Condition-based Monitoring and Prognosis in an Error-Bounded Framework

L. Travé-Massuyès[1,2] and **R. Pons**[1,2] and **P. Ribot**[1,3] and **Y. Pencole**[1,2] and **C. Jauberthie**[1,3]
[1]CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
[2] Univ de Toulouse, LAAS, F-31400 Toulouse, France
[3] Univ de Toulouse, Université Paul Sabatier, LAAS, F-31062 Toulouse, France
e-mail: louise@laas.fr, renaud.pons@gmail.com, pribot,ypencole,cjaubert@laas.fr

Abstract

Condition-based maintenance is recognized as a better health management strategy than regularly planned inspections as used nowadays by most companies. In practice, it is however difficult to implement because it means being able to predict the time to go before a failure occurs. This prediction relies on knowing the current health status of the system's components and on predicting how components age. This paper demonstrates the applicability of interval-based tools in integrated health management architectures, hence proposing an alternative to the standard statistical approach[1].
Keywords: Interval analysis, diagnosis, prognosis.

1 Introduction

Nowadays system's availability is a key ingredient of economical competitiveness. A typical example is the civil aircraft industry for which the unavailability of passenger carriers generate great costs and considerable economical losses. Technical inspections are generally planned on regular bases. If a component fails and needs to be replaced between two successive inspections, the plane is taken to a standstill and the company has to re-schedule the aircraft fleet, implying money loss during this unplanned immobilization. This is why condition-based maintenance is a preferable strategy that means predicting at inspection time the time to go before a new failure occurs. In this case the aircraft company can replace the part whose failure is estimated during the current inspection and then prevent an extra immobilization of the plane. This strategy not only saves a lot of money but also increases reliability and safety.

Integrated systems health management architectures performing condition-based monitoring naturally couple fault diagnosis and prognosis mechanisms [1; 2; 3; 4]. Diagnosis is used to assess the current state of the system and is used to initialize a prediction mechanism based on ageing models that aims to estimate the *remaining useful life* (RUL). In the prognosis process several sources of uncertainty can be identified, in particular the ageing models and the future stress conditions. These uncertainties are commonly taken into account through appropriate assumptions about noise and model error distributions, which are difficult to acquire. An alternative approach is to frame the problem in a set-membership framework and make use of recent advances in the field of interval analysis and interval constraint propagation.

This paper demonstrates the applicability of interval-based tools — briefly introduced in Section 2 — in integrated health management architectures, providing an interesting alternative to the standard statistical approach [5; 6]. It proposes a two stages *set-membership* (SM) *condition-based monitoring* method whose principle is presented in Section 3. The first stage is diagnosis that provides an estimation of the system's health status. It takes the form of SM parameter estimation using *Focused Recursive Partitioning* (FRP) and is the subject of Section 4. The second stage concerns prognosis in the form of the estimation of the remaining system's lifespan. It is based on the use of a *damaging table* and is detailed in Section 5. The case study of a shock absorber is used to illustrate the method and is presented in Section 6 before the concluding Section 7.

2 Interval analysis

Interval analysis was originally introduced to obtain guaranteed results from floating point algorithms [7] and it was then extended to validated numerics [8]. A *guaranteed result* first means that the solution set encloses the actual solution. It also means that the algorithm is able to conclude about the existence or not of a solution in limited time or number of iterations [9].

2.1 Interval

A real interval $x = [\underline{x}, \overline{x}]$ is a closed and connected subset of $\mathbb{R}$ where $\underline{x}$ and $\overline{x}$ represent the lower and upper bound of x, respectively. $\underline{x}$ and $\overline{x}$ are real numbers. The *width* of an interval x is defined by $w(x) = \overline{x} - \underline{x}$, and its *midpoint* by $mid(x) = (\overline{x} + \underline{x})/2$. If $w(x) = 0$, then x is degenerated and reduced to a real number. x is defined as positive (resp. negative), *i.e.* $x \geq 0$ (resp. $x \leq 0$), if $\underline{x} \geq 0$ (resp. $\overline{x} \leq 0$).

The set of all real intervals of $\mathbb{R}$ is denoted $\mathbb{IR}$. Two intervals x_1 and x_2 are equal if and only if $\underline{x}_1 = \underline{x}_2$ and $\overline{x}_1 = \overline{x}_2$. Real arithmetic operations have been extended to intervals [8]:

$$\circ \in \{+, -, *, /\},\ x_1 \circ x_2 = \{x \circ y \mid x \in x_1,\ y \in x_2\}.$$

An *interval vector* or *box* $[\mathbf{x}]$ is a vector with interval components. An *interval matrix* is a matrix with interval components. The set of $n-$dimensional real interval vectors is

[1]This work was supported by the CORALIE Project IA 2012–01–06 of the Council for Research in French Civil Aeronautics (CORAC), WP1 "Contrôle Santé", and by the French National Research Agency (ANR) in the framework of the project ANR-11-INSE-006 (MAGIC-SPS).

denoted by $\mathbb{IR}^n$ and the set of $n \times m$ real interval matrices is denoted by $\mathbb{IR}^{n \times m}$. The width $w(.)$ of an interval vector (or of an interval matrix) is the maximum of the widths of its interval components. The midpoint $mid(.)$ of an interval vector (resp. an interval matrix) is a vector (resp. a matrix) composed of the midpoints of its interval components.

Classical operations for interval vectors (resp. interval matrices) are direct extensions of the same operations for real vectors (resp. real matrices) [8].

2.2 Inclusion function

Given $\mathbf{x}$ a box of $\mathbb{IR}^n$ and a function f from $\mathbb{IR}^n$ to $\mathbb{IR}^m$, an *inclusion function* of f aims at getting a box containing the image of $\mathbf{x}$ by f. The *range* of f over $\mathbf{x}$ is given by:

$$f(\mathbf{x}) = \{f(\nu) \mid \nu \in \mathbf{x}\},$$

where ν is a real vector of the same dimension as $\mathbf{x}$. Then, the interval function $[f]$ from $\mathbb{IR}^n$ to $\mathbb{IR}^m$ is an *inclusion function* for f if:

$$\forall \mathbf{x} \in \mathbb{IR}^n,\ f(\mathbf{x}) \subset [f](\mathbf{x}).$$

An inclusion function of f can be obtained by replacing each occurrence of a real variable by its corresponding interval and by replacing each standard function by its interval evaluation. Such a function is called the *natural inclusion function*. A function f generally has several inclusion functions, which depend on the syntax of f.

2.3 Notations

Throughout the paper and unless explicitly mentioned, variables are assumed to take values in $\mathbb{IR}^d$, where d is the dimension of the variable. Exception is made for overlined and underlined variables that are assumed to take values in $\mathbb{R}^d$, where d is the dimension of the variable. Bold symbols are used to denote multi-dimensional variables (vector or matrices).

