Particle filter-based prognostics: review, discussion and perspectives

Marine Jouin*, Rafael Gouriveau, Daniel Hissel, Marie-Cécile Péra, Noureddine Zerhouni

FEMTO-ST Institute, UMR CNRS 6174 - UBFC / UFC / ENSMM / UTBM,
FC-LAB Research, FR CNRS 3539,
24 rue Alain Savary, 25000 Besançon, France
firstname.lastname@femto-st.fr

Abstract

Particle filters are of great concern in a large variety of engineering fields such as robotics, statistics or automatics. Recently, it has developed among Prognostics and Health Management (PHM) applications for diagnostics and prognostics. According to some authors, it has ever become a state-of-the-art technique for prognostics. Nowadays, around 50 papers dealing with prognostics based on particle filters can be found in the literature. However, no comprehensive review has been proposed on the subject until now. This papers aims at analyzing the way particle filters are used in that context. The development of the tool in the prognostics' field is discussed before entering the details of its practical use and implementation. Current issues are identified, analyzed and some solutions or work trails are proposed. All this aims at highlighting future perspectives as well as helping new users to start with particle filters in the goal of prognostics.

Keywords: Prognostics, Particle filters, Implementation, Issues, Challenges

1. Introduction

Prognostics and Health Management (PHM) is an enabling discipline that aims at utilizing real monitoring data to facilitate relevant indicators and trends that depict the health of a system. Seven modules ranging from data acquisition to decision making are combined to help preserving the integrity of a system [31]. A key activity in PHM is prognostics. Indeed, it enables predicting the remaining useful life (RUL) of the system and helps anticipating and avoiding failure. A great variety of techniques is available to perform prognostics [70] depending on the knowledge and data available.

Among these techniques, particle filters are more and more employed. It has been developing this last decade in the prognostics' field even becoming considered as a state of the art technique. However, no comprehensive review is available to discuss the issues coming with this tool, or to compare the different existing points of view on the subject. It can be interesting to notice, and also to show, that most of works dealing with particle filters in the prognostics field are just using existing approaches and applying it to prognostics. Most of them lack of real adaptation to prognostics requirements. Moreover, with no synthesis of the existing literature, it is quite difficult to start with particle filters to perform prognostics with only basic notions.

To provide answers to these comments, or at least to start addressing them, the main contributions of this paper are the following: (1) the background and a short review about particle filters in a general context, (2) an analysis of particle filters in prognostics applications addressing all the issues from the filter selection to the uncertainty management, and (3) the highlighting of remaining issues and challenges as well as the proposal of solutions or work trails. For this purpose, this paper is organized in two main parts.

In a first part, the theory of particle filter and its basic functioning are presented. A short state of the art is also drawn to show the existing types of implementation and some of the general challenges. Then, in the second part, the use of particle filter in the prognostics' field is studied. A comprehensive analysis of the different techniques available for each step of the implementation is proposed. To do so, first the perception of the tool in the PHM community, the requirements for prognostics as well as its advantages and drawbacks are discussed. Then, the model adaptation needed to have the state and measurement models are discussed and the different types of filters most commonly encountered for prognostics are discussed. Section II. 5 deals with the implementation of particle filters while Section II. 6 discuss how to use it to perform prognostics. Finally, the existing metrics to evaluate the results based on particle filters are summarized before discussing how to deal with uncertainty with such a tool.

*Corresponding author, Tel.: +33 (0)3 81 40 29 04, Fax +33 (0)3 81 40 28 09

Part I
Particle filters - Theory and generalities

Particle filers are used in many fields: robotics, statistics, automatics, etc. and more recently in diagnostics and prognostics. However, the bases always remain the same.

1. Nonlinear Bayesian tracking

1.1. Problem statement

A Bayesian tracking problem is defined by two elements [3]:

1. a state vector that contains all the relevant information required to describe the system under investigation;
2. a measurement vector representing the noisy observations that are related to the state vector. It is generally of dimension equal or lower than the state vector.

The signal is modeled as Markovian, nonlinear, nonstationary and may be non-Gaussian [19]. We remind that a Markov process is a stochastic process with the Markov property, i.e. the conditional probability distribution of future states only depends from the current state and not the past states. Nonlinear refers both to the classical definition of a nonlinear system, i.e. a system for which the output is not directly proportional to the input, and its mathematical representation, i.e. a nonlinear state equation.

The knowledge of this information is translated into at least two models in order to analyze and make inference about the dynamic system [3, 19]:

1. *the state model* (or system model): describes the evolution of the state with time $\{x_t, t \in \mathbb{N}\}$, $x_t \in \mathrm{X}$ is modeled as a Markov process of initial distribution $p(x_0)$ and a transition equation $p(x_t|x_{t-1})$. Note that this state can be unobserved (hidden states).
2. *the measurement model* (or observation model): relates the noisy measurements to the state. The observations are written $\{y_t, t \in \mathbb{N}^*\}$, $y_t \in \mathrm{Y}$ are assumed to be conditionally independent given the process $\{x_t, t \in \mathbb{N}\}$ and of marginal distribution $p(y_t|x_t)$.

According to [3], an important assumption is that these models are available in a probabilistic form. However, to be used in a filtering framework, the models are more commonly found in the following forms [38]:

$$x_t = f(x_{t-1}, u_t, \omega_t) \leftrightarrow p(x_t|x_{t-1}) \tag{1}$$

$$y_t = h(x_t, v_t) \leftrightarrow p(y_t|x_t) \tag{2}$$

where u_t is the command input of the system and ω_t and v_t are white noises, non-necessarily Gaussian. Some examples of non-Gaussian white noises can be found in [24].

The probabilistic state-space formulation and the updating of information based on new measurements are ideal for the Bayesian approaches [3].

1.2. Bayesian approach

The Bayesian approach consists in constructing the posterior probability density function (pdf) of the state based on all available information, such as the knowledge of the system or sets of measurements. In principle, an optimal (with respect to any criterion) estimate of the state may be obtained [3].

The state is estimated recursively via a filtering approach. According to [3], it means that received data can be processed sequentially rather than as a batch so that it is not necessary to store the complete data set nor to reprocess existing data if a new measurement becomes available. In practice, we will see later in the prognostics part that storing the data can be necessary to make improved filtering.

Filtering uses 2 stages:

1. *prediction*: it uses the state model to estimate the current state;
2. *update*: it uses the latest measurement to modify the prediction pdf with the Bayes' rule.

For the record, the Bayes' rule can be formulated as follows for a continuous variable [65]:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\int p(y|x')p(x')dx'} \tag{3}$$

with the requirement that $p(y) > 0$.

In the context of filtering, x is a quantity that we would like to infer from y. The probability $p(x)$ is referred to as *prior probability distribution*, and y is called the data or the observations. The distribution $p(x)$ summarizes the knowledge regarding the variable x prior to incorporating y. $p(x|y)$ is called the *posterior probability distribution* [65].

Based on that, the Bayesian tracking problem consists in recursively calculate some degree of belief in the state x_k at time k, given the data $y_{1:k}$ up to time k, and so perform the construction of the pdf $p(x_k|y_{1:k})$. The initial pdf $p(x_0|y_0) \equiv p(x_0)$ of the state vector is assumed available. The pdf $p(x_k|y_{1:k})$ is obtained recursively from the 2 stages previously mentioned: prediction and update

1. prediction: use of the system model to obtain the prior pdf of the state at time k via Chapman-Kolmogorov equation:

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{k-1})dx_{k-1} \tag{4}$$

2. update: at time k, a measurement y_k becomes available and this may be used to update the prior via Bayes' rule:

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} \tag{5}$$

where the normalizing constant

$$p(y_k|y_{1:k-1}) = \int p(y_k|x_k)p(x_k|y_{1:k-1})dx_k \qquad (6)$$

depends on the likelihood function $p(y_k|x_k)$ defined by the measurement model (details about the likelihood's definition are given further).

This forms the optimal Bayesian solution. However, it is only a conceptual solution: in general it cannot be determined analytically. For that purpose, a whole family of filtering tools exists ranging from the Kalman filter (KF) and its variations (extended KF or unscented KF), histograms and particle filters. The choice between these filters depends on the dynamics of the system and the shape of the noise distributions. A non-exhaustive classification that helps guiding this choice is proposed in [59] and is reproduced in Figure 1.
The focus is now on particle filters.

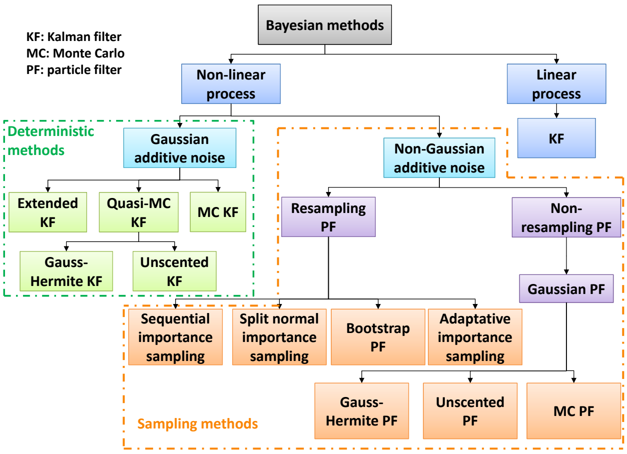


Figure 1: A proposal of Bayesian methods' classification from [59]

2. Particle filtering

Working with particle filters has a lot of advantages but also some drawbacks. A list can be found in [34], however it is not detailed here as we will come back to that subject in Part II.

2.1. Principle and hypotheses

As for perfect Monte Carlo (MC) sampling, particle filtering assumes that we are able to simulate N independent and identically distributed random samples that are called particles according to $p(x_{0:t}|y_{1:t})$ [19]. Based on the MC simulation principle, the particle filter's approximation represents a continuous distribution by discrete random measures composed of particles x_t^i which are possible values of the unknown state x_t at time t.
[38] defines the objective of a particle filter as a sequential estimation of the distribution of the state, including:

- the filtering distribution $p(x_t|y_{1:t})$ (the posterior probability distribution);
- the prediction distribution $p(x_t|y_{1:t-1})$;
- the smoothing distribution $p(x_t|y_{1:T})$ $t < T$;

However, according to the application, not all these distributions are used and it is more relevant for future discussions to focus on:

- the prior probability distribution $p(x_t|x_{t-1})$;
- the posterior probability distribution $p(x_t|y_{1:t})$;
- and the importance density (or proposal distribution) $q(x_t|x_{0:t-1}, y_{1:t})$;

The success of the particle filter algorithm depends on the validity of the following assumption [68]:

- Monte Carlo assumption: the Dirac point mass approximation provides an adequate representation of the posterior distribution;
- Importance sampling assumption: it is possible to obtain samples from the posterior by sampling from a suitable proposal distribution and applying importance sampling corrections

These assumptions become clearer with some explanations of the process.

2.2. A large family of filters

2.2.1. Sequential Importance Sampling: the classic form

Sequential Importance Sampling (SIS) is a MC method that forms the basis for most sequential filters. It is known under different names: bootstrap filtering, condensation algorithm, particle filtering, interacting particle approximation, survival of the fittest [3].
The main idea is to use a set of random samples with associated weights to represent the required posterior density function. The state estimates are then computed based on these samples and weights. Consequently, the discretization of the state space is given by a discrete weighted approximation:

$$p(x_{0:k}|y_{1:k}) \approx \sum_{i=1}^{N} \omega_k^i \delta(x_{0:k} - x_{0:k}^i) \qquad (7)$$

with the weights normalized such that $\sum \omega_k^i = 1$. This is the MC assumption.
The weights are chosen according to the principle of importance sampling [3]. We suppose that $p(x)$ which is proportional to $\pi(x)$ is a probability density from which it is difficult to draw samples but for which $\pi(x)$ can be evaluated. Moreover, let $x^i \approx q(x), i = 1...N$ be samples that

are easily generated from a proposal $q(.)$ called the importance density. We have the weighted approximation density:

$$p(x) \approx \sum_{i=1}^{N} \omega^i \delta(x - x^i) \quad (8)$$

where $w^i \alpha \frac{\pi(x^i)}{q(x^i)}$.

At each iteration, one could have samples constituting an approximation to $p(x_{0:k-1}|y_{1:k-1})$ and want to approximate $p(x_{0:k}|y_{1:k})$ with a new set of samples. From the relation

$$q(x_{0:k}|y_{1:k}) = q(x_k|x_{0:k-1}, y_{1:k})q(x_{0:k-1}|y_{1:k-1}) \quad (9)$$

one can derive the weight update equation

$$w_k^i \alpha w_{k-1}^i . \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, y_k)} \quad (10)$$

and the posterior filtered density:

$$p(x_k|y_{1:k}) \approx \sum_{i=1}^{Ns} \omega_k^i \delta(x_k - x_k^i) \quad (11)$$

Also according to [3], there are three main issues when using the SIS:

1. *a degeneracy problem*: after a few iterations all but one particle have a negligible weight.
   A measure of the degeneracy of the algorithm is the effective sample size (ESS), written N_{eff},

   $$N_{eff} = \frac{N}{1 + var(\omega_k^{*i})} \quad (12)$$

   where ω_k^{*i} is the "true" weight. It cannot be evaluated exactly so it is approximate as:

   $$\hat{N_{eff}} = \frac{1}{\sum(\omega_k^i)^2} \quad (13)$$

   Two possibilities are well-known to reduce this effect: a good choice of importance density and the use of resampling. This leads to points 2 and 3.
2. *the choice of a good importance density*: a method implies choosing the importance density $q(x_k|x_{k-1}^i, y_k)$ to minimize $var(\omega_k^{*i})$ which is equivalent to maximize N_{eff}. However, it has two major drawbacks as it requires the ability to sample from $p(x_k|x_{k-1}^i, y_k)$ and also to evaluate an integral over the new state. There exist two cases in which the use of the optimal importance density is possible:
   - x_k is a member of a finite set;
   - or with a class of models for which $p(x_k|x_{k-1}^i, y_k)$ is Gaussian (dynamics nonlinear with linear measurements).
3. *resampling*: the principle is to eliminate particles that have small weights and to focus on particles with larger weights. This step involves generating a new set of particles $\{x_k^{i*}\}_{i=1}^{N}$ by resampling N times from $p(x_k|y_{1:k})$. The result is an independent identically distributed (iid) sample from the discrete density and the weights are reset to 1/N.
   It introduces practical problems, the most important one is the loss of diversity: particles that have high weights are selected many times eliminating entire parts of the state space.

