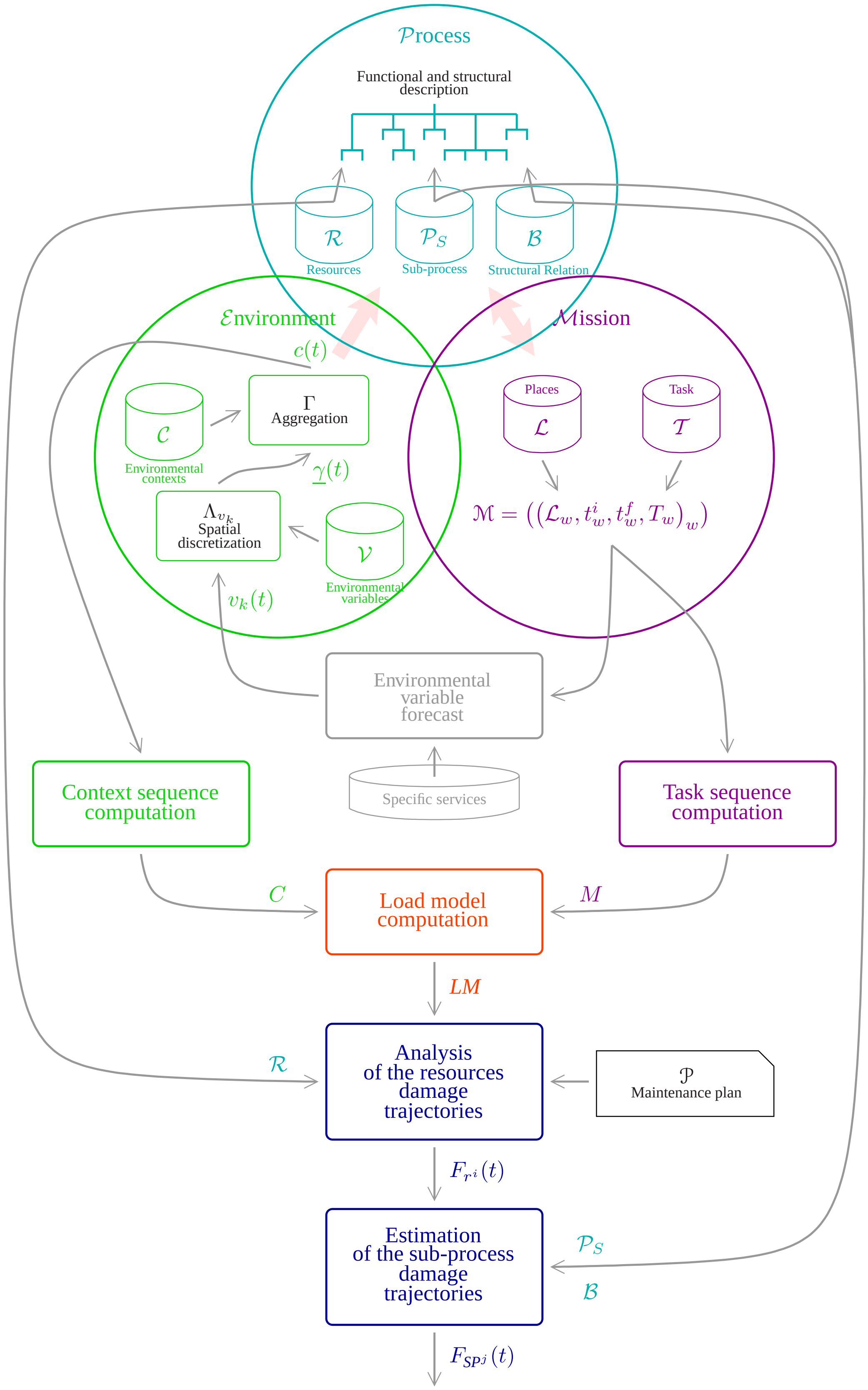


Figure 3: Damage trajectory based prognostic framework for a complex system $\mathcal{S}$

a constant solicitation constraints imposed to the resource. $\mathcal{U}_{r^i}$ is given by a space discretization of operating variable set $\mathcal{X}_{r^i}$:

$$\Phi_{r^i} \quad : \quad \mathcal{X}_{r^i} \quad \longrightarrow \quad \mathcal{U}_{r_i} \tag{4}$$