3 Principle of the Set-Membership Health Management Method

3.1 Method Architecture

The architecture of the preventive maintenance method is shown in Fig. 1. The method relies on two modules:

- A *diagnosis* module that uses the system measured inputs and outputs to compute an estimation of the system's health status; this is performed by estimating the value of the system parameter vector $\boldsymbol{\theta}$ by the means of a *behavioral Model* Σ of the system.
- A *prognosis* module that predicts the parameter evolution over time by using a *Damaging Model* Δ and computes the *Remaining Useful Life* or RUL of the underlying subsystems.

The global model representing the progressive evolution of the system over time, *e.g.* the *Ageing Model*, is obtained by putting together the *behavioral model* Σ and the *damaging model* Δ. The behavioral model takes the form of a state space model, *i.e.* a state equation modeling the dynamics of the system state vector, and an observation equation that links the state variables to the observed variables:

$$\Sigma\ :\begin{cases}\dot{\mathbf{x}}(t) = f(t,\ \mathbf{x}(t),\ \boldsymbol{\theta}(t),\ \mathbf{u}(t)) \\ \mathbf{y}(t) = h(t,\ \mathbf{x}(t),\ \boldsymbol{\theta}(t),\ \mathbf{u}(t))\end{cases} \quad (1)$$

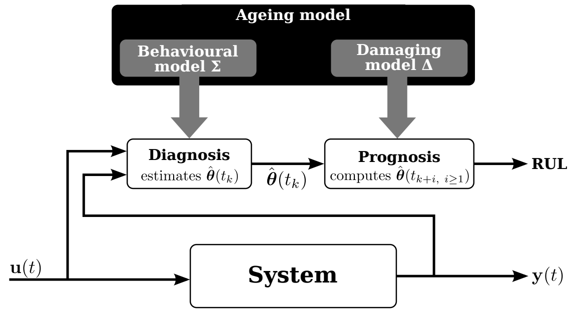


Figure 1 – Health management architecture.

where
$\mathbf{x}(t)$ is the state vector of the system of dimension n_x,
$\mathbf{u}(t)$ is the input vector of dimension n_u,
$\mathbf{y}(t)$ is the output vector of dimension n_y,
$\boldsymbol{\theta}(t)$ is the parameters vector of dimension n_θ.

Σ represents the system's nominal behavior as it is supposed to act when its parameters have not yet suffered any ageing.

The damaging model Δ represents the dynamics of the behavioral model parameters. It is described by the dynamic state equation (2) where the equation states are the system parameters. The equation models how the parameters evolve over time because of the wearing, leakage, etc.:

$$\Delta\ :\dot{\boldsymbol{\theta}}(t) = g(t, \boldsymbol{\theta}(t), \mathbf{w}, \mathbf{x}(t)) \quad (2)$$

where $\mathbf{w}$ is a wearing parameter vector of dimension n_θ.

3.2 Unit Cycles

Predicting the evolution of the system's behavior requires to know *a priori* how the system will be solicited either by the control system or by external causes (*e.g.* environmental conditions, temperature, humidity, etc.) This knowledge is generally difficult to obtain. In our approach, we make assumptions about the future solicitations of the system by determining the most usual way the system is intended to be used and we define the notion of *unit cycles*. A unit cycle $\mathscr{C}$ is defined as a solicitation that repeats in time and that leads to a behavioral sequence that is known to impact system's ageing.

For example, in the case of a pneumatic valve from the Space Shuttle cryogenic refueling system, [2] defines a unit cycle as the opening of the valve, the filling of the tank and the valve closing when the tank is full. In the case of an aircraft, an unit cycle may be chosen to be a flight: it starts with the plane take-off, a cruising stage and landing.

One may simultaneously use unit cycles at different time scales, depending on the dynamics of the system and its subsystems. As an example, one may define a "global" unit cycle for a bus as being the journey from the starting station to the terminus, and another unit cycle for the subsystem "bus doors" as being the opening and the closing of the doors at each station.

4 SM Diagnosis

Diagnosis is achieved through SM parameter estimation. This problem assumes that measured outputs $\mathbf{y}_m(t_i)$ generated by the real system on a time horizon $t_i = t_0, \ldots, t_H$ of length $H \times \delta$, where δ is the sampling period, are corrupted by bounded-error terms that may originate from the system parameters varying within specified bounds, bounded noise, or sensor precision. The $\mathbf{y}_m(t_i)$'s are hence interval vectors of $\mathbb{IR}^{n_y}$. The SM parameter estimation problem for the system Σ is formulated as finding the set $\mathbf{\Theta} \subseteq \mathbb{R}^{n_\theta}$ of real parameter vectors such that the arising outputs $\mathbf{y}(t_i, \boldsymbol{\theta}) \in \mathbb{R}^{n_y}$ hit all the output data sets, *i.e.*:

$$\boldsymbol{\theta} \in \mathbf{\Theta} \Leftrightarrow \mathbf{y}(t_i, \boldsymbol{\theta}) \in \mathbf{y}_m(t_i), \forall t_i \in \{t_0, \ldots, t_H\}.$$

$\mathbf{\Theta}$ is called the *feasible parameter set* (FPS). SM parameter estimation problems are generally solved with a branch-and-bound algorithm like SIVIA [10] that enumerates candidate box solutions thanks to a rooted tree and assumes the full parameter space as the root. At every node, the set of $\hat{\mathbf{y}}(t_i), t_i = t_0, \ldots, t_H$, arising from the considered box parameter vector $[\boldsymbol{\theta}]^*$, *i.e.* solution of Σ for any real $\theta \in [\boldsymbol{\theta}]^*$, is checked for consistency against the measurements and labelled feasible, unfeasible or undetermined. Unfeasible candidates are rejected while undetermined candidates are split and checked in turn until the set precision of the candidate solutions is below a given threshold ε provided by the user.

Such algorithms return an overestimation of the FPS given by the convex union of the candidates that have been labelled feasible and undetermined [11]. Interestingly, the convex union may consist of one set or more, which means that the systems does not need to be identifiable in the classical sense [12].

When considering a SIVIA-based algorithm for dynamical systems like Σ, a critical step is the determination of the inclusion function for the state vector $\hat{\mathbf{x}}(t_i)$ at instants $t_i = t_0, \ldots, t_H$, arising from a given candidate parameter vector $[\boldsymbol{\theta}]^*$, from which the $[\hat{\mathbf{y}}(t_i)], t_i = t_0, \ldots, t_H$ can be computed using the observation equation of Σ. This step relies on set-membership integration for which we have chosen the interval Taylor series integration scheme implemented in the VNODE-LP solver [13]. Although quite well optimized [14], it is well-known that this method is computationally stable only for $[\boldsymbol{\theta}]^*$ of very small size. SIVIA-like parameter estimation algorithms are hence particularly inefficient as they enumerate candidate parameter subspaces starting with the full parameter space. This is why we propose the FRP schema presented in the next section.