The SIS algorithm is the basis of all particle filter algorithms. A great number of derivative filters were proposed all along the last fifteen years however they all follow a classical scheme.

2.2.2. General particle filter scheme

Based on the SIS, all the particle filters follow the same scheme:

1. *Initialization*: N particles are created based on the initial state of the system. According to [65], if one knows the values of x_0, the initial distribution should be initialized with a distribution that centers all probability mass on the correct value of x_0 and assigns 0 probability anywhere else. If we are entirely ignorant of x_0, a uniform distribution over the domain of x_0 can be used (or a related distribution from the Dirichlet family).
2. *Prediction*: the prior probability distribution at time t $p(x_t|x_{t-1})$ is evaluated thanks to the state model, the evaluated state x_{t-1} at time $t-1$ and the inputs of the system u_t.
3. *Update*: A new measurement is available, the weights are calculated thanks to equation (10). Particles propagation together with weight computation is called importance sampling [38].
4. *Resampling*: the particles with low weights are eliminated and the other ones duplicated. Resampling is equivalent to modifying the random measure by improving the exploration of the state space at $t + 1$ [38].

Update and resampling are the most delicate tasks of the procedure.

First, based on equation (10), weight computation requires the knowledge of

- the likelihood distribution $p(y_k|x_k^i)$ that measure the matching between each particle and the latest measurement which expression comes from the measurement model;
- the prior distribution $p(x_k|x_{k-1})$;
- and the importance density $q(x_k|x_{k-1}, y_k)$

The choice of the importance density is far from being a trivial task as the form of $q(.)$ is almost always unknown. There exist different approaches for the density extraction [65]: Gaussian approximation, k-means clustering, histogram, density tree, kernel density estimation.

It is also possible to look for the proposal distribution $q(x_t|x_{0:t-1}, y_{1:t}) = p(x_t|x_{0:t-1}, y_{1:t})$ that minimizes the variance of the importance weights conditional on $x_{0:t-1}$ and $y_{1:t}$ as stated earlier. However, the most popular choice is to use $q(x_t|x_{0:t-1}, y_{1:t}) = p(x_t|x_{t-1})$ [68]. The crucial point is that if one fails to use the latest available information, only a few particles will have significant importance weights when their likelihood is evaluated.

Then the second hot point is resampling. To avoid degeneracy two kind of solutions exists: the brute force strategy consisting in increasing the number of particles and a more refined strategy that consists in implementing evolved resampling strategies. According to [37], to avoid sample impoverishment, resampling should be executed only at deterministic steps, for instance only when the variance of the non-normalized weights is superior to a threshold which consists in the use of the effective sample size (ESS) previously mentioned. This allows controlling the frequency of resampling. Indeed, resampling too often increases the risk of losing diversity. But when resampling too infrequently, many samples might be wasted in low probability regions. Resampling is adopted to force particles to areas of high likelihood from low likelihood areas. There exist a lot of resampling strategies. [37] define them as “blind” when the moving of the particle is done without a specific direction, or “sighted” meaning that new observations are used to direct the moving operation.

2.2.3. Evolved particle filters

A great diversity of particle filters has appeared. Their differences rely mainly on the choice of the importance sampling density and/or changes in the resampling procedure. Not all of them have particular names so when reviewing them, to have a complete overview of the literature, both the different types of filters and the different types of resampling should be studied. Extensive reviews on the subjects are already existing [13, 38], explaining how the procedures work. So here only the names of the filters/resampling are presented. For instance in [38], the authors propose a classification of resampling schemes distinguishing the sequential and parallel implementations as well as four criteria: the selected distribution, the resampling of all the particles in same way or not, the grouping of particles and the use of the latest information available or not.

The different types of filter and resampling are summarized in Tables 1 and 2. It is important to mention that the modified particle filters used in the context of prognostics are not integrated to these tables. A focus on these ones will be proposed in the next part.

As it can be seen, choosing a particle filter can be a hard with such diversity. There are still some challenges linked to that tool mainly regarding resampling [38]: (1) simplifying the resampling algorithms, (2) developing better schemes with a goal of improving performance (speed, convergence, etc.), (3) parallelization or (4) real-time implementation.

Table 1: Types of filters

Type of filter	References
Sequential Importance Sampling (SIS)	[3, 13, 19, 68]
Sampling Importance Resampling filter (SIR)	[3, 13, 34]
Auxiliary Sampling Importance Resampling filter (ASIR)	[3, 34]
Auxiliary particle filter	[13, 68]
Regularized particle filter	[3, 34]
MCMC PF	[13, 52]
Unscented particle filter	[26, 68]
Artificial fish swarm based particle filter	[36]
SMC PHD filter	[40]
Saturated particle filter	[61]
Mixed Kalman particle filter	[13, 30]
Mixture Particle Filters	[13]
Risk sensitive particle filter	[64]
Self-adaptive particle filter	[60]
Direct sampling particle filter	[67]
Rao-Blackwellized filter	[13]
Rejection Particle Filter	[13]
Other Monte Carlo Filters	[13]

These challenges might also be encountered when dealing with prognostics based on particle filters. Now the purpose of what follows is to conduct a survey on how the particle filter is used in prognostics? What are the main improvements already made on that tool in the context of prognostics? And what future works should tend toward on that subject?

Part II
Particle filters in prognostics

1. Related works

Particle filters are more and more used to perform prognostics. However, no review has been made regarding the available literature. For that purpose, 46 published works are considered and analyzed, see Table 3. The idea is to try to tackle the numerous practical issues of this tool.

To follow the process from the selecting the particle filter as a tool to include in a prognostics' framework, to the implementation and the results interpretation, the rest of the paper is organized as depicted in Figure 2. Also note

Table 2: Types of resampling strategies

Multinomial resampling	[38, 68]
Residual sampling	[38, 68]
Minimum variance sampling	[68]
Smoothing Metropolis-Hastings	[68]
Annealed importance sampling	[42]
Bayesian importance sampling	[42]
Adaptive importance sampling	[42]
Numerically accelerated sampling	[42]
Non-parametric importance sampling	[42]
Stratified sampling	[38]
Systematic sampling	[38]
Classification	[38]
Dynamic threshold	[38]
Fixed threshold	[38]
Modified resampling	[38]
Variable size resampling	[38]
Roughening	[38]
KLD-resampling	[22, 41]
IBIS	[21]
Optimal nudging	[43]

that the main conclusions of each section are summarized in the tables of Section 8.

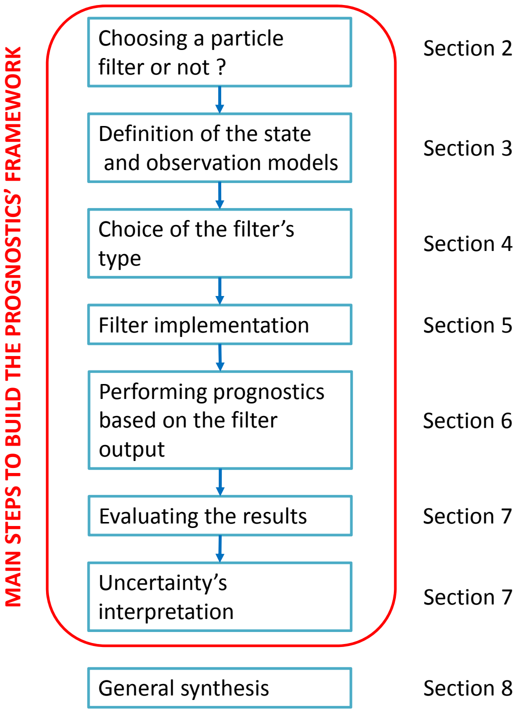

MAIN STEPS TO BUILD THE PROGNOSTICS' FRAMEWORK
Choosing a particle filter or not ?
Section 2
Definition of the state and observation models
Section 3
Choice of the filter's type
Section 4
Filter implementation
Section 5
Performing prognostics based on the filter output
Section 6
Evaluating the results
Section 7
Uncertainty's interpretation
Section 7
General synthesis
Section 8

Figure 2: Synopsis of the discussions

Table 3: Prognostics works based on particle filters considered for the review

Reference	Application case
[4, 50, 51, 62, 73, 75]	crack growth
[1, 2, 14, 16, 18, 23, 25, 27, 28, 37, 44, 45, 46, 49, 53, 54, 55, 56, 57, 63, 66, 71]	Li-ion batteries
[32, 33, 35]	PEM fuel cells
[5, 6]	turbine blade (creep growth)
[12]	water tank system
[15]	pneumatic valve
[72]	tool wear
[10]	landing gear retraction
[9]	semi-conductor manufacturing
[29]	simulated data
[7]	aircraft actuator system
[8]	wind turbine
[20]	LED light sources
[11]	carrier plate
[47]	vibration feature
[69]	jet engine
[74]	bearing

2. Why using particle filter for prognostics?

2.1. Perception by the PHM community

The particle filter is considered by many as the state-of-the-art tool for model-based or hybrid prognostics. Indeed, it allows using a Bayesian formulation of a problem. Both physical models and data can be incorporated to the framework and the formulation thanks to a state vector enables linking the state of the system to numerous inputs. The classification of the approach into model-based or hybrid prognostics is still debated. Indeed, a model is needed to perform state of health estimation but data are also continuously used to update the parameters of the model and the state estimation.
A lot of arguments are used to justify the use of a particle filtering-based prognostics as it will be shown in section 2.4. However, before listing them, it is necessary to think a little further about what is expected from such a tool in the prognostics field.

2.2. Requirements for prognostics

On the one hand, we have the particle filter which primary objective is to estimate the state of the system based on a state model and measurements. If some unknown parameters remain in the model, this objective can be extended to combined parameter and state estimations. On the other hand, there is prognostics which basically uses a set of available measurements to learn the state or the behavior of the system (i.e. the learning) and then predicts its future state-of-health (SoH) and remaining useful life (RUL) (i.e. the prediction). The learning can be model-based, purely data-driven or a combination of both. Based

on these simple remarks, what can be expected from particle filters in prognostics?
According to [28], the requirements for prognostics based on particle filters are quite common:

1. a behavioral model that includes the degradation;
2. degradation measurements;
3. a measurement equation;
4. a failure threshold.

Among these four components, the first three are classical of the Bayesian framework used with particle filters. In [49], it is reminded that these three components together should also satisfy the following statement: a proper representation for the probability of rare events and highly non-monotonic phenomena is required. Events non-related to the standard state evolution should also be considered, among them we can propose:

- changes in the mission profiles;
- maintenance interventions;
- uncontrolled rejuvenation phenomena;
- unpredictable / sudden failures;

Globally, this all enters in two categories when designing the prognostics framework:

1. the modeling of the system;
2. the distributions' management by the filter, i.e. filter design.

However, these requirements are not really prognostics-specific and could be encountered in any applications of SoH estimation of an industrial system based on particle filters.
To perform prognostics whatever the method employed, it is expected to predict the SoH and RUL without new measurements. This is clearly out of the competence range of a particle filter which cannot work without measurement. This leads to an obvious conclusion but never explicitly mentioned in the literature:

Conclusion 1 A particle filter is not used for prognostics but for state estimation.

As stated in [75], it is necessary to devise procedures with capabilities of projecting at future times the current particle population, in absence of new observations, adjusting the weights if necessary. Consequently, a correct implementation of a particle filtering-based prognostics should follow the scheme depicted in Figure 3. Special procedures for prognostics based on the particle filter's output are still very scarce and will be discussed in Section 5.
If we go further, why preferring a particle filter instead of another method for SoH estimation? Before discussing the advantages and drawbacks of the methods, we must try to position this tool with respect to other method for model-based applications.

2.3. Positioning of particle filters

Using particle filters without a deep prior reasoning can lead to bad performance. Indeed, in [39] the authors show that using a filter does not necessary lead to good results if the context is not appropriate.
The first step of the reasoning is to decompose the different stages involved in model-based prognostics, Figure 4. There are mainly three stages:

1. model identification;
2. state reconstruction;
3. prognostics.

Some methods are proposed for each stage. During model identification, unknown parameters of the model are estimated based on the data available. One method should be selected according to the hypotheses linked to the application such as parameter evolution with time, nonlinear model according to the inputs, recursive incorporation of the latest measurements, etc.
Then comes the state reconstruction stage, or SoH estimation. The parameters previously estimated are used with a Markovian form of the model, the data and the command to estimate the current state of the system. During this procedure the parameters of the model can be further adjusted. Particle filters can be used at that stage if the proper hypotheses are formulated: nonlinear non-stationary model, parameters evolving with time, non-Gaussian noise distributions, etc.
Finally prognostics is performed with a procedure adapted to the case. Let's pursue now with the advantages to include particle filters in a prognostics' scheme.

