Evaluation of the Training Process of three different Prognostic Approaches based on the Gaussian Process

Christian Preusche, Christoph Anger, and Uwe Klingauf

Institute of Flight Systems and Automatic Control, Technische Universität Darmstadt, Darmstadt, Hessen, 64287, Germany
preusche@fsr.tu-darmstadt.de
anger@fsr.tu-darmstadt.de
klingauf@fsr.tu-darmstadt.de

ABSTRACT

Data-driven prognostic approaches like Gaussian Process combined with Unscented Kalman Filter (GPUKF) are promising methods for predicting the Remaining Useful Lifetime (RUL) of a degrading component. Whereas the Gaussian Process (GP) is appropriate to derive a suitable degradation model by means of a set of training data, the Unscented Kalman Filter (UKF) employs this model to determine the prediction and its uncertainty.

Since a degradation process is highly stochastic, it is assumed that by applying more sets of training data the accuracy and precision of the GPUKF is increased. In order to examine the performance enhancement two different approaches are investigated in this paper: First, a single GP is trained with all available data sets. The second approach combines several GPs (each created with a data set of one degradation process) by extending the GPUKF with a Multiple Model Method. The development of a third prognostic approach aims at the investigation of the UKF as a suitable tool for the prognostic algorithm. Therefore, a third method applies a Particle Filter in combination with the GP.

For the evaluation of the aforementioned prognosis algorithms according to their precision and accuracy a set of prevalent performance metrics like the Prognostic Horizon and the Mean Average Percentage Error of a prediction is analyzed. The validity of the determined results is increased by considering the variance of certain metrics over several units under test. Moreover, particular focus is set on the examination of the performance change caused by the use of more training data sets. In order to quantify this process known metrics are extended. The evaluation is based on simulated data sets, which are generated by an exponential degradation model.

The analysis of the implemented algorithms indicates that the applied metrics are in a comparable range. However, the three approaches reveal a different behaviour concerning the convergence of the performance values according to the number of training data. In particular cases there is even a decline in accuracy and precision attend by a rising number of training data.

1. INTRODUCTION

In recent years the prognosis of the condition of component parts with a high relevance to safety has become a key technology in Condition-Based Maintenance (CBM), especially in application fields like aerospace or power generation. Although the use of a CBM system is aimed for cost reduction in the overall maintenance cycle, the initial implementation is cost-intensive, since a profound knowledge and observation of the examined element's degradation processes is essential.

Here, data-driven methods can be beneficial, as the origin and the mechanism of a failure is irrelevant for the generation of prognosis models. An additional advantage is the generic coding for possible applications of data-driven algorithms in comparison to model-based methods, which need a specific model for every degradation process. Beside other data-driven methods like the widely spreaded artificial Neuronal Networks or the Support Vector Machine for regression, the Gaussian Process (GP) became a state of the art regression estimator due to its simplicity and the ability to forecast model uncertainties.

The examinations in this paper focus upon the evaluation of three different prognosis methods, which all base on the GP for regression modelling. The first two algorithms use the Unscented Kalman Filter (UKF) for state estimation adapted from the results of (Anger, Schrader, & Klingauf, 2012), whereas the idea to combine a GP with a UKF was first introduced by (Ko, Klein, Fox, & Haehnel, 2007) for an observation model of a robotic blimp. In (Anger et al., 2012) it was proven that the combination of a GP with a UKF (GPUKF) is

generally capable of predicting even highly stochastic degradation processes using the example of a rolling-element bearing. Additionally, a second concept was introduced by applying several GPUKFs with different models which are connected by a superior algorithm, the Interacting Multiple Model (IMM). By means of the IMM in combination with a GP regression (GPMMM) the robustness of the predictions was significantly increased, since it is able to forecast different damage courses w.r.t. the training data sets. Thus, the central question of this enquiry is, if it is more beneficial to separate the available set of training data into different models or to set up one single model with all data points.

Drawbacks in the prediction uncertainty of the aforementioned GPUKF and GPMMM led to a third algorithm, a combination of a GP regression model with a Particle Filter (GPPF). It is shown that the prognosis of the RUL by means of the GPPF is a more straight forward approach according to the handling of variances. Again the idea of this combination has its seeds in the localization, since Ferris et al. used a similar algorithm for location estimation of people within buildings in (Ferris, Hähnel, & Fox, 2006).

For the evaluation of these different algorithms, performance metrics are necessary. In (Saxena et al., 2008) and (Saxena, Celaya, Saha, Saha, & Goebel, 2009) several metrics concerning the accuracy, precision and robustness of predictive algorithms are summarized. In section 4 well-known metrics like Mean Absolute Percentage Error (MAPE) or Prognostic Precision (PP) are extended by their values w.r.t. the applied number of training data sets. Since one major drawback of data-driven approaches is the need of training data, savings are possible, if these extended metrics do not indicate an increase in the prediction performance after a certain amount of training data.

This paper is divided into six sections: first, the three aforementioned algorithms GPUKF, GPMMM and GPPF are introduced in section 2. After that the model of the simulated degradation data is described, whereupon one major demand was simplicity. The evaluation and especially the applied performance metrics are described in section 4 and the results are summarized in section 5. We conclude in section 6.