4.1 Principle of FRP-based SM Parameter Estimation

The principle of the FRP method is based on partitioning the parameter search space $\mathcal{S}(\boldsymbol{\theta})$. Each part of the partition represents a candidate parameter vector $[\boldsymbol{\theta}]_j$ for which SM integration of the state equation of Σ provides a conservative numerical enclosure $\hat{\mathbf{x}}(t_i)_j, t_i = t_0, \ldots, t_H$. The output vector can now be estimated as:

$$\hat{\mathbf{y}}(t_i)_j = h\left(t, \hat{\mathbf{x}}_j, [\boldsymbol{\theta}]_j, \mathbf{u}_m\right), \forall t_i \in \{t_0, \ldots, t_H\} \quad (3)$$

We then keep track of the parameters vectors for which the $\hat{\mathbf{y}}(t_i)_j$'s are consistent with the measurements, for all $t_i = t_0, \ldots, t_H$, *i.e.* the unfeasible ones are discarded. Computing the convex hull then provides us with a minimal and maximal value for the admissible parameter vectors.

The consistency test is defined as testing the intersection of the estimated output vector with the measurements:

$$\text{If } \exists\, t_i \in \{t_0, \ldots, t_H\} \text{ s.t. } \hat{\mathbf{y}}(t_i)_j \cap \mathbf{y}_m(t_i) = \emptyset, \quad (4)$$

then there is no consistency between the estimation and the measured input $\mathbf{u}_m(t)$ and output $\mathbf{y}_m(t)$ with the tested parameter vector $[\boldsymbol{\theta}]_j$. The parameters box $[\boldsymbol{\theta}]_j$ is unfeasible and hence rejected.

$$\text{If } \hat{\mathbf{y}}(t_i)_j \subseteq \mathbf{y}_m(t_i),\ \forall t_i \in \{t_0, \ldots, t_H\}, \quad (5)$$

then $[\boldsymbol{\theta}]_j$ is a parameter vector for which the estimation is consistent with the measurements. The box is added to the list of the solution parameter boxes:

$$\mathbb{P} = \mathbb{P} \cup [\boldsymbol{\theta}]_j. \quad (6)$$

If none of the two previous conditions is true, *i.e.*:

$$[\hat{\mathbf{y}}(t_i)]_j \cap [\mathbf{y}_m(t_i)] \neq \emptyset,\ \forall t_i \in \{t_0, \ldots, t_H\}, \quad (7)$$

it means that the parameter box $[\boldsymbol{\theta}]_j$ is undetermined and that it *partially* contains solutions. The box is also added to $\mathbb{P}$. Two different labels allow us to keep track of the boxes that are feasible or undetermined. The convex union of these boxes provides the estimation $\hat{\boldsymbol{\theta}}$ that encloses the feasible parameter set $\mathbf{\Theta}$, *i.e.* $\hat{\boldsymbol{\theta}} \supseteq \mathbf{\Theta}$.

The quality of the enclosure depends on the size of the boxes of the partition, in other words on the partition precision. A way to improve the enclosure is to proceed with a partition of the obtained solution $\hat{\boldsymbol{\theta}}$ and run another round of consistency tests over the new boxes, and so on recursively. The process of iterating the partition ends when the gain in precision is low with respect to the SM integration and consistency tests computational cost. The method is detailed for a one dimension parameter vector in the next section.

The *estimation precision* $\boldsymbol{\omega}(P)$ obtained for a given partition P can be evaluated by the following percentage:

$$\boldsymbol{\omega}(P) = \left|\text{mid}(\hat{\boldsymbol{\theta}})\right| ./ \left(\left|\text{mid}(\hat{\boldsymbol{\theta}})\right| + w(\hat{\boldsymbol{\theta}})/2\right) \quad (8)$$

where ./ denotes the division of two vectors term by term.

Given two partitions P_i and P_j, one can evaluate the precision gain as:

$$\boldsymbol{G}(P_j/P_i) = \boldsymbol{\omega}(P_j)./\boldsymbol{\omega}(P_i). \quad (9)$$

4.2 Parameter search space

The domain value of the parameter vector $\boldsymbol{\theta}$ is given by $\Omega(\boldsymbol{\theta}) = [.\text{inf}(\boldsymbol{\theta}_{bol}, \boldsymbol{\theta}_{eol}), .\text{sup}(\boldsymbol{\theta}_{bol}, \boldsymbol{\theta}_{eol})]$, where .inf and .sup denote the operators inf and sup applied term by term and $\boldsymbol{\theta}_{bol}$ and $\boldsymbol{\theta}_{eol}$ are the real vectors whose components are given by $\theta_{k,bol}$ and $\theta_{k,eol}$ for each parameter $\theta_k, k = 1, \ldots, n_\theta$:

- $\theta_{k,bol}$ (*bol*: "beginning-of-life") is the factory setting defined by the design specification,
- $\theta_{k,eol}$ (*eol*: "end-of-life") is the maximal/minimal admissible value, *i.e.* the value above/below which the component is considered to have failed and the function is no longer guaranteed.

During the system's life, the impact of ageing results in the parameter vector value evolving in $\Omega(\boldsymbol{\theta})$. Depending on the impact of ageing, its value may decrease or increase with time:

$$\boldsymbol{\theta}(t_i) = \boldsymbol{\alpha}\boldsymbol{\theta}(t_j), t_i \geq t_j \quad (10)$$

where $\boldsymbol{\alpha}$ is an n_θ dimensional real vector whose components α_k are greater or lower than 1 depending on the impact of ageing on the change direction of the parameter.

We assume a health management strategy, which means that diagnosis (and prognosis) is performed according to a given inspection planning at some chronologically ordered times of the system's life $T_0, \dots, T_F$. The parameter vector search space depends on the time and is hence denoted by $\mathcal{S}(\boldsymbol{\theta})_{T_i} = [\underline{\mathcal{S}}(\boldsymbol{\theta})_{T_i}, \overline{\mathcal{S}}(\boldsymbol{\theta})_{T_i}]$, where $T_i \in \{T_0, \dots, T_F\}$. Initially, for the first inspection time T_0, we set $\mathcal{S}(\boldsymbol{\theta})_{T_0} = \Omega(\boldsymbol{\theta})$. Diagnosis then returns the estimated parameter value $\hat{\boldsymbol{\theta}}(T_0)$. For the next inspection time, $\mathcal{S}(\boldsymbol{\theta})$ is updated by taking the parameter value estimation into account as follows:

- if $\alpha_k > 1$, $\theta_{k,bol}$ is replaced by $\inf(\underline{\hat{\boldsymbol{\theta}}_k}(T_0), \overline{\hat{\boldsymbol{\theta}}_k}(T_0))$,
- if $\alpha_k < 1$, $\theta_{k,bol}$ is replaced by $\sup(\underline{\hat{\boldsymbol{\theta}}_k}(T_0), \overline{\hat{\boldsymbol{\theta}}_k}(T_0))$,

and one of the bounds of the components of $\mathcal{S}(\boldsymbol{\theta})_{T_1}$ remains equal to $\theta_{k,eol}$.