$\mathcal{D}_{r^i}$ is the damage behavioral model set. Behavioral models can be in various form as: differential equations, stochastic automata, damage abacus… A damage model defines the damage dynamic for a given operating mode of r^i. In working, the appropriate damage model is given by the function Ψ_{r^i}:

$$\Psi_{r^i} \quad : \quad \mathcal{U}_{r^i} \quad \longrightarrow \quad \mathcal{D}_{r_i} \tag{5}$$

$\mathcal{P}_S$ and $\mathcal{B}$ are detailed in section 4.

Mission level

The mission level $\mathcal{M}$ characterizes the working of $\mathcal{S}$ during a finite time period. $\mathcal{M}$ is given by the 3-tuple:

$$\mathcal{M} = \langle\, \mathcal{L},\ \mathcal{T},\ \mathfrak{M}\,\rangle \tag{6}$$

where $\mathcal{L}$ is a set of known places where $\mathcal{S}$ can operate, $\mathcal{T}$ is the set of tasks that $\mathcal{S}$ can accomplished and $\mathfrak{M}$ is a specific mission i.e. a dated sequence of known tasks in known places. $\mathfrak{M}$ is defined by:

$$\begin{cases} \mathfrak{M} = \left(\left(T_j,\ t^i_j,\ t^f_j,\ \mathcal{L}_j \right)_j \right) \\ j \in \mathbb{N}^*,\ \forall j > 1, t^i_j \geq t^f_{j-1},\ \mathcal{L}_j \subseteq \mathcal{L} \end{cases} \tag{7}$$

with T_j a task, t^i_j and t^f_j respectively the start and end dates of the task T_j. $\mathcal{L}_j$ is the set of place where T_j is realized.

A task T_j is a list of resources r_k associated with an operating profile u_k. T_j is defined by:

$$T^j = \{\ (r_k, u_k)_k\ \},\ k \in \mathbb{N},\ r_k \in \mathcal{R},\ u_k \in \mathcal{U}_{r_k} \tag{8}$$

Environment level

The environment describes the conditions where the process is working. These conditions are independents of the process solicitation. The environment represents meteorological, climatical phenomena… The goal of this level is to create a feature called environmental context that characterizes the environment impact on system damage dynamic. The environment is defined by:

$$\mathcal{E} = \langle\, \mathcal{V},\ \mathcal{G},\ \Gamma,\ \mathcal{C}\,\rangle \tag{9}$$

where $\mathcal{V}$ is the set of characteristic variables of the environment, $\mathcal{G}$ the combination set of the environmental impact features computed for each environmental variable and Γ the passage function from $\mathcal{G}$ to $\mathcal{C}$, the set of the environmental context. For a given environmental context, the constraints impose to $\mathcal{S}$ by the environment is considered as constants.

$$\Gamma \quad : \quad \mathcal{G} \quad \longrightarrow \quad \mathcal{C} \tag{10}$$

Γ is the aggregation block on figure 3.

To model the impact on damage dynamic of each variable $v_k \in \mathcal{V}$. A environmental variable is characterized by:

$$v_k = (\ v_k(t),\ \mathcal{I}_{v_k},\ \rho_{v_k},\ \Lambda_{v_k}\ ) \tag{11}$$

where $v_k(t)$ is the value time series of v_k, $\mathcal{I}_{v_k}$ is its definition domain, ρ_{v_k} its number of impact degree and Λ_{v_k} its space discretization function according to its impact on damage dynamic.

More information on each level are available in (Peysson *et al.*, 2008a), (Peysson *et al.*, 2008b) and (Peysson *et al.*, 2009a).

2.2 Damage trajectories

The damage trajectories prediction is made by simulation of the previously obtained model for a mission from the initial state of resources, tasks to accomplish, environmental forecasts and the maintenance plan. The simulation is based on the analysis of the resource damage evolution i.e. their temporal damage trajectory. The prognostics is thus the damage state of the model at the end of the simulation. As uncertainty is central to any prognostic definition, the prognostic result for each element is given by a interval that represents its possible damage state.