2. PROGNOSIS ALGORITHMS

Three different prognostic concepts are compared in this paper, whereupon all base on the Gaussian Process regression modelling. One motivating question for the approaches is: "Is it more beneficial to train one GP with all available data sets or to establish many models by means of every single data set separately?". There are many benefits and drawbacks assumed, such as: If one GP is trained with many data sets, which result from a similar degradation process, the regression model inherits more information about the process and thus is less prone to process noise. On the other hand in case of highly stochastic degradation, the probability to find a match between trained models and tested degradation courses raises, when the regression models are established separately.

Thus, this section starts with the basics of GP regression modelling that was introduced in (Rasmussen, 2006). Afterwards the two algorithms GPUKF and GPMMM are shortly described. Since the UKF shows drawbacks concerning the prognostic uncertainty calculation, a third algorithm based on a Particle Filter is introduced.

2.1. Gaussian Process for regression modelling

Regression modelling tools like the GP enable the opportunity to reproduce processes without the application of any parametric model. Since the GP defines a Gaussian distribution over a function, see (Rasmussen, 2006), the main advantage of regression modelling with a GP is furthermore the potential to identify the model's uncertainty according to the variance of the distribution.
Thus, the aim of the GP regression modelling is to establish a function $f(\mathbf{X})$ so that a noisy process

$$\mathbf{y} = f(\mathbf{X}) + \epsilon, \tag{1}$$

can be described w.r.t a given training data set $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ...(\mathbf{x}_n, y_n)\}$, where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]^T$ is an $n \times m$ input matrix with m the number of inputs and n the length of the single input vector $\mathbf{x}_i$. $\mathbf{y}$ is an n-dimensional vector of scalar outputs and ϵ represents a noise term, which is drawn from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$.

A Gaussian distribution can be described by its mean μ and covariance Σ. Thus, the GP defines a prior which is a zero-mean Gaussian distribution over the given outputs $\mathbf{y}$ of the training data D, as follows

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{0}; \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}). \tag{2}$$

Here, $\sigma_n^2 \mathbf{I}$ is a Gaussian noise caused by ϵ. The entries of the kernel matrix $\mathbf{K}$ indicate the deviation of the inputs among each other and are defined by the applied covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$. Although the squared exponential is a standard kernel function, in this paper it is extended by a linear and constant term as follows

$$\begin{aligned} k(\mathbf{x}_i, \mathbf{x}_j) &= \sigma_f^2 \exp(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)\mathbf{W}(\mathbf{x}_i - \mathbf{x}_j)^T) \\ &\quad + \sigma_n^2 \delta_{ij} + \sigma_l \mathbf{x}_i \cdot \mathbf{x}_j + \sigma_c, \end{aligned} \tag{3}$$

where σ_f^2 is the signal variance and $\mathbf{W}$ is a diagonal matrix that contains the distance measure of every input. By means of the other parameters σ_l and σ_c, the linear and constant deviation can be tuned separately.

The mean GP_μ and the covariance GP_Σ is then expressed for a given test input $\mathbf{x}_*$ and test output y_* w.r.t. the training data

D by the following equations

$$GP_\mu(\mathbf{x}_*, D) = \mathbf{k}_*^T[\mathbf{K} + \sigma_n^2\mathbf{I}]^{-1}\mathbf{y} \tag{4}$$

for the mean and

$$GP_\Sigma(\mathbf{x}_*, D) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T[\mathbf{K} + \sigma_n^2\mathbf{I}]^{-1}\mathbf{k}_* \tag{5}$$

for the covariance, respectively. For reasons of clarity the abbreviations $\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{K}$ and $\mathbf{k}_*$ the covariance function between the test input $\mathbf{x}_*$ and the training input vector $\mathbf{X}$ are used. Obviously, the mean prediction in equation 4 is a linear combination of the training output $\mathbf{y}$ and the correlation between test and training input. The covariance is the difference of the covariance function w.r.t. the test inputs and the information from the observation $\mathbf{k}(\mathbf{x}_*, \mathbf{x}_*)$.

All in all, the regression modelling with the applied GP requires the optimization of five so-called hyperparameters $\theta = [\mathbf{W}\ \sigma_f\ \sigma_n\ \sigma_l\ \sigma_c]$ for the kernel function and the process noise. This can be done by standard optimization algorithms as conjugate gradient descent.

For the purpose of this paper, the degradation of the Unit Under Test (UUT) is considered as a 1-dimensional state specified by x. Using equation 1 a stochastic dynamic degradation process can be written as

$$x_{k+1} = x_k + \Delta x_k + \epsilon_k. \tag{6}$$

The GP regression modelling is then applied on the state transition Δx_k so that the input $\mathbf{X}$ is the current degradation state x_k and the output $\mathbf{y}$ is the state transition. With the training data set $D = \{\mathbf{x}, \Delta\mathbf{x}\}$ the next degradation state is written as

$$x_k = x_{k-1} + GP_\mu(x_{k-1}, D) \tag{7}$$

and the covariance $GP_\Sigma(x_{k-1}, D)$, both fully describe the Gaussian distribution of the GP. One example of the degradation modelling is given in figure 1.