In the general case, when considering the inspection time T_i, $\mathcal{S}(\boldsymbol{\theta})_{T_i}$ is hence obtained with $\hat{\boldsymbol{\theta}}(T_{i-1})$ as follows:

- if $\alpha_k > 1$, $\inf(\underline{\hat{\boldsymbol{\theta}}_k}(T_{i-1}), \overline{\hat{\boldsymbol{\theta}}_k}(T_{i-1}))$ is replaced by $\inf(\underline{\hat{\boldsymbol{\theta}}}(T_i), \overline{\hat{\boldsymbol{\theta}}}(T_i))$,
- if $\alpha_k < 1$, $\sup(\underline{\hat{\boldsymbol{\theta}}_k}(T_{i-1}), \overline{\hat{\boldsymbol{\theta}}_k}(T_{i-1}))$ is replaced by $\sup(\underline{\hat{\boldsymbol{\theta}}_k}(T_i), \overline{\hat{\boldsymbol{\theta}}_k}(T_i))$.

4.3 FRP Parameter Estimation for a Single Parameter

In this section, we consider one single parameter θ whose evolution is monotonically increasing. As an example, let's state that θ is a bearing friction coefficient that grows with the bearing wearing and the clogging of the environment. In the general case, this kind of knowledge must be brought by an expert of the system and/or the manufacturer.

Let us consider the first inspection time and the initial search space $\mathcal{S}(\theta)_{T_0}$ given by the domain value of the parameter $\Omega(\theta) = [\theta_{\text{bol}},\ \theta_{\text{eol}}]$. The search space is partitioned into boxes, in our case intervals (cf. Fig. 2).

The dynamic equation of Σ is integrated on the time window $t_i = t_0, \dots, t_H$, where $t_H = T_0$, as many times as the number of intervals in the partition P_1. The number of intervals is defined by the partition factor $\epsilon(P_1)$, which equals $1/15$ in our example (cf. Fig. 2). We start with $[\theta]_1 = [\theta_{\text{bol}}, \theta_{\text{bol}} + pw]$, where $pw = \epsilon(P_1)w(\mathcal{S}(\theta)_{T_0})$ is the width of the partition intervals, then proceed with the subsequent intervals $[\theta]_j$. For each interval, we get an estimation of the state vector at times $t_i = t_0, \dots, t_H$, denoted as $\hat{\mathbf{x}}(t_0 \dots t_N)_j$, and obtain $\hat{\mathbf{y}}(t_0 \dots t_N)_j$ thanks to the observation equation (1). This latter is tested for consistency against the measurements $\mathbf{y}_m(t_0 \dots t_N)$.

Depending on the output of the tests (4), (5), and (7), the parameter interval $[\boldsymbol{\theta}]$ is rejected or added to the solution as feasible or undetermined (red-colored, green-colored, and yellow-colored parts, respectively, in Fig. 2).

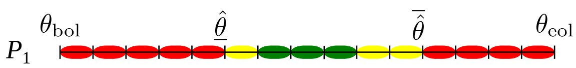


Figure 2 – Partition P_1 and test results for this partition.

The convex union of feasible and undetermined intervals provides a *guaranteed estimation* $\hat{\theta} = \left[\underline{\hat{\theta}},\ \overline{\hat{\theta}}\right]$ of the admissible values for θ. We iterate the process by creating a new partition P_2 of $\left[\underline{\hat{\theta}},\ \overline{\hat{\theta}}\right]$ with a precision $\epsilon(P_2) = 1/10$ (cf. Fig. 3).

We proceed as above for each interval of P_2 in order to refine the bounds of $\left[\underline{\hat{\theta}},\ \overline{\hat{\theta}}\right]$ and find a more precise enclosure of feasible parameter solution (see Fig. 3).

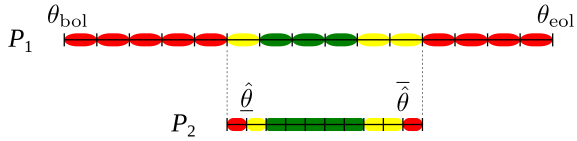


Figure 3 – Partition P_2 and test results for this partition.

We iterate the process until the precision gain $\boldsymbol{G}(P_{i+1}/P_i)$ is greater then a given threshold, as it is shown in Fig. 4.

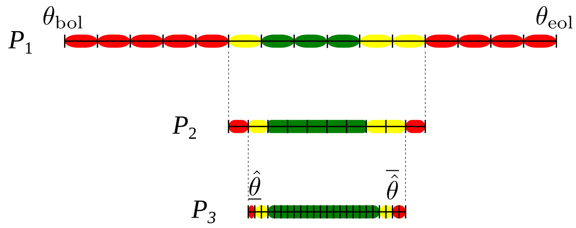


Figure 4 – Test results for partition P_3.

Remarks

The method can be easily generalized to a system whose parameter vector has dimension $n_\theta > 1$. The computing cost is proportional to the number of boxes that are tested, *i.e.* $\sum_{i=1}^{n_P} 1/\epsilon(P_i)$, where n_P is the number of partitions.

Let's notice that the partition may be non-regular. For example, for a slowly ageing parameter, one may choose small boxes for the values of θ that are close to θ_{bol} and larger ones for the values close to θ_{eol}. The result is guaranteed even if the partition has not been properly chosen or if the parameter has evolved in a non expected way, although the computation cost may be higher.

The convex union provides a poor result if the set of admissible values is made of several mutually disjoint connected sets, as shown in Fig. 5. The algorithm may test some boxes that have already been rejected by the tests of the previous partition. This drawback could be addressed by defining the solution as a list of boxes whose labels (unfeasible, feasible, or undetermined) are inherited by the next partition boxes.

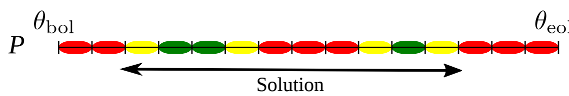


Figure 5 – The returned solution is the convex hull of mutually disjoint connected intervals.