In section 2.1, we established the description formalism of a complex system in order to prognose its damage trajectories. The prognostic methodology principle is depicted on the bottom of the figure 3. The methodology goal is to make a piecewise analysis to built the damage trajectories F_q.

$$\begin{cases} F_q \quad : \quad \left[t^i, t^f\right[\quad \longrightarrow \quad [0,1]^2 \\ \qquad\qquad t \qquad \longmapsto \quad \begin{pmatrix} F_q^+(t) \\ F_q^-(t) \end{pmatrix} \\ q \in \mathcal{R} \cup \mathcal{P}_S \end{cases} \tag{12}$$

where $\left[t^i, t^f\right[$ is the mission time interval. F_q^+ and F_q^- are respectively the fast and slow damage trajectories of the element q. They represent the extreme trajectories that the damage of q could track i.e. that all the possible damage trajectories of q are between F_q^+ and F_q^-. The prognostic methodology is decomposed in three steps.

Load model computation

The first step is the construction of the load model *LM* that characterizes the sequence of operating modes of the system $\mathcal{S}$ during the mission $\mathfrak{M}$. An operating mode *OM* is defined as a constant constraint imposed to $\mathcal{S}$ i.e. by the couple:

$$OM = (\ T,\ c\ ),\ T \in \mathcal{T},\ c \in \mathcal{C} \tag{13}$$

LM is then given by:

$$LM = (\ (\ OM_k,\ d_k\ )_k\ ),\ k \in \mathbb{N}^*,\ d_k \in \mathbb{R}^*_+ \tag{14}$$

with d_k the duration of the operating mode OM_k. Before the *LM* computation, the timed sequences M and C respectively of tasks and contexts need to be compute from $\mathfrak{M}$ (Peysson *et al.*, 2008b).

Resource damage trajectories analysis

The next step of the prognostic is to analyze all the ressource trajectories according the load model *LM*. On each OM_k the adequate damage model for each ressource is simulated during a time of d_k. The ressource damage state at the end of the OM_{k-1} is used as the initial condition for the analyze of OM_k. In this analysis, the maintenance plan $\mathcal{P}$ is also taken into account, cf. section 3.

This step output is the functions $F_r(t)$ with $r \in \mathcal{R}$.

Sub-process damage trajectories estimation

The last step allows to estimate the damage evolution of sub-process from the structural relation between resources and/or sub-process. This means that we have a systemic approach of the damage evolution.

This step is detailed in section 4 , its output is the functions $F_{SP}(t)$ with $SP \in \mathcal{P}_S$.

3 MAINTENANCE

To have a more realistic prognostics for system that made mission of several month as a ship. It is necessary to introduce the maintenance in our analysis. In this paragraph we defined the formalization of the maintenance action and maintenance plan applied to a complex system $\mathcal{S}$.

3.1 Maintenance action

As said resources are identified from the maintenance action. This means a maintenance action is performed to the resource level.

The goal of a maintenance action is to improve the resource health state thus to reduce its damage state. In general, a maintenance action a is defined by a function to evaluate the action performance m_a and by a belief rate η_a. $\mathcal{A}$ is the set of maintenance action.

$$\begin{cases} \mathcal{A} = \left\{a^i\right\},\ a^i = (m_{a^i}(\delta),\ \eta_{a^i}) \\ i \in \mathbb{N}^*,\ \eta_{a^i} \in [0, 1] \end{cases} \tag{15}$$

No duration is associated to a maintenance action because our objective is to model the maintenance plan that in order to provide be optimal anticipate the maintenance when resources are not working.

If $\delta \in [\delta^+, \delta^-]$ is the resource damage state before the maintenance action a^i, its damage δ' after the action will be $\delta' \in \left[\delta'^+,\ \delta'^-\right]$ defined by:

$$\begin{cases} \delta'^+ &= \min\left(\ (2 - \eta_{a^i})\, m_{a^i}(\delta^+),\ 1\ \right) \\ \delta'^- &= \max\left(\ \eta_{a^i}\ m_{a^i}(\delta^-),\ 0\ \right) \end{cases} \tag{16}$$

As example for m_{a^i} function we can cite a threshold or a gain function.