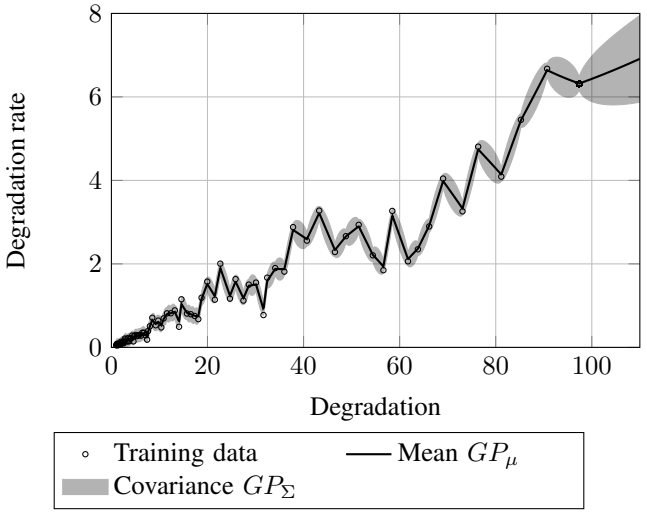

Figure 1. Example of dynamic degradation modelling using Gaussian Process

2.2. Combining GP and an Unscented Kalman Filter

The aforementioned dynamic model of a stochastic degradation process is the basis for the first prognostic algorithm, where one GP is trained with all available training data. Since it is expected that the different degradation courses which have to be forecast are quite similar, the application of all training data in one GP is assumed to be beneficial, as the GP contains more information about the degradation process. Additionally, for uncertainty estimations of the new degradation state w.r.t. measurements and the applied model, an UKF is necessary.

Similar to equation 1, a nonlinear dynamic system in the k^{th} time step can be described as

$$x_k = g(x_{k-1}) + \epsilon_k \tag{8}$$

with the state transition function g, the 1-dimensional degradation x and an additive Gaussian noise term ϵ drawn from a zero-mean Gaussian distribution $\epsilon \sim \mathcal{N}(0, Q_k)$ with the process noise Q_k as covariance.

The basis of the UKF is the scaled unscented transformation introduced in (Julier, 2002). Instead of a linearization process of the transition function g (as in case of the Extended Kalman Filter), sigma points $\chi^{[i]}$ are defined w.r.t. the covariance Σ and the value of degradation x of the previous time step

$$\begin{aligned}
\chi_k^{[0]} &= x_{k-1} \\
\chi_k^{[i]} &= x_{k-1} + (\sqrt{(n+\lambda)\Sigma_{k-1}})_i \quad i = 1, ..., n \\
\chi_k^{[i]} &= x_{k-1} - (\sqrt{(n+\lambda)\Sigma_{k-1}})_{i-n}\ i = n+1, ..., 2n,
\end{aligned} \tag{9}$$

where λ is a scaling parameter to spread the single sigma points. According to the standard UKF, these sigma points are transformed by the transition function g. Since the applied algorithm plans to use the mean function of the GP (see equation 6), the transformed sigma points are as follows

$$\bar{\chi}_k^{[i]} = \chi_k^{[i]} + GP_\mu(\chi_k^{[i]}, D). \tag{10}$$

The mean and covariance of the next time step are then generated by

$$x_k = \sum_{i=0}^{2n} w_m^{[i]} \bar{\chi}_k^{[i]} \tag{11}$$

$$\Sigma_k = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\chi}_k^{[i]} - x_k)(\bar{\chi}_k^{[i]} - x_k)^T + GP_\Sigma(x_k, D) \tag{12}$$

with the weights for mean value w_m and covariance w_c, respectively. Instead of the process noise Q_k, the covariance function of the GP is used. Since this algorithm is only employed for prognosis, a correction step as in the standard UKF-algorithm is omitted.

The entire prediction process is sketched in figure 2.

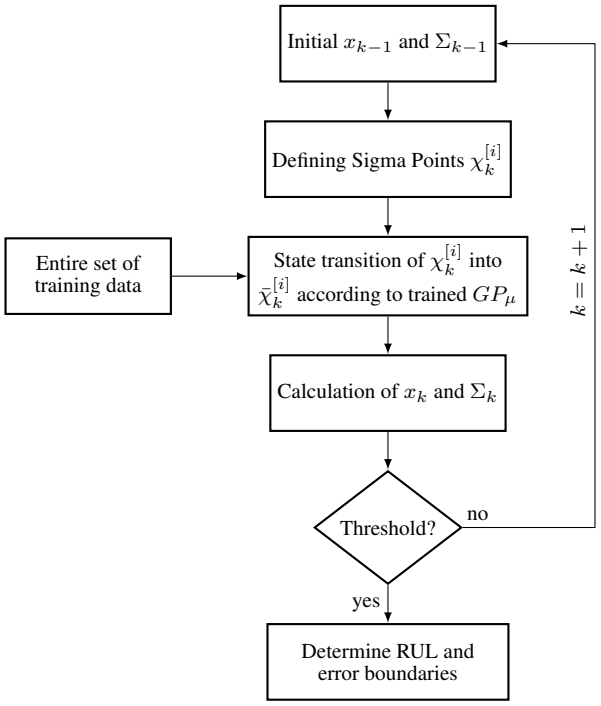

Figure 2. Basic schematic of the GPUKF prognostic approach

2.3. Extension via a Multiple Model Approach

Instead of training one GP with all available data points the second algorithm separates the data sets and creates several trained GPs for degradation prediction. Again, the UKF is used for uncertainty estimation. The superior algorithm that connects the different prognostic models is called Interacting Multiple Model (IMM), which was introduced in (Li & Jilkov, 2003). Since a degradation process is highly stochastic, the prognostic accuracy is expected to increase by the application of various models that could be similar to the tested damage case.