5 SM prognosis

The prognosis phase consists is calculating the number of cycles remaining before anomaly, which is also called the *Remaining Useful Life* or RUL. To optimally adapt this calculation to the system's life requires the knowledge of the health status of the system at the current time, which was the topic of Section 4.

5.1 Component degradation

The global model ($\Sigma + \Delta$) assumes that the parameters of the behavior model Σ given by (1) evolve in time, and that their evolution is represented by the degradation model Δ given by the dynamic equation (2) that is recalled below:

$$\Delta \,:\, \dot{\boldsymbol{\theta}}(t) = g(t, \boldsymbol{\theta}(t), \mathbf{w}, \mathbf{x}(t)).$$

Δ provides the dynamics of the parameter vector as a function of the state of the system $\mathbf{x}(t)$ and of a degradation parameter vector $\mathbf{w}$ that allows one to tune the degradation for each of the considered parameters.

The global model ($\Sigma + \Delta$), in the form of a dynamic model with varying parameters, cannot be directly integrated by VNODE-LP. An original method, coupling the two models Σ and Δ iteratively is proposed in the following. The method is illustrated by Fig. 6 and used to determine the degradation suffered by each parameter during one unit cycle as defined in Section 3.2.

Let us denote $\mathbf{u}_{\mathscr{C}}(t)$, $t \in [\tau, \tau + d_{\mathscr{C}}]$, the system input stress during one unit cycle $\mathscr{C}$. As shown in Fig. 6, the following steps are iteratively executed, every iteration corresponding to a computation step given by the sampling period δ:

1. The normal behavior model Σ is used first with input $\mathbf{u}(t) = \mathbf{u}_{\mathscr{C}}(\tau)$ to compute the state $\mathbf{x}(\tau)$ and the output $\mathbf{y}(\tau)$;
2. The parameters are updated with the degradation model Δ using the value of the state determined previously, *i.e.* $\boldsymbol{\theta}(\tau)$ is computed;
3. The parameters of the behavior model Σ are updated with $\boldsymbol{\theta}(\tau)$;
4. The next stress input value $\mathbf{u}_{\mathscr{C}}(\tau + \delta)$ is considered, and so on until the end of the cycle, *i.e.* until the last value of the cycle $\mathbf{u}_{\mathscr{C}}(\tau + d_{\mathscr{C}})$ is reached.

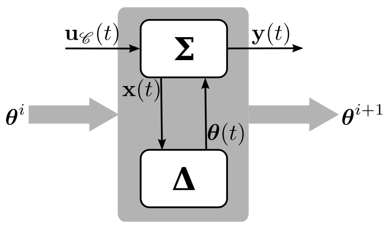


Figure 6 – Computation of the degradation parameters during one unit cycle.

The above algorithm defines the function:

$$\mathcal{D} : \mathbb{IR}^{n_\theta} \to \mathbb{IR}^{n_\theta} \tag{11}$$

where n_θ is the number of parameters of the system. Let's assume the cycle i, then $\mathcal{D}$ maps $\boldsymbol{\theta}^i$ into $\mathcal{D}(\boldsymbol{\theta}^i) = \boldsymbol{\theta}^{i+1}$, which is the value of $\boldsymbol{\theta}$ after one unit cycle.

$\mathcal{D}$ is nonlinear. Thus the value of the parameter vector after one cycle $\boldsymbol{\theta}^{i+1}$ depends on the initial value $\boldsymbol{\theta}^i$. Indeed, we know that a system generally degrades in a nonlinear fashion. We must hence compute $\boldsymbol{\theta}^{i+1}$ for all possible values of the parameter vector $\boldsymbol{\theta}^i$.

For this purpose, the domain value $\Omega(\theta_k)$ of each parameter θ_k is partitioned into N_k intervals. N_k is chosen sufficiently large to reduce non conservatism of the interval function $\mathcal{D}$. The domain value of the parameter vector $\boldsymbol{\theta}$ is hence partitioned into $N_\Pi = \Pi_{k=1}^{n_\theta} N_k$ possible boxes that must be fed as input to $\mathcal{D}$. Let us for instance consider a two parameters vector and its beginning-of-life and end-of-life values as follows:

$$\boldsymbol{\theta} = \begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}, \boldsymbol{\theta}_{bol} = \begin{bmatrix}1\\ 1\end{bmatrix}, \boldsymbol{\theta}_{eol} = \begin{bmatrix}4\\ 9\end{bmatrix} \text{ and } N = \begin{bmatrix}3\\ 2\end{bmatrix}, \tag{12}$$

then, if we select the partition landmarks as $\{5\}$ for θ_1 and $\{2, 3\}$ for θ_2 [2], $\mathcal{D}$ must be run for the following 6 box values:

$$[\boldsymbol{\theta}]_1 = \begin{bmatrix}[1,2]\\ [1,5]\end{bmatrix}, [\boldsymbol{\theta}]_2 = \begin{bmatrix}[2,3]\\ [1,5]\end{bmatrix}, [\boldsymbol{\theta}]_3 = \begin{bmatrix}[3,4]\\ [1,5]\end{bmatrix},$$
$$[\boldsymbol{\theta}]_4 = \begin{bmatrix}[1,2]\\ [5,9]\end{bmatrix}, [\boldsymbol{\theta}]_5 = \begin{bmatrix}[2,3]\\ [5,9]\end{bmatrix}, [\boldsymbol{\theta}]_6 = \begin{bmatrix}[3,4]\\ [5,9]\end{bmatrix}. \tag{13}$$

For each of these box values taken as input for cycle i, *i.e.* $\boldsymbol{\theta}^i = [\boldsymbol{\theta}]_l$, $l = 1, \ldots, 6$, $\mathcal{D}$ returns the (box) value $\boldsymbol{\theta}^{i+1}$ after one unit cycle. This computation is then projected on each dimension to obtain a set of n_θ tables, $\mathcal{D}_{\theta_k}$, $k = 1, \ldots, n_\theta$, that provide the degradation of each individual parameter θ_k after one unit cycle.

5.2 RUL determination

The RUL, understood as a RUL for the whole system, can now be determined by computing the number of cycles that are necessary for the parameters to reach the threshold defining the end-of-life (cf. Fig. 7).