Maintenance plan

The maintenance plan for a mission represents the sequence of all timed maintenance actions on all resources. The maintenance plan $\mathcal{P}$ is thus defined by:

$$\begin{cases} \mathcal{P} = \left(\ (a_k,\ R_{a_k},\ t_k)_k\ \right) \\ k \in \mathbb{N}^*,\, a_k \in \mathcal{A},\, R_{a_k} \subseteq \mathcal{R} \end{cases} \tag{17}$$

where t_k is the action date and R_{a_k} is the sub-set of resources which the action is applied.

4 SYSTEMIC VIEW OF PROCESS

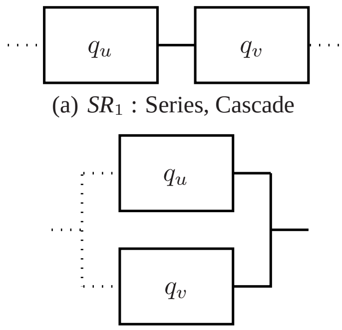
We defined the process in (2) as a decomposition tree of basic resources. But according to objectives, it can be interesting to have a damage feature of the complete system $\mathcal{S}$ or of one part i.e. sub-process *SP*. $\mathcal{P}_S = \left\{SP^j\right\}$ is the set of sub-process, a SP^j is characterized by the couple:

$$SP^j = (\ \mathcal{Q}_{SP^j}, B_{SP^j}\ ) \tag{18}$$

where $\mathcal{Q}_{SP^j}$ is the element set of the sub-process j and B_{SP^j} its structure.

$$\mathcal{Q}_{SP^j} = \{\ q_k\ \},\ k \in \mathbb{N}^{*+},\ q_k \in \mathcal{R} \cup \mathcal{P}_S \tag{19}$$

A sub-process is thus a node of the process tree $\mathcal{P}$. Resources and Sub-process have only one root sub-process.



(a) SR_1 : Series, Cascade

(b) SR_2 : Parallel, Bypass

Figure 4: Simple binaries structural relations

4.1 Structure and damage of sub-process

To estimate a metric of the sub-process damage from this elements i.e. $\mathcal{Q}_{SP^j}$, it is necessary to know how these elements are interconnected. We called a structural relation *SR* an interconnection model between elements. $\mathcal{B}$ denotes the set of *SR*:

$$\mathcal{B} = \{SR_k\} = \{\ (b_k,\ h_k)_k\ \},\ k \in \mathbb{N}^*,\ b_k \in \mathfrak{B} \tag{20}$$

where b_k is an n-ary relation to define SR_k between n elements and h_k is an n-ary function to estimate the damage metrics of SR_k. $\mathfrak{B}$ is the set of b_k. b_k and h_k are defined by applications:

$$\begin{array}{rcccc} b_k & : & (\mathcal{P}_S \cup \mathcal{R} \cup \mathfrak{B})^n & \longrightarrow & \mathfrak{B} \\ & & \{q_i\} & \longmapsto & b_k\left(\{q_i\}\right) \end{array} \tag{21}$$

$$\begin{array}{rcccc} h_k & : & [0,\ 1]^n & \longrightarrow & [0,\ 1] \\ & & \{\delta_{q_i}\} & \longmapsto & \delta_{SR_k} = h_k\left(\{\delta_{q_i}\}\right) \end{array} \tag{22}$$

The structure B_{SP^j} of SP_j is thus defined by a imbrication of the structural relations SR_k between elements of $\mathcal{Q}_{SP^j}$.

4.2 Structural relations definition example

Whether in electrical, mechanical or hydraulic when two components are connected, two possibilities are most often offered: a combination series (cascade) or a combination parallel (bypass).

In the proposed formalization, these two structure examples can be represented by two binaries relations respectively SR_1 and SR_2 for series and parallel. These relations are depicted on 4. The plain lines define the necessary connections to characterize the relation.