2.4. Advantages

Before going deeper in the applications, it is interesting to see how the authors justify, when they do, the use of particle filters in their work. It should allow distinguishing what are the real advantages of applying this tool in the context of prognostics.
All the following items are direct quotes from the literature. Particle filter is an efficient prognostics tool as:

- recursive Bayesian algorithms are well suited to solve the problem of real-time estimation since they incorporate process data. It allows information from multiple measurement sources to be fused to a logical manner [75];
- it is a MC method for nonlinear Gaussian state space models [75];
- it is capable of accounting for the stochasticity of the process and the noise affecting the measurements [75];
- it can be used to estimate and adjust the model parameters and to track (the battery aging) processes with nonlinear and non-Gaussian characteristics [71];

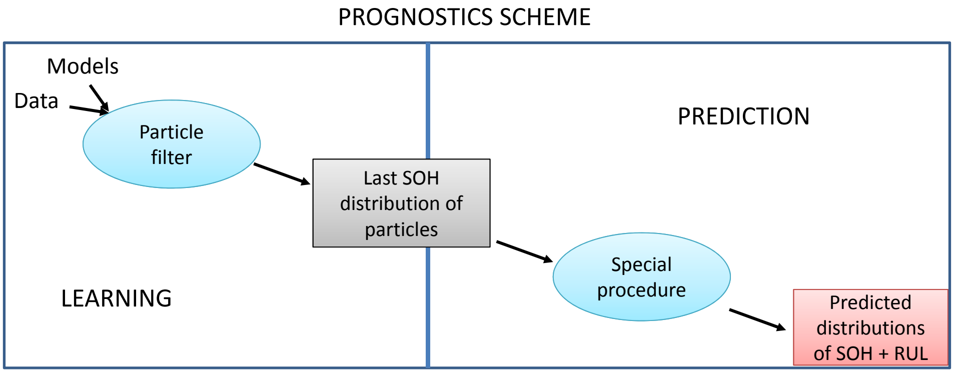


Figure 3: Particle filtering-based prognostics

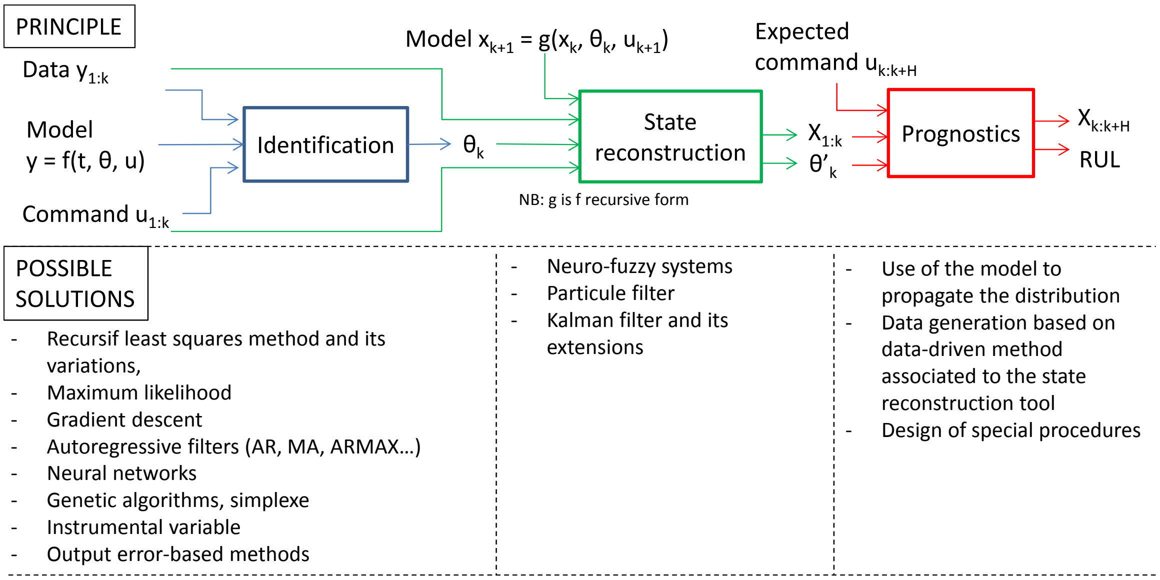

PRINCIPLE
Data $y_{1:k}$
Model
$y = f(t, θ, u)$
Command $u_{1:k}$
Identification
$θ_k$
Model $x_{k+1} = g(x_k, θ_k, u_{k+1})$
State reconstruction
NB: g is f recursive form
$X_{1:k}$
$θ'_k$
Expected command $u_{k:k+H}$
Prognostics
$X_{k:k+H}$
RUL
POSSIBLE SOLUTIONS
- Recursif least squares method and its variations,
- Maximum likelihood
- Gradient descent
- Autoregressive filters (AR, MA, ARMAX…)
- Neural networks
- Genetic algorithms, simplexe
- Instrumental variable
- Output error-based methods
- Neuro-fuzzy systems
- Particule filter
- Kalman filter and its extensions
- Use of the model to propagate the distribution
- Data generation based on data-driven method associated to the state reconstruction tool
- Design of special procedures

Figure 4: Different stages of model-based prognostics and some possible methods for each one

- the parameters of the model can be included as a part of the state vector performing model identification in conjunction with state estimation [56, 71];

- it is applicable to nonlinear and/or non-Gaussian process [12];

- it generates a probabilistic output which is convenient to represent and manage uncertainty [12];

- it supports information fusion from multiple observation sources in a principled manner [12];

- the implementation of SIS helps reducing the number of samples required to approximate the future state probability distribution compared to classical MC methods, increasing the computational speed and efficiency [1, 75].

- its attractiveness relies in the framework provided for handling the significant levels of uncertainty inherent in the generation of long term prediction [10];

- this technique has the ability to tune non-stationary model parameters simultaneously with state estimation, which combination with the representation of the state space as multiple weighted particles, makes it ideal for state tracking and prediction [56];

- it is able to dynamically adjust model parameters of non-stationary conditions and predict the unknown parameters by tracking historical data [66];

It is interesting to notice that only the three last arguments mention a prediction but they are not directly related to prognostics. It confirms the conclusion of the previous subsection.
The argument of applicability with combined parameters

and state identification seems to be the most important if we refer to the number of times it is mentioned.

2.5. Drawbacks and limitations

Fewer references show the drawbacks of particle filters:

- it requires that the state evolution models are accurately updated at initial prediction step and their parameters are approximately static in the future. The prediction results greatly depends on the state particles estimated at initial prediction step, which increases the uncertainty [12];
- there is no update of the model parameters during the prediction [44];
- the contribution of the model uncertainty to the RUL prediction uncertainty is not directly considered, since it is assumed that the degradation dynamics model and observation equation are exactly known [6].
- a potential issue with any non-deterministic prediction technique is that any errors or approximation in the initial pdf can accumulate and grow over a certain time horizon and can severely distort the predicted pdf over a long time frame [62].

Another point raised in [59] is the cases where the particle filter should or should not be used. Particle filters should be used with multivariate or non-standard posterior distribution, nonlinear process, non-Gaussian noise and when a relatively accurate and precise RUL estimate is required. However, it should not be used with a typical deterministic posterior distribution, a linear Gaussian noise, a multiplicative noise, a single variable posterior distribution or if covariate data is not available for the failures of interest. These constrains do not forbid the use of particle filters however they are more suited for Kalman filters or its derivative. Nevertheless, a lot of applications with state equations linear according to the state and with Gaussian noise can be found in the literature as shown later.

These considerations do not integrate implementation issues. Yet, the main problems come with that practical part. First, the assumption of 1^{st}-order hidden Markov model (HMM) is not generally true and a high order model may not be appropriate as it depends on multiple p-step before states [11]. A solution can be to use a high-order particle filter as it will be described later. Another possibility is to introduce the hypothesis that the impact of the m previous states on the degradation is all contained in the last state. It allows approximating the m-order HMM by a first order one.

In [66], the authors recognize in their approach (in this case AFSA-PF), the updating of the parameters is extremely time-consuming which impedes the popularization of the proposed method in engineering applications. It echoes the challenges of parallelization and real-time implementation mentioned in Section 2.2.3.

The common choice of importance density $q(.) = p(x_t|x_{t-1})$ is appropriate to estimate the most likely probability distribution according to a particular set of measurement data. However, according to [63], it does not offer a good estimate of the probability of events associated to high risk conditions with low likelihood. A possible solution is to use a risk-sensitive particle filter (RSPF) that ensures the existence of particles in the tails of the state pdf. This filter will be described in Section 4.1.

Finally, although particle filter-based algorithms have been established as the *de facto* state of the art in failure prognostics, there is still no clear indication about how to perform verification and validation of these approaches [63].

2.6. Open issues

In [62] a certain number of open questions are highlighted:

1. how do we verify the particle filter modules implemented for a particular application?
2. how many particles are needed to accurately represent the uncertainty propagation in the long term predictions?
3. how do we accommodate model parameter uncertainty within the particle filter framework?

These questions are all related to burning issues in prognostics:

1. verification and validation of the approaches;
2. uncertainty representation, quantification and management.

This last point is discussed in Section 7.2.

2.7. Partial synthesis

The main point of this section to keep in mind is that the particle filter is not suitable to perform prognostics but only SoH estimation. Starting from this conclusion, it becomes easier to define why it should be included in a prognostics and what its role is.

As a component of a prognostics scheme, it is expected from the particle filter to give precise SoH estimates based on the current data. This implies taking into account the aging of the system, its behaviors as well as rare and sudden events such as changes in the mission profiles, maintenance interventions, uncontrolled rejuvenation phenomena or unpredictable / sudden failures.

For this purpose, we take advantage of (1) its possibility to fuse multiple measurement sources in a logical manner, (2) its ability to deal with nonlinear non-stationary and/or non-Gaussian processes, (3) its ability to take into account the stochasticity of the process, (4) the possibility to perform parameter and state estimations simultaneously, (5) the probabilistic form of the output and finally (6) its computational efficiency with respect to other MC methods.

However, by integrating this tool in the prognostics scheme, we have to face with its drawbacks. The use of particle filters requires a precise initialization and assumes that the degradation dynamics and the observation equation are exactly known. Also, it might be use without the respect of the first-order HMM hypothesis. It should not be forgotten that the classical choice of the importance density as $p(x_t|x_{t-1})$ may not always be appropriate, that the choices of parametrization can be complicated. Adding to that, the algorithm can be time-consuming so it is necessary to know exactly what the maximum delay expected to obtain results is. Finally, the aspects of uncertainty representation, quantification and management can be tricky. We will see later that we have to precisely define what kind of uncertainty can be handled by particle filters.

3. Modelling

For the record, the particle filter needs a Bayesian formulation of the problem:

$$x_t = f(x_{t-1}, u_t, \omega_t) \leftrightarrow p(x_t|x_{t-1}) \tag{14}$$

$$y_t = h(x_t, v_t) \leftrightarrow p(y_t|x_t) \tag{15}$$

To respect this standard and for an optimal use, the modeling of the system may need some adaptations.

3.1. State model

Using directly degradation models with the particle filter might not be practical in most of the cases. A model adaptation is needed.

First, to respect the Bayesian formulation, the model has to be transformed to express x_t as the function of the previous state x_{t-1}. It is interesting to mention that one aspect to justify the use of particle filter is its ability to deal with non-linearity. However, in a lot of works the proposed state model is clearly linear with respect to the state such as in [12, 16, 57] and in most of these cases have a Gaussian noise.

Then the parameters of the state and observation models to identify should be included to the state vector. Their update equations have also to be defined. Two cases appear here:

1. the parameters are constant through the prognostics process;
2. the parameters are time-dependent and vary all along the prognostics process.

Different solutions emerge according to the dimension of the state vector and the knowledge about the parameters to identify, written α_t.

In [57], a solution is proposed based on the assumption that the system health state is 1-dimensional given by x_t. The authors assume that the parameter values of h in equation (15) are known. However, they precise that this assumption can be relaxed in a more generic approach. Consequently, their main issue is to formulate the state equation for their state parameters α_t. An easy solution proposed is the Gaussian random walk:

$$\alpha_{j,t} = \alpha_{j,t-1} + w_{j,t-1} \tag{16}$$

with $w_{j,t-1} \in \mathcal{N}(0, \sigma_j^2)$. The particle filter will converge to the actual value according to the law of large numbers. The variance σ_j^2 may be chosen to be high in order to cover more state-space but it can delay the convergence. A solution is to define the noise variance σ_j^2 as a state variable that increases if the associated weight is lower than a preset threshold:

$$\sigma_{j,t} = c_{j,t}.\sigma_{j,t-1} \begin{cases} c_{j,t} < 1 & \text{if } w_{t-1} > w_{th} \\ c_{j,t} = 1 & \text{if } w_{t-1} = w_{th} \\ c_{j,t} > 1 & \text{if } w_{t-1} < w_{th} \end{cases} \tag{17}$$

This approach can be computationally expensive so not feasible for all the parameters of a high order model. In this case, the authors in [57] recommend performing a sensitivity analysis and selecting only the most sensitive parameters for a noise variance update.

The Gaussian random walk previously mentioned remains the most used method as in [5, 15, 27, 45].

A totally different approach consists in not using a state model in its classical formulation. In [29], a relevance vector machine (RVM) regression is used to replace of the classical transition function and generate the particles.

3.2. Measurement model

In some cases the measurement model may not be available. This problem is tackled by the authors in [4]. First, an important hypothesis of the method is that the measurement model can be written:

$$y(t) = f(t) + \nu(t) \tag{18}$$

where f is a biunivocal function and ν a Gaussian noise. Then, the method is based on the use of bagged ensemble of artificial neuron networks (ANNs) which are employed to build an interpolator of the available training patterns and the distribution of the measurement noise. This gives the pdf $p(y_t|x_t)$ to the particle filter.

3.3. Uncertainty management in the model formulation

For uncertainty management, the authors in [62] consider two main types of adjustments:

1. adjustments in the unknown parameters in the state equation;
2. adjustments in the hyper-parameters that define the noise pdf.

The second option can be considered thanks to an outer correction loop that modifies the variance of the measurement noise:

$$var\left\{w_2(t+1)\right\} = \begin{cases} p * var\left\{w_2(t)\right\} & \text{if } \frac{||pred-error(t)||}{||feature(t)||} < Th \\ q * var\left\{w_2(t)\right\} & \text{if } \frac{||pred-error(t)||}{||feature(t)||} > Th \end{cases} \tag{19}$$

$pred-error(t)$ is the short term prediction error computed at time t (a 5-step prediction error in the illustrative example provided in [62]), $||\ ||$ is a norm, the L_2 norm is proposed by the authors, p, q and Th are constants such that $0 < p < 1$, $q > 1$ and $0 < Th < 1$.
This type of approach raises a question: is that correct to control the noise variance so precisely? Indeed, it may change the interpretation of the resulting uncertainty (see section 7.2).