Considering equation 8, the extension to the multiple model approach follows as

$$x_k = g(x_{k-1}, m^i) + \epsilon_k, \tag{13}$$

where additionally m^i is the i prognostic model of M available models. The first steps of the IMM algorithm consist of a reinitializing step with the calculation of a mode probability ξ_{k-1}^i of every i^{th} model

$$\begin{aligned} \xi_{k-1}^i &= p(m_k^i|y_{1:k}) \quad \text{for } i = 1, ..., M \\ &= \sum_{j=1}^{M} h_{ij}\xi_{k-1}^j \end{aligned} \tag{14}$$

with the entries $h_{ij} = p\left\{m_k = m^j | m_{k-1} = m^i\right\}$ of the transition matrix **H**. The application of the transition matrix **H** prevents the prognostic approach of insisting on one model, as it offers the possibility of a change in the mode probability from model i to j during every time step. Therefore, the transition matrix **H** describes a Markov chain, whereupon **H** is assumed to be time invariant.

In comparison to other hybrid estimators as the Autonomous Multiple Model the IMM uses the results of every integrated filter by the application of a weighting factor according to

$$\begin{aligned} \xi_{k-1}^{j|i} &= p(m_{k-1}^i|y_{1:k-1}, m_k^i) \\ &= \frac{h_{j|i}\xi_{k-1}^j}{\xi_{k-1}^i}. \end{aligned} \tag{15}$$

Herewith an individual reinitializing value for the state for every filter

$$\begin{aligned} \bar{x}_{k-1}^i &= E[x_{k-1}|y_{1:k-1}, m_k^i] \\ &= \sum_{j=1}^{M} \hat{x}_{k-1}^j \xi_{k-1}^{j|i} \end{aligned} \tag{16}$$

and similarly a covariance $\bar{\Sigma}_{k-1}^i$ is computed. W.r.t. these initial values the several models m^i predict the degradation state of the next time step, independently. In the end the results of the i filters are fused w.r.t. the model probability ξ_k^i

$$\hat{x}_k = \sum_{i=1}^{M} \hat{x}_k^i \xi_k^i \tag{17}$$

$$\Sigma_k = \sum_{i=1}^{M} [\Sigma_k^i + (\hat{x}_k - \hat{x}_k^i)(\hat{x}_k - \hat{x}_k^i)^T]\xi_k^i. \tag{18}$$

The entire algorithm is sketched in figure 3. In comparison to the first algorithm, the GPMMM requires a previous state estimation to identify the model probability ξ_k^i that remains constant during each prediction. According to the likelihood L_k^i, which depends on the residuum $e_k^i = z_k^i - \hat{z}_k^i$ of the measured and estimated state, respectively, the probability that i is the correct model is calculated as

$$\xi_k^i = \frac{\xi_{k-1}^i L_k^i}{\sum_{j=1}^{M} \xi_{k-1}^j L_k^j} \tag{19}$$

at the beginning of every prediction.

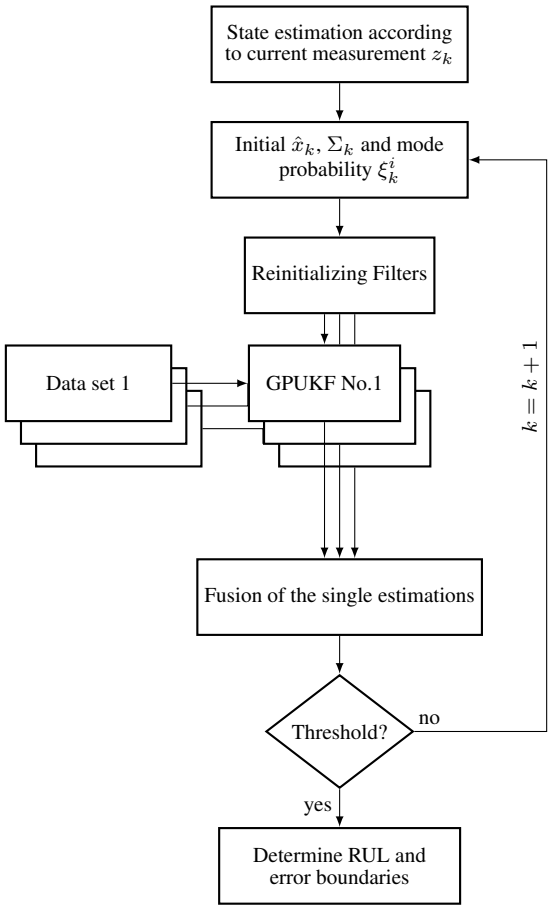

Figure 3. Basic schematic of the GPMMM prognostic approach

One major problem that comes along with the application of the UKF in combination with the model of equation 10 is an observable drift of the sigma points, which can also be identified in several plots concerning the position estimation in (Ko et al., 2007). Consequently, considering equation 12 the covariance of the respective filters rises and the drift is intensified. Since this process is repeated in every time step, the covariance diverges especially in case of early predictions that results in a possible negative degradation, i.e. bettering, of the examined element.

To counteract the drift of the sigma points, the weighting factor $w_c^{[i]}$ was reduced so that the prediction uncertainty remains within an acceptable limit.