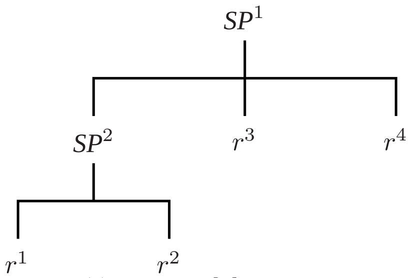
In term of availability, when the elements q_u et q_v are in series, if one of them are unavailable the function is not realized. Thus, the damage metric associated to the relation $b_1\left(q_u,\ q_v\right)$ is given by the more damaged elements:

$$h_1 = \max\left(\delta_{q_u},\ \delta_{q_v}\right) \tag{23}$$

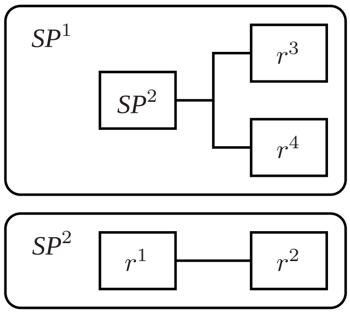
When q_u and q_v are in bypass, they form a redundant structure. So if one of them becomes unavailable the function is always realized. The damage metric associated to $b_2\left(q_u,\ q_v\right)$ is thus:

$$h_2 = \min\left(\delta_{q_u},\ \delta_{q_v}\right) \tag{24}$$

When structural relations are binaries, the SP^j structure can be represented as an abstract syntaxic tree where the node are the relations and the leaves are the element of Q_{SP^j}.



(a) Functional description



(b) Structural description

Figure 5: Process $\mathcal{P}$ example

Process example

Figure 5 shows an academic example of functional and structural decomposition of a simple system in two sub-process and four resources. In our prognostic formalism the process of this system is written by:

$$\begin{cases} \mathcal{R} = \{\ r^1,\ r^2,\ r^3,\ r^4\ \} \\ \mathcal{P}_S = \{\ SP^1,\ SP^2\ \} \\ \mathcal{B} = \{\ (b_1,\ h_1),\ (b_2,\ h_2)\ \} \\ SP^1 = (\ \mathcal{Q}_{SP^1}, B_{SP^1}\ ) \\ SP^2 = (\ \mathcal{Q}_{SP^2}, B_{SP^2}\ ) \\ \mathcal{P} = \langle\ SP^1, \mathcal{R}, \mathcal{P}_S, \mathcal{B}\ \rangle \end{cases} \tag{25}$$

with:

$$\begin{cases} \mathcal{Q}_{SP^1} = \{\ SP^2,\ r^3,\ r^4\ \} \\ B_{SP^1} = b_1(SP^2, b_2(r^3, r^4)) \\ \mathcal{Q}_{SP^2} = \{\ r^1,\ r^2\ \} \\ B_{SP^2} = b_1(r^1, r^2) \end{cases} \tag{26}$$

where h_1 and h_2 are the previously defined structural relations.

The *SR* damage metric allows to estimate the sub-process damage as:

$$\begin{cases} \delta_{SP^1} = \max(\delta_{SP^2}, \min(\delta_{r^3}, \delta_{r^4})) \\ \delta_{SP^2} = \max(\delta_{r^1}, \delta_{r^2}) \end{cases} \tag{27}$$

4.3 Sub-process damage estimation algorithm

The algorithm 1 gives the F_{SP} routine estimation. This algorithm is based on depth tree algorithm, the implementation of recurrent function allows to begin by estimate the damage trajectories of low level sub-process i.e. composed only by resources, and then to back by level to the function F_{SP_r} of the root sub-process. Trajectories are computed from B_{SP} where each n-ary relation b_k is replaced by its associated damage metric h_k.

Algorithm 1 Sub-process damage trajectories

Require: $\mathcal{P}$
Ensure: F_S for $S \in \mathcal{P}_S$
  $F_{SP_r}(t) \leftarrow \text{SPDAMAGE}\ (SP_r)$

  function SPDAMAGE (p)
    $Q = \mathcal{Q}_p \cap \mathcal{P}_S$
    if $Q \neq \emptyset$ **then**
      for all $q \in Q$ **do**
        $F_q(t) \leftarrow \text{SPDAMAGE}\ (q)$
      end for
    end if
    $\delta_p \leftarrow B_p$
    $\delta_p \leftarrow \text{REPLACE}\ (\delta_p, \mathcal{R} \cup \mathcal{P}_S, F_q(t))$
    $\delta_p \leftarrow \text{REPLACE}\ (\delta_p, \mathcal{B}, h_k)$
    $F_p(t) \leftarrow \text{EVALUATE}\ \delta_p$
    return $F_p(t)$
  end function

5 CONCLUSION

In this we presented the main lines of a novel generic framework for prognostics, some complementary informations can be found in cited publications. The framework is composed by a formalism to describe all kind of complex system and by a methodology to estimate damage trajectories over mission. According to objective and knowledge about the mission, this framework can be used to make a prognostics before or during the mission. But also after, if any parameters of the mission can be known a priori.