3.4. Specific architectures for modeling

In some applications both models and measurements are adapted by an integrated architecture. In [72], the state transition equation is based on a NARX (nonlinear autoregressive exogenous model) model while the measurement equation is SVM-based ARMAX (Auto Regressive Moving Average with eXternal inputs) modeling.

3.5. Partial synthesis

Some questions remain, most of them regarding the state model.
First, whatever the approach chosen to update the parameters of the model, a choice has to be made regarding the variance of the noise in equation (16). The question is: how to define a proper range for the variance faster than with a trial and error approach? Indeed, the trial-error technique is currently the simplest manner to define the variance when no precise knowledge is available. Nevertheless, it can very fast become time-consuming.
Then the problem of a high dimension state vector has not been addressed yet. Indeed, the state model main contain a great number of parameters to identify. The idea of performing a sensitivity analysis to determine on which parameters focusing is a good starting point. However, what can be done with these parameters? Moreover, according the stage of aging of the system, the most sensitive parameters may differ with the time period. It might imply that their update equation evolves in time. A possible solution is to include an inner sensitivity analysis in the prognostics scheme just before the particle filter as proposed in Figure 5.

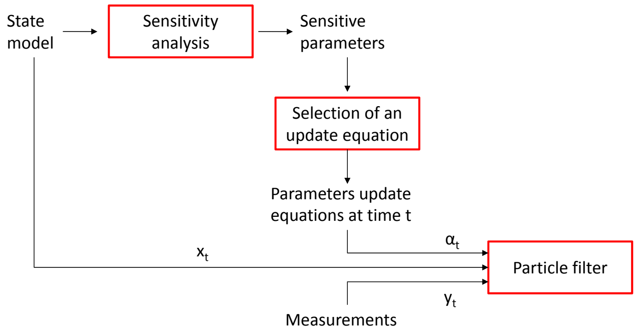


Figure 5: Adaptive selection of the parameters' update equation

Then regarding the measurement model, a solution is proposed for the cases where this model is not available. However, this solution involves a Gaussian assumption. Other options to tackle the problem should be proposed without implying this hypothesis. This will be discussed again in Sections 5.4 and 5.5.
Finally, it is hard to comment the suggestions proposed for uncertainty management. Indeed before discussing this point, it is important to define the type(s) of uncertainty handled by the filter. Section 7.2 will be dedicated to that topic.

4. Type of filter and resampling procedures

Once the models are set, the question "what filter should we use?" comes. There are currently no guidelines to help practitioners choosing. Traditional and existing types of particle filters can be used for prognostics, as well as existing resampling procedures (see Tables 4 and 5). However, some papers propose new ideas.

4.1. Types of filters

A great majority of the prognostics' works do not indicate what filter they used. When they do, it can be noticed that the SIR seems to be the most employed. Other types can be encountered. It is interesting to see that all of them were already existing in the general literature (see Part I, Section 2.2.3).

Table 4: Type of particle filter used in the prognostics' literature

Type	References
SIR	[2, 5, 6, 15, 20, 23, 32, 37, 51, 56, 63, 66, 72, 73]
UPF	[14, 45]
RSPF	[48, 63]
RPF	[69]
Adaptive PF	[35]
AFSA-PF	[66]

Let's focus on the six types of filter encountered in particle filtering-based prognostics and try to understand why they were selected among the huge existing literature.

The sampling importance resampling filter - SIR. The SIR is the most classical form of particle filter and is easy to implement. It is clearly described in [3, 13]. The key points are explained here.
The attractiveness of the SIR relies in the fact that the assumptions required to use the filter are very weak. The state dynamics and measurement functions need to be known, and it is required to be able to sample realizations from the process noise distribution and from the prior [3]. Also, the availability of the likelihood function is needed for pointwise evaluation. This algorithm can be easily derived from the SIS described in Part I by the appropriate

choices of (1) the importance density (chosen to be the prior density) and (2) the resampling step (applied at every time step). For more details on the algorithm, please refer to [13].

One advantage of the algorithm is that any resampling procedure can be integrated. Moreover the choice of the prior density as importance density is very convenient. However, the importance sampling density for the SIR is independent of measurement, so it means that the state space is explored without any knowledge of the observations. Consequently, this filter is sensitive to outliers and can be inefficient. Adding to that, as resampling is performed at every iteration, this may result in a rapid loss of diversity in particles.

The unscented particle filter - UPF. The UPF algorithm was first introduced in [68]. It uses the unscented Kalman filter (UKF) to generate a proposal distribution for getting the posterior probability taking account of the latest observation. UPF is divided into two steps:

1. applying the UKF algorithm to get the proposal distribution;
2. using the standard particle filter algorithm to get the results.

The key point in the UPF is its ability to use the latest observation, contrary to the SIR.

Risk sensitive PF-based prognostics. This type of filter incorporates a cost model in the importance distribution to generate more particles in high-risk regions of the state space. It was first introduced in [64] and then encountered in [48] for prognostics purpose. According to the authors of the last paper, it is critical to note that this particle filter approach makes use of exogenous models to evaluate and estimate the risk associated with every fault mode. In practice, RSPF implies a more conservative estimate of the RUL of a piece of equipment. It is expected that RSPF-based routines will anticipate possible sudden changes in the system's operating conditions.
In simple words, RSPF incorporates a model of costs when generating particles. By incorporating a cost model, states that are more critical to the system performance are more likely to be tracked. In [48], the approach proposes to use a variant of RSPF algorithm, where the cost function allows sampling particles from regions of the state space representing high-risk conditions for the system and where the fault dimension has low likelihood. It is implemented by modifying the kernel of the process noise $w(t)$:

$$w(t) \approx \delta w'(t) + (1-\delta) w^*(t) \tag{20}$$

where $w'(t) \sim N(0, \sigma'^2)$, $w^*(t) \sim N(d, \sigma^{*2})$, $d = Ew^*(t) \neq 0$, σ' and σ^* are the variances of Gaussian kernels and $0 \leq \delta \leq 1$. The parameters process noise is corrected thanks to the inner correction loop proposed in [50] that will be explained later.

The regularized particle filter - RPF. When using the RPF, the idea is to change the discrete approximation of the posterior pdf to a continuous one in the resampling stage with a rescaled kernel procedure [69]:

$$p(x_t|y_t) = \sum_{i=1}^{N} w_t^i K_h(x_t - x_t^i) \tag{21}$$

with

$$K_h(x) = \frac{1}{h^{n_x}} K(\frac{x}{h}) \tag{22}$$

where $K(.)$ is the recalled kernel density and h is the kernel bandwidth.

The adaptive particle filter. The self-adaptive particle filter was proposed by [60]. It gives to the particle filter the ability to adjust the number of particles as well as the propagation function at each iteration.
The fact behind the idea of adjusting the number of particles is that the use of a fixed number of particles is often inefficient [60]. The dynamics of the system's processes introduce a great variability in the complexity of the posterior distribution. So the initial number of particles can be much larger than the real number of particles needed to perform a good estimation of the posterior distribution. On the opposite, it can be too small causing the divergence of the filter. The method developed to adjust the number of particle starts from the KLD-sampling algorithm that can be found in Part I - Table 2. This kind of procedure is expected to give better state estimates and reduce the computational load when the number of particles is overestimated.

Artificial fish swarm-based particle filter - AFSA-PF. Artificial fish swarm algorithm (AFSA) was initially proposed as a new random searching optimization algorithm based on the simulation of fish swarm in the wild. This algorithm imitates four behaviors of fish swarm: (1) preying behavior to find the optimal solution, (2) swarming behavior to gather the fish swarm in the optimal region, (3) following behavior to free the fish swarm from local optimal solutions and (4) random behavior to find the optimal solution in a larger scope. It has the advantages of rapid and global optimization, insensitivity to initial values, robustness and easy operation [66]. This algorithm is used to drive the particles to the region of high likelihood. We will come back to its use in prognostics application in Section 5.6.

4.2. Resampling procedures

We previously saw that a great number of resampling procedures exist. However, it seems that the authors of prognostics applications stuck to the simplest ones. Very few share the procedure employed but we can find some novelties comparing to general literature.

Table 5: Type of resampling procedure used in the prognostics' literature

Type	References
Inverse CDF method	[2, 20, 72, 75]
Bootstrap resampling algorithm	[5]
SVR	[18]
Monotonic resampling	[72]

We can find minor changes as for example in [5] where the importance weights are defined slightly differently than usual:

$$w_t^i = \frac{L.w_{t-1}^i}{\sum L.w_{t-1}} \tag{23}$$

with L being the likelihood. But new types of resampling can be found in [18] and [72].

The resampling procedure proposed in [18] is based on support vector regression (SVR). The idea of rebuilding the posterior distribution by SVR is an optimization problem using a regularized function with constraints. When the effective sample size (ESS) falls below the threshold, a resampling of the posterior distribution using the SVR algorithm occurs. The training pairs are the particle x_t^i and its corresponding weight $w_t^i = F(x_t^i)$. These pairs are used to rebuild the resampling posterior distribution.

In [72], the authors introduce a monotonic resampling scheme. Indeed, they highlight that even with a monotonic degradation trend, with SMC methods monotonic results are not guaranteed. So to keep only the monotonic behavior estimates, they modify the particles weights according to:

$$w_{k+p}^i = \begin{cases} w_{k+p}^i & \text{if } x_1^i(k+p) > b_{k+p} \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

When resampling is also an interesting question. While some are using the ESS as above in [18], other set directly the frequency of resampling as in [6] where resampling occurs every 5 measurements.

The impact of the resampling on the uncertainty is also questioned. Is there still any uncertainty present at the initialization of the filter in the resampled distribution? And for the resampling based on a variance's threshold (as ESS), is that correct to control so precisely the variance of the distribution? Is that relevant to help interpreting the results? Answering these questions is far from being trivial.

4.3. Comparison of resampling approaches

One paper attempting to compare prognostics results obtained with several resampling approaches is available [25]. The resampling algorithms considered are: multinomial, residual, stratified and systematic resampling. According to the results presented, systematic and stratified resampling show the best results with a slight advantage to systematic resampling which is proved theoretically superior.

The idea of comparing different algorithms to provide guidelines for prognostics is interesting. However, this paper suffers from several limitations. First, the experiments are made on a very simple battery modeling. How would the results evolve if the experiments are made on a completely different system with a more complex state model? How would they change with a different initialization of the particle filter? Can we draw the same conclusion if we change the number of particles (initially 500)? Indeed drawing any conclusion when comparing these algorithms in a prognostics perspective can be difficult and need further investigation as well as precise evaluation methods.

4.4. Partial synthesis

How do we choose a particle filter adapted to a specific application? That is the question of this section. Choosing based on the results provided in the PHM literature can lead to a biased opinion. Indeed, when improved particle filters are used, they are always compared to the most basic form (i.e. the SIR) or to a version of the Kalman filter. These types of comparison prove nothing as evolved particle filters were designed to outperform these classical approaches and this was already demonstrated in the dedicated literature. Also, practitioners have to wonder if it is really necessary to implement evolved particle filters instead of sticking to the simplest ones. Indeed, implementing a more sophisticated filter may not necessary improve the results significantly and can be computationally costly. The gain is not always obvious.

It seems that starting with the most basic version, the SIR, is the preferred option in the PHM community. This appears to be a good starting point. However, it should be modified to incorporate more knowledge of the system and also to give fast and reliable prediction. Incorporating the knowledge available on the system has not be seen yet. It may seem strange in the context of model-based PHM. Indeed, to build a model a deep understanding of the system is required and some information that is not integrated in the modeling can be injected elsewhere in the filter. For instance, in application using a health indicator which is supposed to be monotonic, for example increasing, eliminating a particle that shows a decreasing trajectory looks logical. This would lead to a biased particle filter but clearly adapted to the state estimation of a particular system.

An interesting idea would be designing a filter that takes all the advantages of the proposed filters and incorporate new ideas. Such a filter would be able to:

- create different groups of particle: one for the state and others for parameters estimates;

- adapt its number of particle by itself (adaptive particle filter);

- take into account sudden events that affect the system state if necessary (risk sensitive particle filter);
- perform a resampling to obtain a monotonic health indicator estimation (monotonic resampling);
- constrain the particle values only in the part of the state space coherent with the age of the system (constrained state space [67]).

Such a filter may offer good performances but the algorithm might be time-consuming. The user should select the appropriate improvement according to his system and application.

5. Implementation and practical issues

Once the type of filter chosen, the implementation of the filter can start. According to [20], a particle filtering-based approach for prognostics has 6 steps:

1. model definition;
2. parameter initialization;
3. parameter sampling and prediction;
4. dynamic update;
5. particle weighting and resampling;
6. RUL prediction

The distinction between points 3 and 4 is not obvious, they could be gathered in a same step. With respects to previous discussions and to figure 3, the implementation of the SoH estimation, i.e. items 1 to 5, is proposed in this section. While the prognostics procedure will be described in Section 6.
This section will discuss the following issues:

1. the formulation of problems breaking the assumption of 1^{st}-order Markov process;
2. the choice of the number of particles;
3. the definition of the process and measurement noises;
4. the formulation of the likelihood function;
5. the importance sampling;
6. the introduction of inner correction loops to improve the performance.

Let's start with the issue of breaking the assumption of 1^{st}-order Markov process.