2.4. Combining GP and a Particle Filter

The particle filter is established as a flexible mathematical method to represent and manage uncertainties of a long-term prediction, see (Orchard & Vachtsevanos, 2009) or (Saha, Goebel, & Christophersen, 2009). The problems of the aforementioned prognosis approaches in handling the uncertainties of the prediction motivate to combine the GP with a Particle Filter (GPPF). Likewise, the GPPF approach includes various degradation behaviors by means of an arbitrary number of dynamic degradation models, each represented by an individual GP.

In this section only a brief introduction of the operating principle of the particle filter is given. The reader is encouraged to follow (Arulampalam, Maskell, Gordon, & Clapp, 2002) for more detailed information. Differences to the basic filter and enhancements due to the supporting of multiple prognosis models are highlighted.

The essential idea behind the particle filter is to estimate the Probability Density Function (PDF) of the UUT's degradation by means of a weighted set of particles. With an appropriate amount of particles the current and future PDF of the degradation can be estimated. In the suggested approach an individual particle n is characterized by its level of degradation $x_{n,k}$ at time k and a parameter j_n, which is independent of the time and assigns a particle to a i^{th} prognostic model.

Figure 4 illustrates the basic schematic of the implemented prognostic approach. In the initialization step, the degradation $x_{n,0}$ and the parameter j_n of each particle is defined. Given a total amount of N particles and M trained prognostic models each model is equally represented by N/M particles.

During the prediction step each particle is transferred from the state $x_{n,k-1}$ to the state $x_{n,k}$ using the training data set D of the assigned prognostic model j_n. Given the last degradation of a particle $x_{n,k-1}$, the mean function GP_μ and covariance function GP_Σ the evolution of each particle can be written as:

$$\begin{aligned} x_{n,k} &= x_{n,k-1} + \mathcal{N}(\mu, \Sigma) \\ \mu &= GP_\mu(x_{n,k-1}, D) \\ \Sigma &= GP_\Sigma(x_{n,k-1}, D). \end{aligned} \tag{20}$$

When more information about the degradation of the UUT accumulates over time, the measured degradation z_k is applied to update the weight of each particle $w_{n,k}$ and to determine the model probability ξ_k^i. The weight of a particle is updated by using the normal probability density function and the previous weight:

$$w_{n,k} = w_{n,k-1} \cdot \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{\left(-\frac{(x_{n,k}-z_k)^2}{2\sigma^2}\right)}, \tag{21}$$

where σ is a known noise distribution of the measured degradation. After updating all particles the weights are normalized ($\sum_{n=1}^{N} w_{n,k} = 1$). In order to update the model probabilities ξ_k^i, the weights of particles assigned to the same model

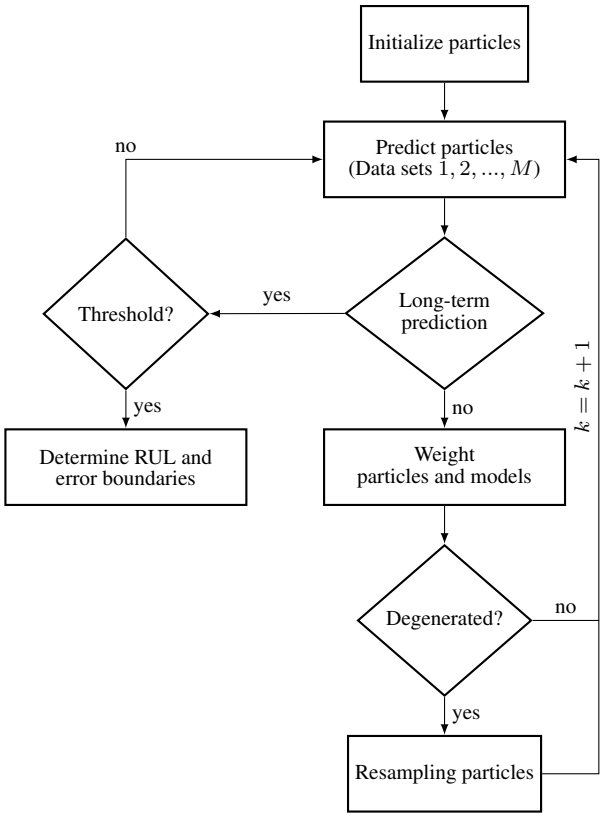

Figure 4. Basic schematic of the GPPF prognostic approach

i are summarized and normalized

$$\begin{aligned} \hat{\xi}_k^i &= \xi_{k-1}^i \cdot \sum_{n=1}^{N} a(n) \\ a(n) &= \begin{cases} w_{n,k} & j_n = i \\ 0 & j_n \neq i \end{cases} \\ \xi_k^i &= \frac{\hat{\xi}_k^i}{\sum_{m=1}^{M} \hat{\xi}_k^m}. \end{aligned} \tag{22}$$

One major problem by using a particle filter is that after several iterations all but some particle weights are close to zero. To avoid this situation, the resampling step is executed. A helpful indicator to test whether a resampling of the particles is needed or not is the Effective Sample Size (ESS). Regarding (Arulampalam et al., 2002), ESS can be calculated by

$$\mathrm{ESS} = \frac{1}{\sum_{n=1}^{N} w_{n,k}^2}. \tag{23}$$

If ESS passes a defined threshold, particles with a low probability are replaced by particles with a high probability. Thereby, it is assured that each model i is still represented by the same amount of particle as before. Consequently, particles of a prognostic model, which inappropriately describes the current degradation behavior, profit from a well matching model. As a result the resampling does not only prevent the degeneration of particles but also the degradation of models.