Yet most of the parameters that are need to build the prognostic model must be extracted from experts interview. Our future works are focused on use data-driven techniques such as machine learning to extract automatically the knowledge from an historical data set. These works requires, in a first time, to define what are the data that we need to have enough knowledge for a good prognostics.

REFERENCES

(Byington *et al.*, 2003) Carl S. Byington, Patrick W. Kalgren, Robert Johns, and Richard J. Beers. Prognosis enhancements to diagnostic system for improved condition based maintenance. In *IEEE Systems Readiness Technology Conference, AUTOTESTCON*, pages 320–329, California, USA, September 2003.

(Lebold and Thurston, 2001) Mitchell Lebold and Michael Thurston. Open standards for condition-based maintenance and prognostic systems. In *5th Annual Maintenance and Reliability Conference, MARCON*, Gatlinburg, USA, 2001.

(Peysson *et al.*, 2008a) Flavien Peysson, Mustapha Ouladsine, Rachid Outbib, and Jean-Baptiste Leger. Pronostic de l'état de dégradation d'un système complexe. In *Conférence Internationale Francophone d'Automatique*, Bucharest, Roumanie, September 2008.

(Peysson *et al.*, 2008b) Flavien Peysson, Mustapha Ouladsine, Rachid Outbib, Jean-Baptiste Leger, Olivier Myx, and Claude Allemand. Damage Trajectory Analysis based Prognostic. In *First IEEE International Conference on Prognostics and Health Management - PHM*, Denver, USA, October 2008.

(Peysson *et al.*, 2009a) Flavien Peysson, Mustapha Ouladsine, Rachid Outbib, and Jean-Baptiste Leger. Expert knowledge impact on damage trajectory analysis based prognostics. In IFAC, editor, *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS'09*, Barcelone, Espagne, jun 2009.

(Peysson *et al.*, 2009b) Flavien Peysson, Mustapha Ouladsine, Rachid Outbib, Jean-Baptiste Leger, Claude Allemand, and Olivier Myx. A Generic Prognostic Methodology using Damage Trajectory Models. *IEEE Transactions on Reliability*, 58(2):277–285, jun 2009.

(Racoceanu, 2006) Daniel Racoceanu. Mémoire d'habilitation à diriger des recherches, Contribution à la surveillance des Systèmes de Production en utilisant les Techniques de l'Intelligence Artificielle. Université de Franche-Comté de Besançon, France, January 2006.

(Vachtsevanos *et al.*, 2006) George .J. Vachtsevanos, Lewis Frank L., Michael Roemer, Andrew Hess, and Wu Biqing. *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. Hoboken, NJ: John Wiley & Sons, 2006.

Flavien Peysson received his master's degree in industrial engineering and system information science from the Aix-Marseille University (France) in 2005. He has been at the Aix-Marseille University where he is a Ph.D. Student in Computer Science and Mathematics. He worked on diagnostics, and his current works are on prognostics of complex systems.

Mustapha Ouladsine received his Ph.D. in 1993 in the estimation and identification of nonlinear systems from the Nancy University (France). In 2001, he joined the LSIS in Marseille (France). His research interests include estimation, identification, neural networks, control, diagnostics and prognostics and their applications in the vehicle, and aeronautic and naval domains. He has published more than 80 technical papers.

Rachid Outbib received his Ph.D. degree in applied mathematics in 1994, and his HDR in automatics control in 1998, from the University of Metz and Amiens (France). From 2003 to 2007, he was full professor at the University of Technology at Belfort (France). Since 2007, he has served as full professor at the University of Aix-Marseille. His main research interests concern non linear systems methods (control, diagnostics, and prognostics) with applications to fluid power systems, and vehicles.