5.1. Bayesian estimation using m-order HMM

The authors in [11] raise the issue of high-order HMM. The problem formulation is the following:

$$\begin{cases} x_t = f_t(x_{t-1}, ..., x_{t-m}, w_{t-1}) \\ y_t = h(x_t, \nu_t) \end{cases} \tag{25}$$

They demonstrate that the formulation of the weights remains the same:

$$w_t^i = w_{t-1}^i.p(y_t|x_t^i) \tag{26}$$

For simplicity the demonstration is not reproduced here and the reader may refer to [11].
However building a state model that uses the p-step before states can be complicated. So an approach using ANFIS is proposed for state modeling [11]. It uses the four last states x_{t-1} to x_{t-4} as inputs and gives an estimate of the updated state $\hat{x}_t$ and the process noise w_{t-1}. This process noise follows a Gaussian distribution which initial mean and variance are generated via ANFIS's modeling error. The different stages of the procedure are:

1. the ANFIS is trained with available condition data to model the fault propagation process;
2. the fault growth model represented by the ANFIS and the process noise is employed with a 4^{th}-order particle filter to draw a set of particles. One-step ahead and multi-step ahead predictions are made;
3. the process noise is updated with:

$$\mu_w = \frac{\sum_{i=0}^{n-1} y_{x-i}}{n} \tag{27}$$

$$\sigma_w = \sqrt{\frac{\sum (y_{x-i} - \mu_w)^2}{n}} \tag{28}$$

with y_{x-i} a residual;
4. repeat 2 and 3

5.2. Defining the number of particles

Choosing the right number of particles is far from being simple as no clear methodology is proposed. The authors in [57] demonstrate that the convergence of the filter is dependent of the number of particles N but not of the dimension of the state n_x. They support their discussion with figures reproduced from [17] and presented here in Figure 6.

If we look at the different works, we notice that the number of particles ranges from 30 to 10 000 (Table 6). These values seem consistent with Figure 6. However, such diversity does not help choosing a standard number at all. A possible way to overcome the problem is to start with the number of particles prescribed by Figure 6 and then use an adaptive particle filter to adapt this number.

Table 6: Number of particles chosen for experiments

References	Number of particles N
[74]	30
[11]	100
[66]	200
[15, 25, 37, 72]	500
[5, 6, 16, 44, 75]	1000
[35]	2000
[12]	1500
[2]	between 1000 and 5000
[28]	1000, 5000 and 10000

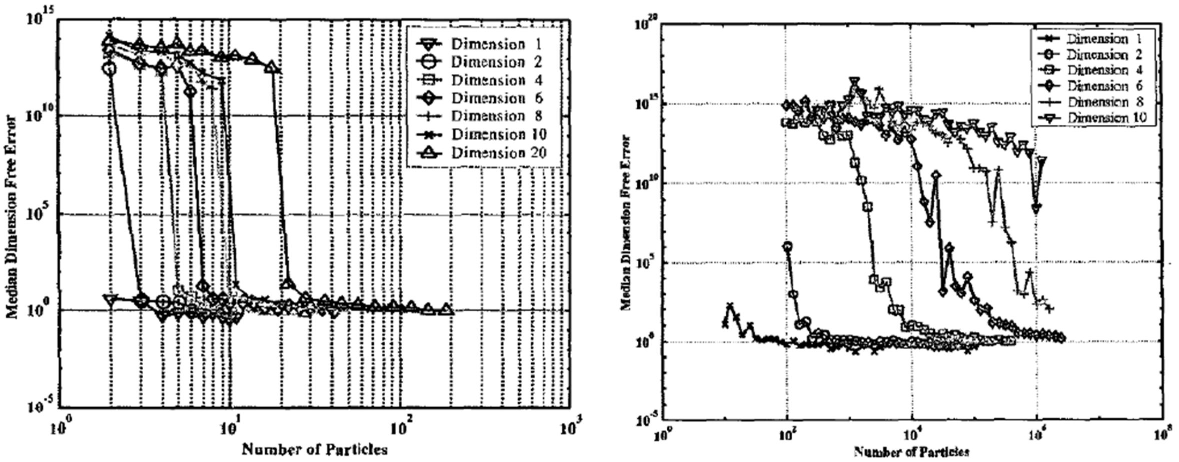


Figure 6: Left: Dimension free error vs. number of particles for PF with good proposal density, Right: Dimension free error vs. number of particles for PF with poor proposal density [17]

5.3. State and parameter initializations

Choosing a correct initialization of the filter is crucial to ensure the particles convergence toward the true state. Initializing the particle filter consists in two main steps:

1. defining the initial distribution of the system state;
2. defining the initial distributions of the parameters of the models;

However, these steps are often unmentioned in the literature.

5.3.1. Initial distribution of the system state

As stated in section 2.2.2, the initial distribution should be built according to the known state of the system. In the PHM context, the system is continuously monitored until its early life or until a fault alarm. So normally, the initial state of the system is known with a quantified uncertainty. However, there is always a possibility that the system is more degraded as expected and this may not appear on the measurements.

Consequently, a possibility for the initial distribution is combination of a Gaussian distribution with a uniform one as proposed in Figure 7. The Gaussian would be centered on the assumed present state of the system, with a standard deviation σ linked to the uncertainty on the measurements. The uniform would be defined according to the knowledge of the degradation, particularly its kinetics. If these kinetics are very slow and the risk that the system is more degraded than expected closed to zero, the uniform distribution could be ignored and the initial distribution of the state would be only a Gaussian.

5.3.2. Initial distributions of the parameters of the models

One of the major advantages of the particle filter is its ability to update the parameters of a model while estimating the state. A practical manner to do that is to incorporate the model parameters to the state vector and update them via their own updating equation. Consequently, a proper initialization is required. A bad initialization of the state model surely leads to a poor state estimate. There is no standardized method to initialize the parameters distribution. The general literature dealing with particle filters remains very vague on that subject. However some interesting proposal have started to appear in the PHM community.

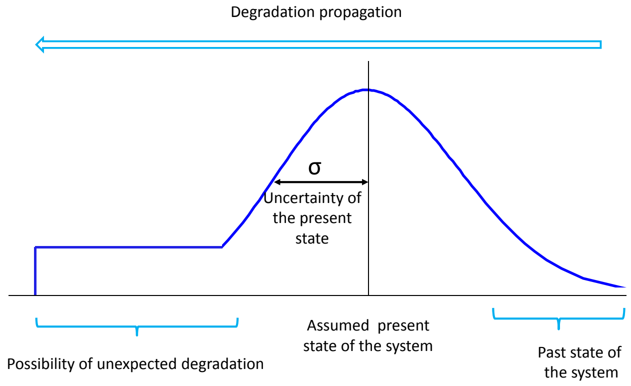

Degradation propagation
σ
Uncertainty of the present state
Assumed present state of the system
Possibility of unexpected degradation
Past state of the system

Figure 7: Proposal of initial state distribution

When expert knowledge is available, defining precise initial distribution is possible. It is illustrated in [6] by the use of different types of distributions: Gamma, deterministic and Normal. However, it does not seem that this case is the most common one.

In [2], as no prior information on the initial distributions of parameter is available, the choice to use uniform distributions is made. However, there is no explanation about how the ranges of these distributions are defined. Uniform distributions are also adopted for the initial state, parameter values and standard deviation error of the measurements in [10] centered on estimation obtained by fitting for the first two and a bootstrap procedure for the last. Other examples using uniform distributions can be found in [28, 32, 69, 74].

Fitting the data available for learning seems to be the most classical approach. Such idea can be found in [10, 14, 20, 23, 27, 33, 37, 44, 45, 71, 74]. Some examples can be provided. In [20], initial distributions are uniform and come from a nonlinear least square regression. In [14], a curve fitting is performed thanks to the dedicated Matlab toolbox and the initial value of the parameters is the mean of the fitting estimates of three datasets. The Matlab fitting toolbox is also used in [45]. In [74], a least square algorithm with a forgetting factor is employed to determine the model parameters. To initialize the parameters of the model, the authors in [71] use the average value through curve fitting based on training samples: both linear and nonlinear least square fitting are used according to the different parameters. They also state that initializing the parameters' variance is time consuming and decide to fix $[min(\gamma_i), max(\gamma_i)] = 6\sigma$ with an equal variance $\sigma = 10^{-5}$ for all their parameters.

However, these simple approaches are not always efficient and should be combined for improvements. In [27], the authors show that using the least square approach alone has as major limitation: when historical data coming from different systems are used, different values of the parameters appear according the context of use and aging and it might lead to a bad initialization of the particle filter. Consequently, their approach has two steps. First a Gauss-Newton algorithm is used to fit the model to different subsets of the data. Then a mixing combination rule of the Dempster-Shafer theory (DST) is used to obtain an initial value of each parameter based on the weighted sum of the parameters obtained from the DST. These values are then used to initialize the particle filter with Gaussian distributions.

Then, in [23, 54, 53], the particle filter is combined with relevant vector machines (RVM). The RVM is used to find representative aging curves and fit them to the data to perform parameter identification. It helps them defining the mean and the standard deviation of their initial Gaussian distributions.

Finally, in [9], the data is modeled as Gaussian mixture models. The problem is to estimate the statistical parameters for each Gaussian. This time another approach is proposed: the Figueiredo-Jain algorithm, as it is more adapted to such a type of model.

5.3.3. Impact of initialization on the results

An interesting comparison is performed in [6] to show the importance of the initial distribution on the spread on the final distribution. In a first test, it is assumed to known the exact value of the parameters while in a second one only a distribution is known. The second test corresponds to the introduction of uncertainty on the model. The results of the comparison are proposed in Figure 8. Unsurprisingly, the results are better for the first test, showing more accurate prediction with narrower final distributions.

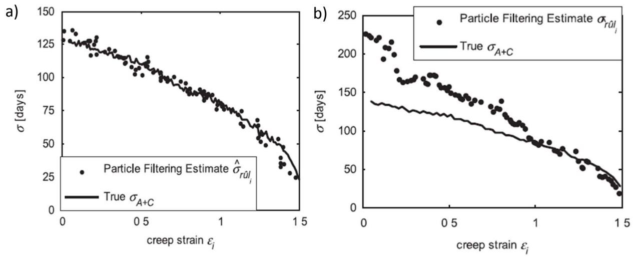


Figure 8: Comparison of predictions proposed in [6] using exact model parameters (a) or an approximated distribution as initialization (reproduced from [6])

5.4. Noise definition

For both process and measurement, noises intervene in the model. An interesting fact regarding noise definition is that almost all papers justify the use of particle filter because of its ability to deal with non-Gaussian noises. However, Table 7 shows that a great majority are using this kind of noise distribution.

Table 7: Noise distributions chosen for experiments

Reference	Process noise	Measurement noise
[9, 12, 16, 23, 25, 27, 56, 57, 66, 72, 73]	zero-mean Gaussian	zero-mean Gaussian
[6, 10, 35]	not precised	Gaussian
[11] [5]	Gaussian not Gaussian	not precised not precised

In [2], the authors introduce an interesting hypothesis: the process noise can be ignored because it can be handled through the uncertainty in the model parameters. We can go further with this hypothesis. Indeed for prognostics what is often needed is a global trend of the aging indicator not the precise. The noise is here to represent the error that exists between the model and reality. And it is of a very small order of magnitude. Consequently, as general trends and not precise models are expected, we consider that this noise can be ignored in a great majority of prognostics applications.

Regarding the measurement noise, the Gaussian hypothesis is almost always used as a default option. It is very convenient as it allows deriving a simple expression of the likelihood. But it is never mentioned that a precise study of the data was performed to support that hypothesis and to define the characteristics (more precisely its standard deviation) of a Gaussian distribution. If the Gaussian hypothesis is made but without the knowledge of the standard deviation, it is possible to include it to the state vector and let the filter estimating it as proposed in [2].

To conclude, defining the measurement noise requires a great work on the data. It could be interesting to be able to bypass this stage and find another approach to define the consequent likelihood.

5.5. Likelihood definition

The expression of the likelihood directly comes from the measurement equation. As stated in [2], for a measurement function: $y_t = x_t + \nu_t$ with a Gaussian noise: $\nu \sim \mathcal{N}(0, \sigma^2)$, the likelihood function is written:

$$L(y_t|x_t^i, \theta_t^i, \sigma_t^i) = \frac{1}{\sqrt{2\pi}\sigma_t^i} exp\left[-\frac{1}{2}\left(\frac{y_t - x_t^i\theta_t^i}{\sigma_t^i}\right)^2\right] \quad (29)$$

However, such common formulation only takes into account the latest prediction. The authors in [29] propose a likelihood of the whole trajectory for each particle, keeping the Gaussian assumption form:

$$L(y_t|x_t^i) = \sum_{t=1}^{T} \frac{1}{\sqrt{2\pi\nu_t}} exp\left[-\frac{(y_t - x_t^i)^2}{2\nu_t^2}\right] \quad (30)$$

These are examples with Gaussian measurement noises, however no other expression is proposed it the PHM literature.

5.6. Importance sampling

Driving the particles to high likelihood regions is a classical mean to improve the quality of the results. Several solutions are proposed in the literature.

5.6.1. AFSA-PF

Artificial fish swarm algorithm based particle filter (AFSA-PF) has already been proposed as an alternative to the classical particle filter (see Section 4.1). The authors in [66] propose to apply AFSA-PF in the context of prognostics. The combination of AFSA and PF is applied in the selection of the objective function and the adjustment of the importance weights. AFSA drives the particles move to the high likelihood areas based on an objective function Y (i.e. the measurement) defined by:

$$p(y_k^i|x_k^i) = \frac{1}{(2\pi\lambda^2)^{\frac{1}{2}}} exp\left[-\frac{1}{2\lambda^2}(y_k - y_{k|k-1}^i)^2\right] \quad (31)$$

5.6.2. PF-KS

In [28], a procedure to move the particle using kernel smoothing (KS) is explained. KS is based on the application of two stages: shrinkage and perturbation.

1. shrinkage moves the particle parameter values toward their estimated value:

$$\tilde{p}_t^i = p_t^i\sqrt{1-h^2} + \hat{p}_t(1 - \sqrt{1-h^2}) \quad (32)$$

   where $\tilde{p}_t^i$ is the new particle value, p_t^i the current particle value, $\hat{p}_t$ the expected value and h is the kernel parameter between 0 and 1 (the authors in [28] suggest 0.1).