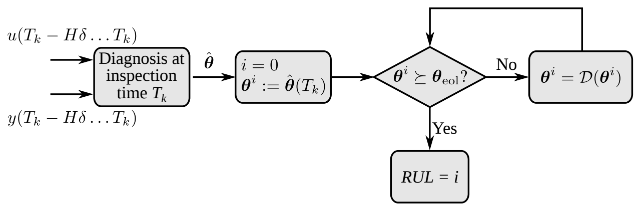


Figure 7 – RUL computation

For the cycle $i = 0$, $\boldsymbol{\theta}^0$ is initialized with $\hat{\boldsymbol{\theta}}$, which is the result of the parameter estimation computed by the diagnosis engine. $\hat{\boldsymbol{\theta}}$ is given as input to $\mathcal{D}$, which returns $\mathcal{D}(\hat{\boldsymbol{\theta}}) = \boldsymbol{\theta}^1$. i is incremented by 1 and $\boldsymbol{\theta}^1$ is given as input to $\mathcal{D}$ and so on until the set-membership test $\boldsymbol{\theta}^i \succeq \boldsymbol{\theta}_{eol}$ is achieved, which provides the stopping condition. This test may take several forms as explained in Section 5.3. If the test is true, then the index i is the number of cycles required to reach the degradation threshold, so RUL = i.

For a given cycle i, the box value $\boldsymbol{\theta}^i$ that must be given as input to $\mathcal{D}$ is not necessarily among the values $[\boldsymbol{\theta}]_l$, $l = 1, \ldots, N_\Pi$, of the partition. We propose to compute $\boldsymbol{\theta}^{i+1}$ by assuming that the mapping between $\boldsymbol{\theta}^i$ and $\boldsymbol{\theta}^{i+1}$ is linear in every domain l of the partition. Considering $\mathbf{p} \in \mathbb{R}^{\mathbf{n}_\theta}$, $\mathcal{D}(\mathbf{p})$ is approximated as follows:

$$\forall \boldsymbol{\theta} \in [\boldsymbol{\theta}]_1, \mathcal{D}(\boldsymbol{\theta}) \approx \mathbf{a} \odot \boldsymbol{\theta} + \mathbf{b},\ \text{l=1}, \ldots, \mathbf{N_\Pi} \tag{14}$$

where a= $w(\mathcal{D}([\boldsymbol{\theta}]_l))./w([\boldsymbol{\theta}]_l)$, b= $\underline{\mathcal{D}([\boldsymbol{\theta}]_l)} - a\underline{[\boldsymbol{\theta}]_l}$, and $\odot$ is the product of two vectors term by term.

Equation (14) is applied to $\overline{\boldsymbol{\theta}}^i$ and $\underline{\boldsymbol{\theta}}^i$ to obtain an approximation of $\mathcal{D}(\boldsymbol{\theta}^i)$.

[2] Notice that the intervals issued from the partitioning are not required to be of equal length.

5.3 Set-membership test for the RUL

The set-membership test implemented with the order relation $\preceq$ may take several forms. For instance, if the test $\boldsymbol{\theta}^i \succeq \boldsymbol{\theta}_{eol}$ is interpreted as:

$$\exists k \in \{1, \ldots, n_\theta\} \mid$$
$$\overline{\theta}_k^i \geq \theta_{k,eol} \text{ if } \alpha_k > 1 \text{ or } \underline{\theta}_k^i \leq \theta_{k,eol} \text{ if } \alpha_k < 1, \quad (15)$$

then it means that the bound of the interval value of at least one parameter θ_k is above or below its end-of-life threshold value $\theta_{k,eol}$. The RUL is then qualified as the "worst case RUL", which means that the RUL indicates the earliest cycle at which the system may fail.

One can also test whether the value higher bound of one of the parameters is higher than its end-of-life threshold, that is to say:

$$\exists k \in \{1, \ldots, n_\theta\} \mid$$
$$\underline{\theta}_k^i \geq \theta_{k,eol} \text{ if } \alpha_k > 1 \text{ or } \overline{\theta}_k^i \leq \theta_{k,eol} \text{ if } \alpha_k < 1. \quad (16)$$

The RUL then represents the cycle at which it is certain that the system will fail.

It is obviously possible to combine these different tests applied to the different individual parameters depending on their criticality.

6 Case study

6.1 Presentation

The case study is a shock absorber that consists of a moving mass connected to a fixed point via a spring and a damper as illustrated by Fig. 8. The movement of the mass takes place in the horizontal plane in order to eliminate the forces due to gravity. Aerodynamic friction forces are neglected.

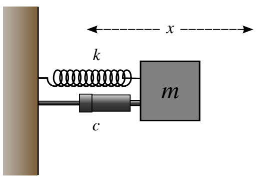


Figure 8 – Spring and damper system

The Newton's second law is written as:

$$m\vec{a} = \Sigma\vec{F} = \vec{F}_r + \vec{F}_c + \vec{u} \quad (17)$$

where m is the mass, $\vec{a}$ is the acceleration, $\vec{F}_k$ is the spring biasing force, $\vec{F}_c$ is the friction force exerted by the damper and $\vec{u}$ is the force applied on the mass. Expressing the forces and the acceleration as a function of the position of the mass $x(t)$, we get:

$$\ddot{x}(t) + \frac{c}{m}\dot{x}(t) + \frac{k}{m}x(t) = u(t) \quad (18)$$

where k is the spring stiffness constant (N/m), m is the mobile mass (kg), and c is the damping coefficient (Ns/m). (18) is a second order ODE. Let us rewrite

$$\frac{c}{m} = 2\zeta\boldsymbol{\omega}_0 \text{ and } \frac{k}{m} = \boldsymbol{\omega}_0^2$$

and we get

$$\boldsymbol{\omega}_0 = \sqrt{\frac{k}{m}} \text{ and } \zeta = \frac{c}{2\sqrt{km}}.$$

The impulse response of such system depends on the value of ζ:

- if $\zeta = 0$, then the answer is a sinusoid;
- if $0 < \zeta < 1$, then the answer is a damped sinusoid;
- if $\zeta \geq 1$, then the answer is a decreasing exponential.

The state model is given by the equation:

$$\begin{cases} \dot{X}(t) = \begin{bmatrix} 0 & 1 \\ \frac{-k}{m} & \frac{-c}{m} \end{bmatrix} X(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} U(t) \\ Y(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} X(t) \end{cases} \quad (19)$$

with $X(t) = [x(t),\ \dot{x}(t)]^T$, and the transfer function is:

$$\frac{X(p)}{U(p)} = \frac{1}{p^2 + \frac{c}{m}p + \frac{k}{m}}. \quad (20)$$

An example of bounded error step response obtained with VNODE-LP with a sampling parameter $\delta = 0.1$ s, $c = 1$, $m = 2$ and $k = [3,9\,;4,1]$ is shown in Fig. 9a. There, $\zeta \simeq 0.177$ and the step response is a damped sinusoid. Because k is assumed to have an uncertain value bounded by an interval, the outputs are in the form of envelops.

6.2 Unit cycle

In the case study, a unit cycle is defined by the application of a power unit for a determined time. The force is applied at time t_0+5s, where t_0 is the cycle starting time. The force lasts 20s and cancels at t_0+ 25s as shown by the red curve of Fig. 9b. The cycle ends at $t_0 + 50$s.