In cases of long-term prediction is required, the prediction step is executed iteratively until all particles pass a predefined failure threshold of the system. Given the prediction equation 20 and a prognosis model (see figure 1) it is obvious that a particle passes the threshold at some time. In other words, the predicted PDF of the degradation will always indicate a deterioration of the UUT. The problem of a possible negative degradation described in section 2.3 is prevented by using the GPPF approach.

Considering the estimated EoL_n of each particle the expected RUL and their uncertainty limits are determined. The probability of a failure at time k is determined by the probability of the particles, which passed the threshold at this time, and the probability of the assigned prognosis models. Figure 5 illustrates an obtained distribution of the EoL_n using the particle filter approach trained with three training data sets. The estimated and real RUL as well as the upper and lower predicted limits are marked. Since the obtained distribution cannot be classified as normal distribution, an investigation by means of the expected value and the standard deviation is not appropriate. Instead it is preferred to rely on the median and the percentiles to specify the RUL and uncertainties.

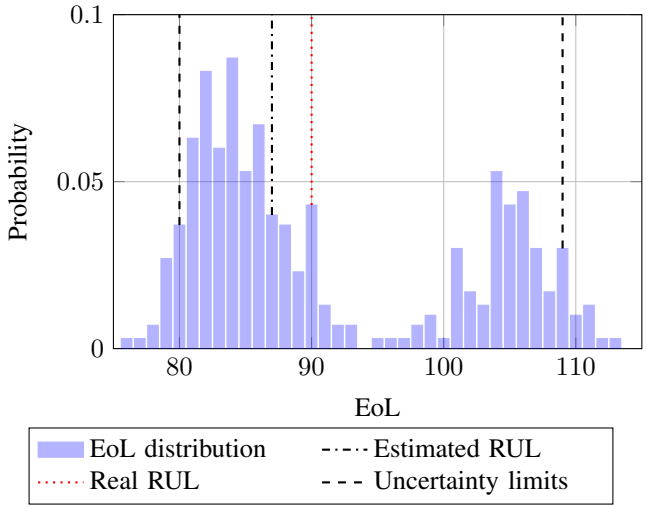

Figure 5. Distribution of the predicted EoL and determination of the estimated RUL including upper and lower limits (GPPF is trained with three data sets)

3. SIMULATED DATA

For a performance investigation of the presented prognostic algorithms, a set of training and test data is needed. This section introduces the developed mathematical model which is applied to generate a data pool containing various degradation courses, each simulating a run-to-failure behavior of an individual UUT.

The object of the derived model is to describe an exponential failure process of a UUT including a stochastic part to assure an analogy to reality. Moreover, an important requirement we set up is to keep the mathematical model as simple as possible. This should encourage the reader to rebuild the model and compare the presented results in section 5 with other prognosis algorithms.

The equation of the mathematical model can be written as following:

$$\begin{aligned} z_k &= z_{k-1} + \frac{\ln(100)}{100} \cdot e^{(a_{k-1} \cdot k)} + \mathcal{N}(0, b_{k-1}) \\ a_k &= a_{k-1} + \mathcal{N}(0, \upsilon_a) \\ b_k &= \frac{z_{k-1}}{b_S}. \end{aligned} \tag{24}$$

The course of the degradation z_k is subject to several effects. First, the state a influences the exponential course by varying each step according to a noise term, defined by a normal Gaussian distribution with zero mean and variance υ_a. Furthermore, the second noise term $\mathcal{N}(0, b_{k-1})$ simulates an instability of the UUT reasoned by the advanced fault, implemented by the dependency of the variance b on the degradation.

For the generation of a data pool, the failure threshold is set to a degradation level of 100, the model was designed to reach the limit in approximately 100 time steps. The noise values are defined as $\upsilon_a = 8 \cdot 10^{-4}$ and $b_S = 70$, the initial level of the parameter are $y_0 = 1$, $a_0 = 5 \cdot 10^{-2}$ and $b_0 = 1 \cdot 10^{-3}$. Figure 6 shows an example of four obtained UUTs.

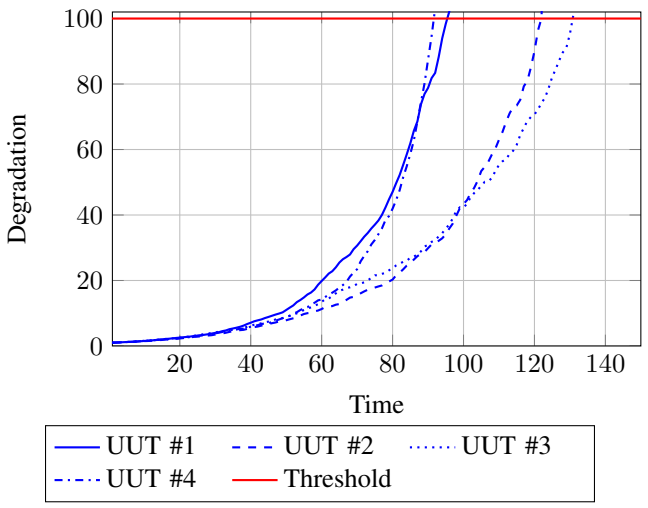

Figure 6. Samples of simulated degradation courses generated by the developed mathematical model