2. while perturbation adds controlled noise to maintain the desired variance in the population:

$$p_{t+1}^i = \tilde{p}_t^i + \mathcal{N}(0, h^2\nu(p_t^i)) \quad (33)$$

PF-KS has the advantage to keep unmodified the variance of the parameter values in the particle population.

5.7. Inner correction loops

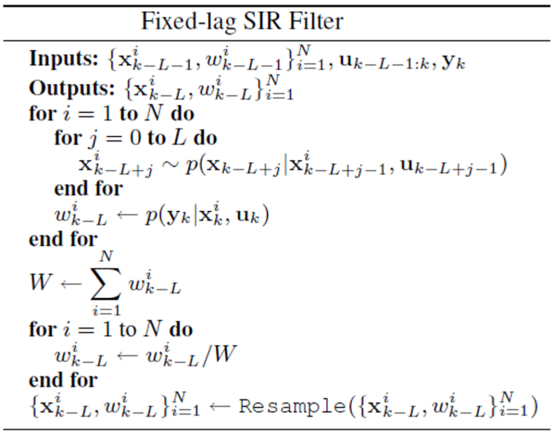
In order to improve the accuracy of the results, the creation of inner correction loops in the particle filter has been proposed with more or less complexity. The main idea is to correct past predictions based on more recent ones.

5.7.1. Fixed-lag particle filter

The principle of fixed-lag particle filter is very simple [15]: the current state estimate x_t is used to correct the previous $t - L$ estimate x_{t-L}. Consequently, the fixed-lag smoothing distribution can be written:

$$p(x_{t-L}|y_{0:t}) = \sum w_t^i \delta(x_{t-L} - x_{t-L}^i) \quad (34)$$

Computing this distribution implies only minor changes in the particle filter as shown in Figure 9. The authors tested different values of L ranging from 0 to 4 and stated that L=2 is optimal for almost the metrics they tested with their data.



```
Fixed-lag SIR Filter
Inputs: ${\{\mathbf{x}_{k-L-1}^i, w_{k-L-1}^i\}_{i=1}^N, \mathbf{u}_{k-L-1:k}, \mathbf{y}_k}$
Outputs: $\{\mathbf{x}_{k-L}^i, w_{k-L}^i\}_{i=1}^N$
for $i = 1$ to $N$ do
    for $j = 0$ to $L$ do
        $\mathbf{x}_{k-L+j}^i \sim p(\mathbf{x}_{k-L+j}|\mathbf{x}_{k-L+j-1}^i, \mathbf{u}_{k-L+j-1})$
    end for
    $w_{k-L}^i \leftarrow p(\mathbf{y}_k|\mathbf{x}_k^i, \mathbf{u}_k)$
end for
$W \leftarrow \sum_{i=1}^{N} w_{k-L}^i$
for $i = 1$ to $N$ do
    $w_{k-L}^i \leftarrow w_{k-L}^i / W$
end for
$\{\mathbf{x}_{k-L}^i, w_{k-L}^i\}_{i=1}^N \leftarrow$ Resample($\{\mathbf{x}_{k-L}^i, w_{k-L}^i\}_{i=1}^N$)
```

Figure 9: Fixed-lag particle filter algorithm [15]

5.7.2. Outer feedback correction loop

The correction loops typically measure the prediction capability of the fault progression model, via short-term prediction error, and improve the algorithm performance either modifying the structure of the model or updating hyper-parameters that define process/measurement noise of model update equation [50].
To introduce the correction loop, the problem statement is modified:

$$\begin{cases} x_{t+1} = x_t + \alpha_F(x_t, \alpha_t) + w_t \\ \alpha_{t+1} = L(\alpha_t, e_t^s) + w_t' \\ y_t = x_t + \nu_t \end{cases} \quad (35)$$

where α_t is a time-variant parameter related to the growth rate of the state x_t and $e_t^s = y_t - \hat{y}_t^s$ is the s-step ahead prediction error with $\hat{y}_t^s = \hat{x}_t^s$.
The first approach is to manipulate the variance of the noise kernel w_t':

$$\begin{cases} L(\alpha_t, e_t^s) = \alpha_t \\ var(w_t') = \begin{cases} p.var(w_t') \text{ if } |e_t^s| \leq e^{th} \\ q.var(w_t') \text{ if } |e_t^s| > e^{th} \end{cases} \end{cases} \tag{36}$$

It allows the algorithm to increase the probability of drawing samples from a broader subset of the domain for the second component of the state vector $[x_t, \alpha_t]$.
The second approach manipulated both the transition function of α_t and the noise variance of w_t'.

$$\begin{cases} L(\alpha_t, e_t^s) = \begin{cases} \alpha_t \text{ if } |e_t^s| \leq e^{th} \\ \alpha_t + \eta.e_t^s \text{ if } |e_t^s| \leq e^{th} \end{cases} \\ \\ var(w_t') = \begin{cases} p.var(w_t') \text{ if } |e_t^s| \leq e^{th} \\ \sigma_0^2 \text{ if } |e_t^s| > e^{th} \end{cases} \end{cases} \tag{37}$$

where η is the feedback gain and σ_0^2 a constant.

5.8. Partial synthesis

Practical issues are numerous with particle filters. First regarding the initialization, a new approach is proposed to initialize the state pdf. For the parameters, different methods have already been tested. It is not possible to prefer one among the others as the approach used may depend of the quantity of data available. For instance, one can apply the DST with n datasets however it is useless if only one dataset is available for initialization.
Then ignoring the process noise is proposed in the state equation when only a trend is needed for prognostics. However, the noise remains a problem in the measurement equation if no precise study of the data is previously performed. An interesting solution would be to forget this equation and to set independently the expression of the likelihood function. Indeed, the likelihood serves to measure the degree of matching between the estimated state and the data. To avoid sensitivity to outliers and give better performance, it should take into account the whole state trajectory. Consequently the following expression is proposed:

$$L_T^i = \frac{1}{\frac{1}{T}\sum_{t=1}^{T} .abs(x_t^i - y_t)} \tag{38}$$

where T is the current time index. The idea behind this proposal is to give the higher likelihood to the particles that show the trajectory the closest to the data. So the likelihood is chosen inversely related to the absolute error of the trajectory of the particle.
To continue, some methods were proposed to drive artificially the particles to regions of high likelihood. Two limits can be found to such practices:

1. it may hide bad procedures for weighting and resampling: improving them can make the moving of the particles useless;
2. by driving the particles to high likelihood regions, a risk of forgetting the states locating in the tails of the likelihood can be created. Moreover, this seems to increase the loss of particle diversity.

Finally, regarding the integration of feedback loops, they appear to bring improved results and not to involve heavy procedure. It seems interesting to consider including one in the particle filter.

6. Prognostics based on the particle filter output

We saw earlier that the particle filter is not able to perform prognostics. But the introduction of an adequate procedure at the end of the filtering process allows making predictions based on the filter's outputs.

6.1. Generation of long term predictions

A critical point for prognostics is: how to use the particle filter to generate long term predictions? Indeed, at the end of the learning no more data is available, preventing the filter to work in its traditional way. Two families of solutions can be found in the literature. Let's call them:

1. particles projection;
2. and creation of artificial measurements.

On one hand, particles projection family uses the last particle distribution of the learning as a starting point to project the particle through different possible pathways in the future. This can be done by simply using the state equation or with more complex procedures as described in Section 6.1.1. On the other hand, the second family uses complementary prognostics tools to generate measurements after the end of the learning and consequently continuously feed the particle filter with new measurements. These approaches are described in Section 6.1.2

6.1.1. Particles projection

Method 1 [75]. The only thing known is the state dynamic model which allows to say that at the future times $j = k+1...k+l$, the state x_{j-1} will pass onto state x_j with the transition pdf $p(x_j|x_{j-1})$. It is necessary to project the initial condition $p(x_k|y_{0:k})$ among all possible future paths weighted by their probability $\prod_{j=k+1}^{k+l} p(x_j|x_{j-1})dx_{j-1}$.
By combining the model equation and the current state pdf estimate, the l-step ahead posterior distribution is obtained:

$$p(x_{k+l}|y_{0:k}) = \int ... \int \prod_{j=k+1}^{k+l} p(x_j|x_{j-1})p(x_k|y_{0:k}) \prod_{j=k}^{k+l-1} dx_j \tag{39}$$

Thanks to the MC pdf approximation this equation can be re-written:

$$p(x_{k+l}|y_{0:k}) = \sum_{i=1}^{N} w_{k+l-1}^{i}.p(x_{k+l}|x_{k+l-1}^{i}) \qquad (40)$$

The whole demonstration to transform the left term in Equation (40) into the right is not reproduced here. Please refer to [75] for complete explanations.
In practice, to obtain (40), an inverse transform resampling procedure of the particle population is performed:

1. draw N values of a uniform random variable in the interval [0,1): $u^i \sim U[0,1), i = 1...N$
2. the generic i^{th} realization x_{k+l}^{i} of $p(x_{k+l}|y_{0:k})$ is obtained by interpolation of the cumulative state distribution:
$$F(X_{k+l} \leq x_{k+l}) = \int_{-\infty}^{x_{k+l}} \hat{p}(x_{k+l}|x_{1:k+l-1})dx_{k+l} \qquad (41)$$
such that $x_{k+l}^{i} = F^{-1}(u^i)$ which is equivalent to x_{k+l}^{i} such that $F(x_{k+l}^{i}) = u^i$
3. the weights of the resampled particles are kept unchanged $w_{k+l}^{i} = w_{k+l-1}^{i}$
4. repeat until T, the horizon of interest

For an illustration of this procedure, the reader can refer to [2].

Method 2 [51]. Long term predictions can be written:

$$\hat{p}(x_{t+k}|\hat{x}_{1:t+k-1}) \approx \sum_{i=1}^{N} w_{t+k-1}^{(i)}.\hat{p}(x_{t+k}^{(i)}|\hat{x}_{t+k-1}^{(i)}) \qquad (42)$$

The weight of every particle should be modified (at each prediction step) to take into account the fact that the noise and process non-linearity could change the shape of the state pdf as time passes. For this reason, the authors distinguish two approaches:

1. the first one is the most classical one and consists in successively taking the expectation of the update equation for every future instant. However, this approach states that the error that can be generated by considering the particle weights invariant for future time instants is negligible with respect to other sources of error (model inaccuracies, assumption on noise parameters). It is the most commonly used. Indeed, long-term prediction based on the projected paths for each particles can be found in [5, 8, 9, 14, 15, 28];
2. the second and novel approach proposes that uncertainty for future transitions may be incorporated by simply resampling the predicted state pdf. The state is then given by the position of the particles, not the weight value.

This second approach is described in detail now. It is based on the assumption of uncorrelated noise and the use of kernel transitions to describe the state pdf before resampling. Let's consider a discrete approximation of (42):

$$\hat{p}(x_{t+k}|\hat{x}_{1:t+k-1}) \approx \sum_{i=1}^{N} w_{t+k-1}^{(i)} K_h(x_{t+k} - E\left[x_{t+k}^{(i)}|\hat{x}_{t+k-1}^{(i)}\right]) \qquad (43)$$

where $K(\bullet)$ is a kernel density function which may correspond to the process noise pdf, a Gaussian kernel or a rescaled version of the Epanechnikov kernel. This last solution is retained by the authors.

$$\begin{cases} K_h = \frac{1}{h^{n_x}} K(\frac{x}{h}) \\ h_{opt} = A.N^{-\frac{1}{n_x+4}} \\ A = (8C_{n_x}^{-1}(n_x+4)(2\sqrt{\pi})^{n_x})^{\frac{1}{n_x+4}} \\ K(x) = \begin{cases} \frac{n_x+2}{2C_{n_x}}(1-||x||^2) & \text{if } ||x|| < 1 \\ 0 & \text{otherwise} \end{cases} \end{cases} \qquad (44)$$

with C_{n_x} the volume of the unit sphere in $\mathbb{R}^{n_x}$.
The procedure for the creation of a new population of equally weighted particles for $t+k$ is given by the following pseudo code.

```
1. Apply a modified inverse transform resampling procedure: for $i = 1, ..., N$, $w_{t+k}^{(i)} = N^{-1}$
2. Calculate $\hat{S}_{t+k}$, the empirical covariance matrix of $\left\{E\left[x_{t+k}^{(i)}|\hat{x}_{t+k-1}^{(i)}\right], w_{t+k}^{(i)}\right\}_{i=1}^{N}$
3. Compute $\hat{D}_{t+k}$ such that $\hat{D}_{t+k}\hat{D}_{t+k}^{T} = \hat{S}_{t+k}$
4. For $i = 1, ..., N$ draw $\epsilon^i \sim K$, the Epanechnikov kernel and assign $\hat{x}_{t+k}^{(i)*} = \hat{x}_{t+k}^{(i)} + h_{t+k}^{opt}\hat{D}_{t+k}\epsilon^i$ where $h_{t+k}^{opt}$ is computed as written above.
```

This approach is also applied in [49, 63].

6.1.2. Creation of artificial measurements

Method 1: PF-LSSVR framework [12]. Classical approaches assume that the time varying parameters $\theta_{i,k}$ remain equal to their estimation at the initial prediction step k and invariant in the whole prediction phase. This fixed value is not consistent with the principle of time variation. Consequently, a new approach is proposed.
The particle filter used in this framework is dual particle filter (DPF). DPF uses a separate state-space representation for the state and the parameters offering better state and parameter estimations. The prognostics framework is divided into four phases: (1) data collection, (2) observation series prediction, (3) successive dual estimation and (4) RUL calculation.

In phase (2), after obtaining the latest observation y_k, LSSVR learning samples are chosen according to a sliding time window, they are trained to obtain a prediction model as well as a recursively calculation of the future observation data (each observation variable parameter data treated as an independent series). Then during phase (3), a DPF is used based on the predicted observations. The authors mention that although a DPF is used, there is no constraint on the choice of particle filter type in this framework.