Fig. 9b presents the system's response for a spring constant $k = [3.9,\ 4.1]$ N/m, a mass $m = 2$ kg, a damping coefficient $c = 10$ Ns/m, and initial speed and position equal to zero. The response is a decreasing exponential.

6.3 Degradation model

The degradation model chosen is the ageing of the damper cylinder. It is represented by a reduction of the damping coefficient proportional to the velocity of the mass [15]:

$$\dot{c} = \beta\dot{x},\ \beta < 0. \quad (21)$$

The more the spring is used, the weaker it becomes, characterized by the change in the damping coefficient.

6.4 Diagnosis

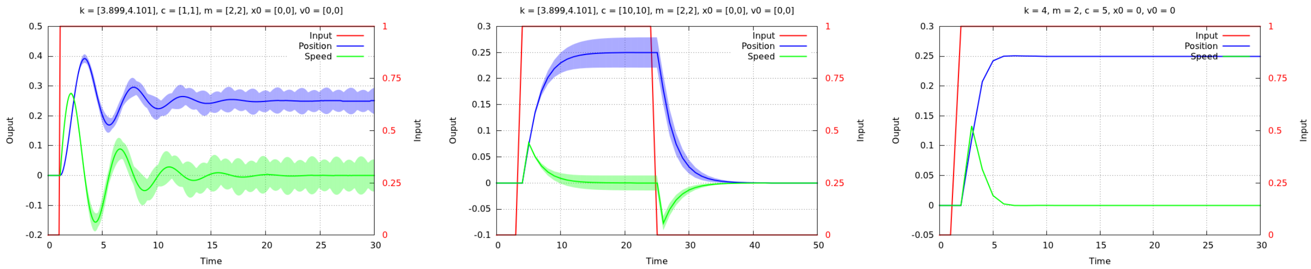
The FRP parameter estimation method presented in Section 4 has been used with the measures shown in Fig. 9c. These measures were obtained for

$$\boldsymbol{\theta} = \begin{bmatrix} c \\ k \\ m \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \\ 2 \end{bmatrix} \quad (22)$$

The goal is to estimate the damping coefficient c and the stiffness constant k. The search space is defined by the interval $[4\ 9]$ for c and $[3.5,\ 9]$ for k. The value of m is assumed to be known $m = 2$. Using the notation introduced above, we have:

$$\boldsymbol{\theta}_{bol} = \begin{bmatrix} 4 \\ 3.5 \\ 2 \end{bmatrix}, \boldsymbol{\theta}_{eol} = \begin{bmatrix} 9 \\ 9 \\ 2 \end{bmatrix} \quad (23)$$

The partition P_1 is achieved with a precision $\epsilon(P_1) = 1/10$ for the two parameters to be estimated c and k. Fig. 10 presents two examples of prediction results with two parameter boxes of P_1: $[\boldsymbol{\theta}]_i = \big[[4,1,4,2],\ [4,7,4,8],\ 2\big]^T$ on the



(a) Step response for $\zeta = 0.177$. (b) Unit cycle for the case study. (c) Measured input and output.

Figure 9 – Cases study simulation and data plots.

left and $[\boldsymbol{\theta}]_j = \left[[5,\ 5, 1],\ [4,\ 4, 1],\ 2\right]^T$ on the right. On the left figure, one can see that there is no intersection between the estimate and the measurement for the position, hence the box used for the simulation is rejected. On the right, there is an intersection between the measurement and the estimation for all time points, but the estimate is not included in the measure envelop, hence the parameter box is considered undetermined.

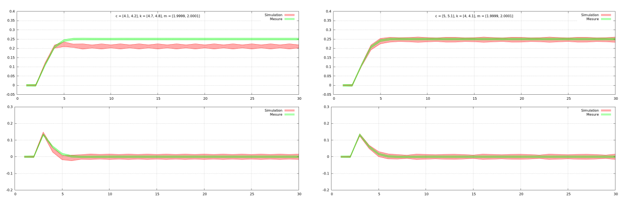


Figure 10 – Estimation results with a rejected parameter box (left) and an indetermined box (right)

The results for partition P_1 are presented in Fig. 11a and we obtain a first estimation for $\boldsymbol{\theta}$:

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} [4, 1,\ 5.8] \\ [3, 7,\ 4, 2] \\ 2 \end{bmatrix}.$$

The estimation precision for partition P_1 is given by:

$$\boldsymbol{\omega}(P_1) = \left|\text{mid}(\hat{\boldsymbol{\theta}})\right| ./ (\left|\text{mid}(\hat{\boldsymbol{\theta}})\right| + w(\hat{\boldsymbol{\theta}})/2) = \begin{bmatrix} 0.85 \\ 0.94 \\ 1 \end{bmatrix} \quad (24)$$

The first estimation for $\boldsymbol{\theta}$ is used as the search space for partition P_2, whose precision is increased by a factor of 10, *i.e.* $\epsilon(P_2) = 0, 1$. The obtained estimation results are shown in Fig. 11b.

The estimation is refined as:

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} [4, 51,\ 5, 57] \\ [3, 85,\ 4, 14] \\ 2 \end{bmatrix}.$$

The precision is now $\boldsymbol{\omega}(P_2) = [0.9,\ 0.96,\ 1]^T$, and the precision gain is $\boldsymbol{G}(P_2/P_1) = [0.056,\ 0.025, 0]^T$. The values for the gain indicate that partitioning a third time might be quite inefficient. To confirm this fact, let us perform a third partition P_3, whose precision is increased by a factor of 5, *i.e.* $\epsilon = 0.02$ (cf. Fig. 11c). The new estimation for $\boldsymbol{\theta}$ is $\hat{\boldsymbol{\theta}} = \left[[4.548,\ 5.526],\ [3.872,\ 4.132], 2\right]^T$, and the precision gain is $G(P_3/P_2) = [0.0073,\ 0.0036,\ 0]^T$. As expected, the gain is quite negligible with respect to the computation time increase.

6.5 RUL computation

In this section we apply the set-membership method described in Section 5.2 to compute the RUL for the damping coefficient c.

The damper is assumed to fail when $c \leq c_{eol} = 2$. The degradation model (21) with $\beta = -0, 1^3$ allows us to determine the degradation table $\mathcal{D}_c$ for the parameter c for a unit cycle:

$\mathcal{D}_c =$ (25)

c^i	$\mathcal{D}(c^i) = c^{i+1}$	c^i	$\mathcal{D}(c^i) = c^{i+1}$
[9, 10]	[8.917, 9.977]	[4, 5]	[3.874, 4.977]
[8, 9]	[7.911, 8.978]	[3, 4]	[2.863, 3.973]
[7, 8]	[6.898, 7.979]	[2, 3]	[1.721, 2.97]
[6, 7]	[5.814, 6.982]	[1, 2]	[0, 1.98]
[5, 6]	[4.859, 5.979]	[0, 1]	[0, 0.9755]

After proceeding to the linear interpolation given by (14), the graphical representation of c^{i+1} as a function of c^i is given by Fig. 12.