4. EVALUATION CONCEPT

During recent years much effort was put into the definition of metrics to assess the performance of prognostic methods and to make them comparable with each other. Since the performance of a data-driven prognostic approach depends on the number of available historical run-to-failure data (Anger et al., 2012), the evaluation should not only consider the individual performance metrics but the change of those metrics when given more training data. It is assumed that not always better results are achieved, when applying more historical data. In some cases, a degradation of the metrics is expected, since inappropriate training data may irritate a prognosis algorithm. However, prognostic methods are rarely investigated regarding their ability in handling various training data. The purpose of the following evaluation concept is to analyse the change of performance metrics according to the number of training data and to figure out an appropriate way to quantify this process.

For the evaluation we included four metrics, namely MAPE, MAD, PH and PP, to cover accuracy as well as precision properties of the three prognosis methods. A brief description of the metrics is given in section 4.1. The procedure of the evaluation is explained in 4.2, whereas a way to quantify the change of the performance is described in section 4.3.

4.1. Performance Metrics

The applied metrics are based on the suggestions given by (Saxena et al., 2008) or (Saxena et al., 2009). Some notations of the metrics domain are given in the following glossary:

P	Time of the first prediction
EoL	End of Life
i	Prediction index $i = 1, 2, ..., I$
I	Total number of predictions
l	UUT index $l = 1, 2, ..., L$
L	Total number of UUTs
λ	Normed time of the entire range $(EoL - P)$
$r^l(i)$	Estimated RUL of prediction i for the l^{th} UUT
$r^l_*(i)$	Real RUL of prediction i for the l^{th} UUT
$\Delta^l(i)$	Error between predicted RUL and true RUL $\Delta^l(i) = r^l_*(i) - r^l(i)$

4.1.1. Mean Average Percentage Error

The Mean Absolute Percentage Error (MAPE) of a prediction testing the l^{th} UUT is specified by

$$MAPE^l = \frac{1}{I} \sum_{i=1}^{I} \left| \frac{100 \cdot \Delta^l(i)}{r^l_*(i)} \right|. \tag{25}$$

The value of MAPE determines the predicted error w.r.t the real RUL.

4.1.2. Mean Absolute Deviation

The Mean Absolute Deviation (MAD) describes the spread of the prediction error and quantifies the precision of a method. The metric can be written as

$$MAD^l = \frac{1}{I}\sum_{i=1}^{I}\left|\Delta^l(i) - M^l\right|, \tag{26}$$

where M is the median of the errors $M^l = median(\Delta^l)$. Using multiple model prognostic methods a high value of MAD indicates that the method does not show a clear tendency towards a prognosis model. When a method changes frequently, the favored model for each prediction the error consequently spreads.

4.1.3. Prognostic Horizon

The Prognostic Horizon (PH) is determined by the time when the predicted RUL remains stable within a given constant error bound. The upper and lower accepted error limit depend on the accuracy value α, therefore, the metric can be written as

$$[1-\alpha]\cdot r_*^l \leq r^l(i) \leq [1+\alpha]\cdot r_*^l. \tag{27}$$

Figure 7 illustrates the PH. The predicted RUL approaches the true RUL over the time and finally stabilizes for $\lambda \geq 0.6$. The PH is defined by the remaining time until the system failure occurs. In the following evaluation, the PH is expressed as normalized time range. It is clearly visible that the higher the PH the better the performance of a method. Throughout the evaluation the accuracy value was set to $\alpha = 0.1$.

4.1.4. Prognostic Precision

Whereas the PH observes the estimated RUL the Prognostic Precision (PP) considers the uncertainty of the RUL, which is specified by the lower and upper predicted limit of the RUL. The metric is specified by the time the limits remain stable within a constant error bound. Figure 7 shows the determination of PP, the limits of the prediction converge after $\lambda \geq 0.7$. Consequently, the metric allows a statement about how the prognosis method is able to reduce the uncertainty of a forecast as more information accumulates over time.

4.2. Evaluation Procedure

In order to investigate the performance change depending on the number of training data sets, the evaluation of the three methods was organized as follows: By means of the model equations presented in section 3 we generated a data pool of 40 UUTs, which was subdivided in training and test data. The training data contains 15 degradation courses, whereas the test data consists of 25 UUTs.

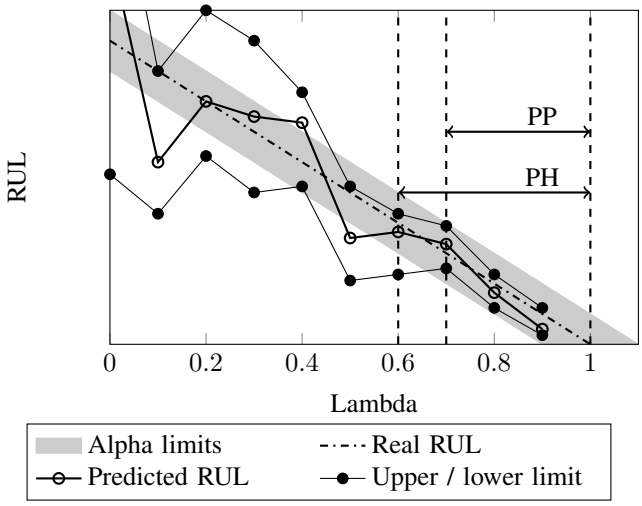

Figure 7. Illustration of the prognosis horizon and prognosis precision

As a first step, we trained each prognosis method by the first training data set and determined the presented metrics for all 25 test data sets by using the estimated RUL and uncertainties of nine predictions at the time $\lambda = 0.1, 0.2, ..., 0.9$. Then we included the next training data set and tested again all 25 data sets. This procedure was repeated until all 15 training data sets were available for the three prognostic methods. The obtained results for a specific metric, e.g. MAPE, can be summarized as shown in diagram 8.