This framework also considers the uncertainty propagation. To obtain an unbiased result, a time-varying observation noise ν_k is embedded in the prediction phase. A growth model of the variance is used:

$$var(\nu_{k+i_s}) = (1 + \frac{q}{p} i_s) var(\nu_k) \tag{45}$$

where i_s is the index of the prediction step, q an adjustment coefficient which indicates the multiple difference of the noise variance at k and $k+p$. The value of q depends on p and the quality of the predicted observation.

Method 2: PF-neural networks [44]. To solve the problem of lack of measurement, a data-driven method based on neural networks (NN) is used to update the model during the prediction. Three data-driven methods are proposed: (1) feedforward NN, (2) neuro-fuzzy and (3) recurrent neural fuzzy with 5 inputs for each one. The global scheme is describe in Figure 10.

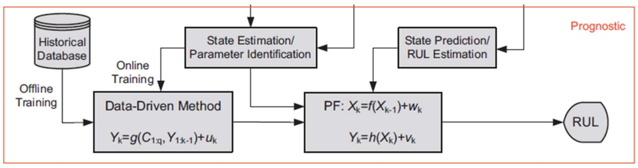


Figure 10: PF-NN framework [44]

6.2. Final distribution

The final pdf is composed only with the particles under the failure threshold. The weight can be re-normalized to take this into account [5].

6.3. RUL estimate

A classical approach to build the RUL is to take the point where each particle crosses the threshold. It is formulated in [75], the RUL is the time remaining before the crossing of a threshold λ and its pdf is defined as

$$\hat{p}(RUL \leq l | y_{0:k}) = \hat{p}(x_{k+l} \geq \lambda | y_{0:k}) \tag{46}$$

The authors assert that the use of alternative biased estimators for prognostics is more efficient when a large grain of uncertainty is involved which possibly introduces outliers. The proposed biased estimator is:

$$\tilde{p}(RUL \leq l | y_{0:k+l}) = \frac{\sum_{r=k}^{k+l-1} \hat{\nu}_r(y_{r+1:k+l}) \hat{p}(RUL \leq l | y_{0:r})}{\sum_{r=k}^{k+l-1} \hat{\nu}_r(y_{r+1:k+l})} \tag{47}$$

where $\hat{\nu}_r(y_{r+1:k+l})$ is called the credibility weights, properly defined to measure the credibility of $\hat{p}(RUL \leq l | y_{0:r})$. Three proposal are given for these weights (but not detailed here): interval weights, likelihood weights, RMSE weights.

A simpler solution is to use the particle distribution to calculate the RUL pdf by fitting a mixture of Gaussian in a least square sense [53].

6.4. Partial synthesis

Two families of procedures are proposed to perform prognostics. It is hard to prefer one rather than another as they have never been compared. However, an orientation toward the projection methods seems more appropriate. Indeed creating new measurements is dangerous: one makes predictions based on a prediction of data. Moreover, it raises a question. What is the interest to use particle filter-based prognostics if a prediction is already made with another tool? First, if the predicted measurements are false, the whole procedure is invalid. Then, it is hard to give significance to the final uncertainty as different quantities of noise terms are involved.

Based on these remarks, our preference for prognostics goes to the following classical procedure:

1. estimate the SoH thanks to the particle filter;
2. take the final distribution and reset the weight of each particle to $1/N$ if necessary;
3. propagate the particles with the state model until the failure threshold is reached;
4. interpret the final result thanks to the spatial location of the particles.

7. Validation

7.1. Evaluation metrics

To evaluate their results, prognostics practitioners are using different kinds of metrics. Some of them can be used in every prognostics applications while other tend to be more specific to particle filter utilization. They are summarized in Table 8. For their description, please refer to the references provided.

According to that table, 22 metrics are used in the different references. We can wonder if such a number is necessary, it appears to only make the comparison of the results harder. But such interrogation is beyond the scope of this study. What is important to notice is that only the ESS is specific to particle filters.

The question about how to evaluate the performance of the particle-filter based algorithm still an open issue at that stage.

7.2. Dealing with uncertainty in particle filters

7.2.1. Current ideas

In its recent paper dealing with uncertainty in prognostics [58], Sankararaman discuss about uncertainty treatment in Bayesian approaches. Its opinion on the subject

Table 8: Metrics used to evaluate the prognostics results

Type	References
$\alpha - \lambda$ metric	[15, 44, 63]
Accuracy	[46, 47, 48, 49]
Accuracy index	[28, 63]
Accuracy penalized	[46]
Average lower and upper bound error	[72]
Critical-α index	[46]
Dynamic standard deviation (DSTD)	[46]
ESS and Estimation error	[25]
Precision	[47, 48, 49]
Precision index	[28]
Prediction accuracy	[72]
Prognostics horizon	[12]
Relative accuracy	[12, 15]
Risk index	[28]
RMSE	[11, 14, 69]
RUL accuracy-precision index	[48, 49]
RUL error	[25]
RUL-OPI and RUL-OSI	[48, 49, 63]
Steadiness index	[28]

is summarized here. Some sentences are direct quotes from the reference.

According to Sankararaman, papers that claim to account for uncertainty in prognostics using Bayesian techniques are not technically accurate. He states that filtering can be used only to estimate the health state of the system based on data and cannot be used for future prediction. The use of a limited number of samples causes uncertainty regarding the estimated pdf.

Also in [58], it can be learned the Bayesian methodologies are based on subjective probabilities. These probabilities are considered as degrees of belief and quantify the existing knowledge. It is asserted that, with time, subjective and Bayesian have become synonymous. In subjective approaches, the subjective degree of the analyst's belief regarding a quantity may be represented by a pdf, even for a deterministic one. Consequently, pdf are assigned to parameters needing estimation. Such interpretation facilitates the propagation of the uncertainty. Note that, in some cases, subjective probabilities are applied in situations already involving physical ones.

Some keys to the theory of subjective probability rely in the concept of likelihood and its use in Bayes' theorem. A variety of problems are solved using Bayesian methods. Filtering approaches, as Kalman filters or particle filters, are known as Bayesian tracking methods. First as they use the Bayes' theorem, but also since the provided uncertainty has to be interpreted subjectively. It means that there is no uncertainty regarding the true states at each time steps but these states are not precisely known. The estimated pdf reflect the subjective knowledge regarding the state and the variables.

7.2.2. Partial synthesis

The claim that filtering cannot be used for future prediction meets our previous conclusion. What has to be further discussed is what is the uncertainty involved in a particle-filtering approach. Let's try to reason step by step based on the above citations and the knowledge of the filter functioning. For the record, the three stages of the particle filter functioning are: prediction, update and resampling. The case where the measurements are predicted to complete the available data is not considered.

Initially, we have data. These data contain uncertainty linked to the sensors. For the filter functioning, we need a model with parameters to identify to reflect the state according to the data. As the state is not known precisely, it is represented by a pdf $p(x_t|x_{t-1})$ at the end of the prediction. State estimates and data intervene both at the same time when estimating $p(y_t|x_t)$ during the update stage just before resampling. At that time, the pdf reflects both the measurement uncertainty and the ignorance of the state. Then comes the resampling which contains stochastic processes. So the pdf obtained after the resampling contains:

1. the measurement uncertainty;
2. the ignorance on the precise system state;
3. an uncertainty inherent to the stochastic processes.

As the filtering evolves with the coming of new data, the ignorance on the system state should decrease. While the uncertainty due to stochastic processes should increase with the number of resampling times. However, verification is not possible. What is important to note is that the particle filter output is drawn by 3 factors, two of them being uncertainties.

Then the prognostics start based on this output. The particles are now projected in the future. Based on Section 6, two cases should be considered. In the first case, a new tool is introduced to project the particles. We can assume that the tools used to extrapolate the future trajectories introduce a new kind of uncertainty: the uncertainty due to the prognostics tool. In the second case, the particles are just propagated thanks to the state model. Is there a new uncertainty introduced this time? In principle, no. There might only be an increase of the existing one.

8. Synthesis and conclusion

This part II of the study was dedicated to the review of 46 references to better understand the use of particle filters in the context of prognostics. It allowed showing the advantages and weaknesses of the tool and also distinguishing different issues linked to its use. Some points such as the need of pre-processing the data or in which proportion the initialization impact the results have not been discussed due to the lack of information.

Tables 9 to 11 summarizes the hypotheses of use and the issues coming from the particle filter as well as the proposals to tackle these problem from the analysis in the

Table 9: Summary of the analysis - Part 1

	Hypotheses	Issues	Proposals	Section
BEFORE USE	H1: 1-order Markov process	I1: the system does not respect H1	• P1: Adapting the tool such as in [11] to take into account m orders • P2: try to modify the equation to have a 1-order system at each time	5.1
	H2: The following elements are available • degradation model(s) • degradation measurements • measurement equation • failure threshold(s)	I2: Accurate degradation modeling is not available	P3: Consider changing the SoH estimation approach	3
		I3: Do the data require preprocessing?		
		I4: The measurement model is unknown	• P4: Implementing the proposal from [4] with the Gaussian assumption • P5: Defining the likelihood independently from the measurement model	

context of prognostics.

Particle filters are more and more used in the context of prognostics as they have a lot of advantages when dealing with non-linear non-stationary models and with noises non-necessary Gaussian. However, it is important to notice that particle filters have no predictive capabilities and can be used only for learning the current state of health of the system. In this case, a lot of issues regarding the choice of the filter type and its adaptation to the needs of a specific industrial system appear. Prognostics is then performed based on the filter's output by propagating it until a desired horizon.

It is important to mention that until now, particle filter-based prognostics' works mainly used existing filters. It already gives good results. Nevertheless, improvements could be made based on the knowledge of the system, its environment and maintenance constrains. Indeed, a deeper contextualization may lead to transformations in the algorithm and benefits to the state of health estimation.

The critical point in the literature about particle filter based-prognostics is the current lack of validation. Comparative studies between the filters used in the context of prognostics are very scarce and lack of exhaustiveness. This is also true for the prediction procedures. Different procedures to propagate the particles from the final filter distribution exist but have not been compared yet. Next steps of this work would be to launch different comparative studies to build a grid useful for industrial applications. First, by testing different filters on a same industrial system to see if one outperforms the others in the state of health estimation. Then by applying them on different systems to confirm the results. And finally, by trying the different propagation procedures to see their impact on the RUL estimates.

Validation is a crucial step to further use particle filter based-approaches in actual industrial system. The question of how to integrate such prognostics tools in an embedded industrial system starts from the validation and a comprehensive lecture on how well implementing this tool.

9. Acknowledgement

The authors would like to thank the ANR project PROPICE (ANR-12-PRGE-0001) and the Labex ACTION project (contract ”ANR-11-LABX-01-01”) both funded by the French National Research Agency for their support.

References

[1] M. Abbas, A.A. Ferri, M.E. Orchard, and G.J. Vachtsevanos. An intelligent diagnostic/prognostic framework for automotive

Table 10: Summary of the analysis - Part 2

	Hypotheses	Issues	Proposals	Section
FILTER IMPLEMENTATION	H3: Expert knowledge and a prior degradation and failure analysis is available	I5: Choosing a filter	P6: Starting from the basic SIR and modifying it based on the system knowledge, for example with a constrained state space or a special resampling procedure	4
		I6: Parameters evolution • which method for parameters' update? • how to define the variance of the parameters distribution?	• P7: Using a Gaussian random walk • P8: Defining the variance thanks to historical data if available • P9: Incorporate an optimization method to the filter	3
		I7: How to manage high dimension vectors?	P10: Performing a sensitivity analysis to select only the most critical parameters to update	3.5
		I8: How to choose the importance density	P11: Sticking to the classical $p(x_k \mid x_{k-1})$	4
		I9: How to define the number of particles	P12: Implementing a similar to the adaptive particle filter	5.2
		I10: Results depend strongly on the initialization		
		I11: The filter is time consuming	• P13: Simplifying the procedure • P14: Trying to parallelize the tasks	2.2.3

electrical systems. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 352–357, June 2007.

[2] Dawn An, Joo-Ho Choi, and Nam Ho Kim. A tutorial for model-based prognostics algorithms based on matlab code. In *Proceedings of the annual conference of the prognostics and health management society*, 2012.

[3] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.

[4] P. Baraldi, M. Compare, S. Sauco, and E. Zio. Ensemble neural network based particle filtering for prognostics. *Mechanical Systems and Signal Processing*, 41(1-2):288–300, 2013.

[5] Piero Baraldi, Francesco Cadini, Francesca Mangili, and Enrico Zio. Model-based and data-driven prognostics under different available information. *Probabilistic Engineering Mechanics*, 32(0):66 – 79, 2013.

[6] Piero Baraldi, Francesca Mangili, and Enrico Zio. Investigation of uncertainty treatment capability of model-based and data-driven prognostic methods using simulated data. *Reliability Engineering and System Safety*, 112(0):94 – 108, 2013.

[7] D. Brown, G. Georgoulas, H. Bae, G. Vachtsevanos, R. Chen, Y.H. Ho, G. Tannenbaum, and J.B. Schroeder. Particle filter based anomaly detection for aircraft actuator systems. In *Aerospace conference, 2009 IEEE*, pages 1–13, 2009.

[8] S. Butler, F. O'Connor, D. Farren, and J.V. Ringwood. A feasibility study into prognostics for the main bearing of a wind turbine. In *Control Applications (CCA), 2012 IEEE International Conference on*, pages 1092–1097, 2012.

[9] S. Butler and J. Ringwood. Particle filters for remaining useful life estimation of abatement equipment used in semiconductor manufacturing. In *Control and Fault-Tolerant Systems (SysTol), 2010 Conference on*, pages 436–441, 2010.

[10] Paula Borges Olivio Cerdeira, Roberto Kawakami Harrop Galvão João Pedro, and Pinheiro Malère. Particle filter prognostic applied in landing gear retraction. In *Annual Conference of the Prognostics and Health Management Society 2013*, 2013.