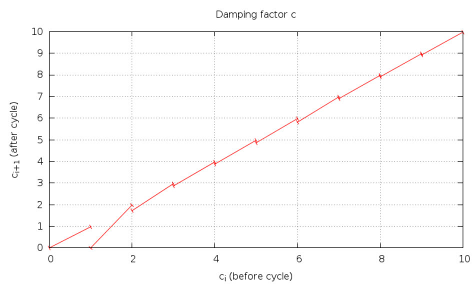


Figure 12 – Approximated degradation of the damping coefficient c

The number of elements of the partition has been chosen relatively small to better illustrate the method. In a real situation, this number should be high in order to obtain less conservative predictions.

The value of c has been previously estimated and is

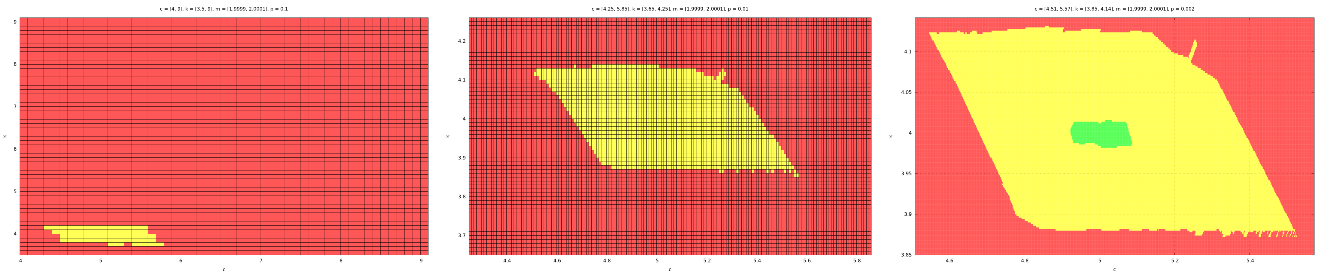
$$\hat{c} = [4.548,\ 5.526].$$

The graph of Fig. 12 allows us to approximate the predicted value after one unit cycle:

$$\mathcal{D}(\hat{c}) = c^1 = [4.4787,\ 5.4481].$$

The next iteration of the algorithm allows us to compute c^2, etc. After 30 iterations, we obtain $c^{30} = [1.7665,\ 3.4235]$.

[3] The coefficient β has been chosen arbitrarily to illustrate the approach; it does not represent the real ageing of a damper.



(a) Partition P_1 (b) Partition P_2 (c) Partition P_3

Figure 11 – Partitions and estimation results (red, yellow and green boxes are resp. rejected, undetermined, accepted parameters values).

Since $\underline{c}^{30} < c_{eol}$, we get $\underline{RUL} = 30$ cycles. After the 44*th* iteration, we get $c^{44} = [0.037591,\ 1.985928]$. We then have $\overline{c}^{44} < c_{eol}$ and hence $\overline{RUL} = 44$ cycles. The RUL of the damper is hence given by:

$$RUL = [30,\ 44] \text{ cycles.}$$

7 Conclusion

This paper addresses the condition-based monitoring and prognostic problems with a new focus that trades the traditional statistical approach by an error-bounded approach. It proposes a two stages method whose principle is to first determine the health status of the system and then use this result to compute the RUL of the system. This study uses advanced interval analysis tools to obtain *guaranteed* results in the form of interval bounds for the RUL.

The results for the case study demonstrate the feasibility of the approach. The next step is to adapt the FRP-based SM parameter estimation algorithm in order to output a list of boxes instead of a single box given by the convex hull of the boxes. The convex hull is indeed a very conservative approximation when the solution set is not convex.

The second stream of work is to consider contextual conditions and their associated uncertainties. Environmental conditions, like weather, different usage, etc. may indeed significantly affect the stress input and prognostics results.

References

[1] Indranil Roychoudhury and Matthew Daigle. An integrated model-based diagnostic and prognostic framework. In *Proceedings of the 22nd International Workshop on Principle of Diagnosis (DX'11). Murnau, Germany*, 2011.

[2] Matthew J Daigle and Kai Goebel. A model-based prognostics approach applied to pneumatic valves. *International Journal of Prognostics and Health Management*, 2:84, 2011.

[3] J. Luo, K. R Pattipati, L. Qiao, and S. Chigusa. Model-based prognostic techniques applied to a suspension system. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 38(5):1156–1168, 2008.

[4] Q. Gaudel, E. Chanthery, and P. Ribot. Hybrid particle petri nets for systems health monitoring under uncertainty. *International Journal of Prognostics and Health Management*, 6(022), 2015.

[5] D. Gucik-Derigny, R. Outbib, and M. Ouladsine. Estimation of damage behaviour for model-based prognostic. In *Fault Detection, Supervision and Safety of Technical Processes*, pages 1444–1449, 2009.

[6] X. Guan, Y. Liu, R. Jha, A. Saxena, J. Celaya, and K. Geobel. Comparison of two probabilistic fatigue damage assessment approaches using prognostic performance metrics. *International Journal of Prognostics and Health Management*, 1(005), 2011.

[7] R.E. Moore. *Automatic error analysis in digital computation*. Technical report LMSD-48421, Lockheed Missiles and Space Co, Palo Alto, CA, 1959.

[8] R.E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, 1966.

[9] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with examples in parameter and state estimation, Robust control and robotics*. Springer, Londres, 2001.

[10] L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29:1053–1064, 1993.

[11] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with examples in parameter and state estimation, Robust control and robotics*. Springer, Londres, 2001.

[12] C. Jauberthie, N. Verdière, and L. Travé-Massuyès. Fault detection and identification relying on set-membership identifiability. *Annual Reviews in Control*, 37:129–136, 2013.

[13] N. Nedialkov. VNODE-LP, a validated solver for initial value problems for ordinary differential equations.

[14] R. J. Lohner. Enclosing the solutions of ordinary initial and boundary value problems. In E. W. Kaucher, U. W. Kulisch, and C. Ullrich, editors, *Computer Arithmetic: Scientific Computation and Programming Languages*, pages 255–286. Wiley-Teubner, Stuttgart, 1987.

[15] Ian M Hutchings. *Tribology: friction and wear of engineering materials*. Butterworth-Heinemann Ltd, 1992.