[11] Chaochao Chen, Bin Zhang, G. Vachtsevanos, and M. Orchard. Machine condition prediction based on adaptive neuro-fuzzy and high-order particle filtering. *Industrial Electronics, IEEE Transactions on*, 58(9):4353–4364, Sept 2011.

[12] Xiongzi Chen, Jinsong Yu, Diyin Tang, and Yingxun Wang. A novel pf-lssvr-based framework for failure prognosis of nonlinear systems with time-varying parameters. *Chinese Journal of Aeronautics*, 25(5):715 – 724, 2012.

[13] Zhe Chen. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.

[14] Heng-Juan Cui, Qiang Miao, Wei Liang, Zhonglai Wang, and Michael Pecht. Application of unscented particle filter in remaining useful life prediction of lithium-ion batteries. In *Prognostics and System Health Management Conference 2012*, 2012.

[15] Matthew Daigle and Kai Goebel. Model-based prognostics with fixed-lag particle filters. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, 2009.

[16] M Dalal, J Ma, and D He. Lithium-ion battery life prognostic

Table 11: Summary of the analysis - Part 3

	Hypotheses	Issues	Proposals	Section
PREDICTION		I12: How to propagate the final distribution for prognostics? I13: There is no update of the model parameters during prediction	P15: use the proposed procedure based on simple model projection P16: Trying to extract scheme from parameters trajectories during the learning according to the mission profile	6
VALIDATION		I14: How to interpret the uncertainty in the predictions?	P17: Quantifying at the initial stage measurement uncertainty and the knowledge of the system to deduce the uncertainty of the prognostics process	7.2
		I15: How do we verify and validate the results?		

health management system using particle filtering framework. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 225(1):81–90, 2011.

[17] F. Daum. Nonlinear filters: beyond the kalman filter. *Aerospace and Electronic Systems Magazine, IEEE*, 20(8):57–69, Aug 2005.

[18] Hancheng Dong, Xiaoning Jin, Yangbing Lou, and Changhong Wang. Lithium-ion battery state of health monitoring and remaining useful life prediction based on support vector regression-particle filter. *Journal of Power Sources*, 271(0):114 – 123, 2014.

[19] Arnaud Doucet, Nando De Freitas, and Neil Gordon. *Sequential Monte Carlo methods in practice*. Springer, 2001.

[20] Jiajie Fan, Kam-Chuen Yung, and Michael Pecht. Predicting long-term lumen maintenance life of LED light sources using a particle filter-based prognostic approach. *Expert Systems with Applications*, 42(5):2411 – 2420, 2015.

[21] Paul Fearnhead and Benjamin M. Taylor. An adaptive sequential monte carlo sampler. *Bayesian Anal.*, 8(2):411–438, 06 2013.

[22] Dieter Fox. Adapting the sample size in particle filters through kld-sampling. *International Journal of Robotics Research*, 22:2003, 2003.

[23] K. Goebel, B. Saha, A. Saxena, J.R. Celaya, and J.P. Christophersen. Prognostics in battery health management. *Instrumentation Measurement Magazine, IEEE*, 11(4):33–40, 2008.

[24] Mircea Grigoriu. Linear and nonlinear systems with non-gaussian white noise input. *Probabilistic Engineering Mechanics*, 10(3):171 – 179, 1995.

[25] Limeng Guo, Yu Peng, Datong Liu, and Yue Luo. Comparison of resampling algorithms for particle filter based remaining useful life estimation. In *IEEE PHM conference 2014*, pages 1–8, June 2014.

[26] Katalin Gyrgy, Andrs Kelemen, and Lszl Dvid. Unscented kalman filters and particle filter methods for nonlinear state estimation. *Procedia Technology*, 12(0):65 – 74, 2014.

[27] Wei He, Nicholas Williard, Michael Osterman, and Michael Pecht. Prognostics of lithium-ion batteries based on dempster–shafer theory and the bayesian monte carlo method. *Journal of Power Sources*, 196(23):10314–10321, 2011.

[28] Yang Hu, Piero Baraldi, Francesco Di Maio, and Enrico Zio. A particle filtering and kernel smoothing-based approach for new design component prognostics. *Reliability Engineering and System Safety*, 134(0):19 – 31, 2015.

[29] Yang Hu and Pengcheng Luob. Performance data prognostics based on relevance vector machine and particle filter. *Chemical Engineering*, 33:349–354, 2013.

[30] CHEN Huanyuan, LIU Xindong, SHE Caiqi, and Yao Cheng. Power system dynamic state estimation based on a new particle filter. *Procedia Environmental Sciences*, 11:655–661, 2011.

[31] K. Javed, R. Gouriveau, N. Zerhouni, and P. Nectoux. Enabling health monitoring approach based on vibration data for accurate prognostics. *Industrial Electronics, IEEE Transactions on*, 62(1):647–656, 2015.

[32] Marine Jouin, Rafael Gouriveau, Daniel Hissel, Marie-Cécile Péra, and Noureddine Zerhouni. Prognostics of PEM fuel cell in a particle filtering framework. *International Journal of Hydrogen Energy*, 39(1):481 – 494, 2014.

[33] Marine Jouin, Rafael Gouriveau, Daniel Hissel, Marie-Cécile Péra, and Noureddine Zerhouni. Prognostics of proton exchange membrane fuel cell stack in a particle filtering framework including characterization disturbances and voltage recovery. In *Proceedings of the 2014 IEEE International Conference on Prognostics and Health Management*, 2014.

[34] Miroslav Krupa. Verification of particle filtering based framework implemented in matlab®. *SEEK Digital Library*, 2012.

[35] James Kuria Kimotho, Tobias Meyer, and Walter Sextro. PEM fuel cell prognostics using particle filter with model parameter adaptation. In *IEEE PHM conference 2014*, pages 1–6, June 2014.

[36] He Yong-feng Li, Ming and Fu-zhong Nian. An improved particle filter algorithm and its performance analysis. *AISS: Advances in Information Sciences and Service Sciences*, 4(23):547–555, 2012.

[37] Junfu Li, Chao Lyu, Lixin Wang, Liqiang Zhang, and Chenhui Li. Remaining capacity estimation of li-ion batteries based on temperature sample entropy and particle filter. *Journal of Power Sources*, 268:895–903, 2014.

[38] T. Li, M. Bolic, and Djuric P. Resampling methods for particle filtering. *IEEE Signal Processing Magazine (to appear)*.

[39] T. Li, J. M. Corchado, J. Bajo, S. Sun, and J. F. Paz. Do we always need a filter? *Cornell University Library*, 2015.

[40] Tiancheng Li, Shudong Sun, J.M. Corchado, and Ming Fei Siyau. A particle dyeing approach for track continuity for the smc-phd filter. In *Information Fusion (FUSION), 2014 17th International Conference on*, pages 1–8, July 2014.

[41] Tiancheng Li, Shudong Sun, and Tariq Pervez Sattar. Adapting sample size in particle filters through kld-resampling. *Electronics Letters*, 49(12):740–742, 2013.

[42] Tiancheng Li, Shudong Sun, Tariq Pervez Sattar, and Juan Manuel Corchado. Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches. *Expert Systems With Applications*, 41(8):3944–3954, 2014.

[43] N. Lingala, N. Perkowski, H.C. Yeong, N. Sri Namachchivaya, and Z. Rapti. Optimal nudging in particle filters. *Probabilistic Engineering Mechanics*, 37(0):160 – 169, 2014.

[44] J. Liu, W. Wang, F. Ma, Y.B. Yang, and C.S. Yang. A data-model-fusion prognostic framework for dynamic system state

forecasting. *Engineering Applications of Artificial Intelligence*, 25(4):814 – 823, 2012.

[45] Qiang Miao, Lei Xie, Hengjuan Cui, Wei Liang, and Michael Pecht. Remaining useful life prediction of lithium-ion battery with unscented particle filter technique. *Microelectronics Reliability*, 53(6):805 – 810, 2013.

[46] B.E. Olivares, M.A. Cerda Munoz, M.E. Orchard, and J.F. Silva. Particle-filtering-based prognosis framework for energy storage devices with a statistical characterization of state-of-health regeneration phenomena. *Instrumentation and Measurement, IEEE Transactions on*, 62(2):364–376, Feb 2013.

[47] M. Orchard, G. Kacprzynski, K. Goebel, B. Saha, and G. Vachtsevanos. Advances in uncertainty representation and management for particle filtering applied to prognostics. In *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, pages 1–6, Oct 2008.

[48] M Orchard, Liang Tang, Bhaskar Saha, Kai Goebel, and George Vachtsevanos. Risk-sensitive particle-filtering-based prognosis framework for estimation of remaining useful life in energy storage devices. *Studies in Informatics and Control*, 19(3):209–218, 2010.

[49] M Orchard, Liang Tang, and George Vachtsevanos. A combined anomaly detection and failure prognosis approach for estimation of remaining useful life in energy storage devices. In *Annual Conference of the Prognostics and Health Management Society*, pages 1–7, 2011.

[50] M Orchard, F Tobar, and G Vachtsevanos. Outer feedback correction loops in particle filtering-based prognostic algorithms: Statistical performance comparison. *Studies in Informatics and Control*, 18(4):295–304, 2009.

[51] Marcos E Orchard and George J Vachtsevanos. A particle-filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*, 31:221–246, 2009.

[52] Michael K. Pitt, Ralph dos Santos Silva, Paolo Giordani, and Robert Kohn. On some properties of markov chain monte carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171(2):134 – 151, 2012.

[53] B. Saha and K. Goebel. Uncertainty management for diagnostics and prognostics of batteries using bayesian techniques. In *Aerospace Conference, 2008 IEEE*, pages 1–8, March 2008.

[54] B. Saha, K. Goebel, S. Poll, and J. Christophersen. An integrated approach to battery health monitoring using bayesian regression and state estimation. In *Autotestcon, 2007 IEEE*, pages 646–653, Sept 2007.

[55] B. Saha, E. Koshimoto, C.C. Quach, E.F. Hogge, T.H. Strom, B.L. Hill, S.L. Vazquez, and K. Goebel. Battery health management system for electric uavs. In *Aerospace Conference, 2011 IEEE*, pages 1–9, March 2011.

[56] Bhaskar Saha and Kai Goebel. Modeling li-ion battery capacity depletion in a particle filtering framework. In *Proceedings of the annual conference of the prognostics and health management society*, pages 1–10, 2009.

[57] Bhaskar Saha and Kai Goebel. Model adaptation for prognostics in a particle filtering framework. *International Journal of Prognostics and Health Management Volume 2*, page 61, 2011.

[58] Shankar Sankararaman. Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction. *Mechanical Systems and Signal Processing*, 5253(0):228 – 247, 2015.

[59] J.Z. Sikorska, M. Hodkiewicz, and L. Ma. Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, 25(5):1803 – 1836, 2011.

[60] Alvaro Soto. Self adaptive particle filter. In *IJCAI*, pages 1398–1406, 2005.

[61] Pawe Mirosaw Stano, Zsfia Lendek, and Robert Babuka. Saturated particle filter: Almost sure convergence and improved resampling. *Automatica*, 49(1):147 – 159, 2013.

[62] Liang Tang, J. DeCastro, G. Kacprzynski, K. Goebel, and G. Vachtsevanos. Filtering and prediction techniques for model-based prognosis and uncertainty management. In *Prognostics and Health Management Conference, 2010. PHM '10.*, pages 1–10, Jan 2010.

[63] Liang Tang, M.E. Orchard, K. Goebel, and G. Vachtsevanos. Novel metrics and methodologies for the verification and validation of prognostic algorithms. In *Aerospace Conference, 2011 IEEE*, pages 1–8, March 2011.

[64] S. Thrun, Langford. J., and V. Verma. Risk sensitive particle filters. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.

[65] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. 2005.

[66] Ye Tian, Chen Lu, Zili Wang, and Laifa Tao. Artificial fish swarm algorithm-based particle filter for li-ion battery life prediction. *Mathematical Problems in Engineering*, 2014, 2014.

[67] Sridhar Ungarala. A direct sampling particle filter from approximate conditional density function supported on constrained state space. *Computers and Chemical Engineering*, 35(6):1110 – 1118, 2011.

[68] Rudolph Van Der Merwe, Arnaud Doucet, Nando De Freitas, and Eric Wan. The unscented particle filter. In *NIPS*, pages 584–590, 2000.

[69] Peng Wang and Robert X Gao. Particle filtering-based system degradation prediction applied to jet engines. In *Annual conference of the prognostics and health management society 2014*, 2014.

[70] Achmad Widodo and Wahyu Caesarendra. Summary of the recent developed techniques for machine heatlh prognostics. *ROTASI*, 16(1):21–27, 2014.

[71] Yinjiao Xing, Eden W.M. Ma, Kwok-Leung Tsui, and Michael Pecht. An ensemble model for predicting the remaining useful performance of lithium-ion batteries. *Microelectronics Reliability*, 53(6):811 – 820, 2013.

[72] Heng-Chao Yan, Chee Khiang Pang, and Jun-Hong Zhou. Precognitive maintenance and probabilistic assessment of tool wear using particle filters. In *Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE*, pages 7382–7387, 2013.

[73] Weibo Yang, Shenfang Yuan, Lei Qiu, Hua Zhang, and Biyun Ling. A particle filter and lamb wave based on-line prognosis method of crack propagation in aluminum plates. 2012.

[74] Bin Zhang, Chris Sconyers, Romano Patrick, and George Vachtsevanos. A multi-fault modeling approach for fault diagnosis and failure prognosis of engineering systems. In *Annual conference of the prognostics and health management society, San Diego, CA*, 2009.

[75] Enrico Zio and Giovanni Peloni. Particle filtering prognostic estimation of the remaining useful life of nonlinear components. *Reliability Engineering and System Safety*, 96(3):403 – 409, 2011.