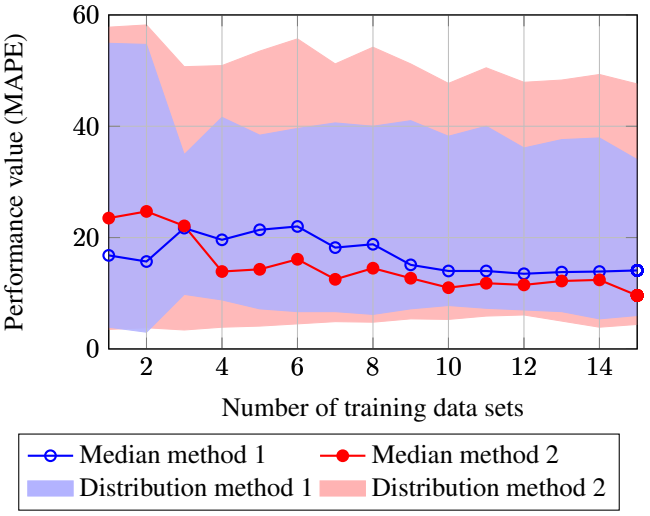

Figure 8. Course of the MAPE at an increasing number of training data sets (comparison of two prognostic methods)

The results lead to a distribution of the MAPE metric, since the performance naturally varies from one tested UUT to another. The figure illustrates the evolution of this distribution for two prognostic methods. The distribution cannot be considered as a Gaussian distribution. It is helpful to rely on the

median and percentiles when discussing the distribution. In the following analysis the 10^{th} and 90^{th} percentile are used. Accordingly, the displayed distribution in the figure covers 80 percent of all predictions or in other words it presents the results of 20 UUTs.

4.3. Enhancement of Metrics

By comparing the courses of the medians in figure 8, method 1 reveals a better performance using less training data sets but is easily outperformed by the second method. Whereas method 2 improves the metrics as more historical data is available, the performance of the first method even deteriorates at the beginning and benefits later from the training data. This deterioration indicates difficulties of method 1 in handling the trained run-to-failure data and selecting the appropriate model for the prediction. According to the course of the median, one tends to rely on the second method since a better performance is reached even with less training data. Involving the distribution in the decision shows that method 2 has a higher range in which the performance is located. This means that the second method reached more often worse performance values by the prediction of the RUL. This motivates to involve the course of the median as well as the distribution in order to assess the performance of prognostic methods.

For further discussion we enhance the aforementioned notations by the following:

N	Total number of training data sets
n	Number of applied data sets for the prediction
$MAPE(n)_m$	Median of the distribution (testing L UUTs)
$MAPE(n)_d$	Difference between the upper and lower percentile of the distribution (testing L UUTs)
$MAPE_{m,N}$	Rating of $MAPE(n)_m$ $n = 1, 2, ..., N$
$MAPE_{d,N}$	Rating of $MAPE(n)_d$ $n = 1, 2, ..., N$

This is done by the example of the performance metrics MAPE. Of course, this notation can be transferred to other metrics. Instead of taking the MAPE metrics to assess the performance, we examine the obtained values for $MAPE_{m,N}$ and $MAPE_{d,N}$. Thereby, the change of MAPE over the number of training data is taken into account. Moreover, the range within the performance value varies over the number UUTs is considered. In the following, we introduce a straight-forward method to quantify both values. Again, this approach is assignable to other metrics.

In a first attempt to determine the values appropriately, the mean values of $MAPE(n)_m$ and $MAPE(n)_d$ are considered. In this way, all reached performance values are independently of the number of used training data. Thus, deterioration or improvements of the observed metric are not covered by this method. To include the course of the metric, we suggest to calculate the values by means of the weighted mean value w.r.t the number of training data sets. The equations can be written as:

$$MAPE_{m,N} = \frac{\sum_{i=1}^{N}(i \cdot MAPE(i)_m)}{\sum_{i=1}^{N} i} \quad (28)$$

$$MAPE_{d,N} = \frac{\sum_{i=1}^{N}(i \cdot MAPE(i)_d)}{\sum_{i=1}^{N} i}. \quad (29)$$

Weighting the performance by the number of training data has several effects: First, the performance using less data has a lower influence on the final result. Since the performance at the beginning strongly depends on the order of trained data sets this is a desired consequence. A change in the training order would have a high impact on the determined values. Additionally, with an increasing number of used training data sets a prognosis method should exhibit an improvement or at least a stable behavior of the performance. Therefore, the weighted mean downgrades occurred deterioration in case of more historical data.

Figure 9 illustrates the difference of the results for $PH_{m,N}$ obtained by the mean value and weighted mean value. In the diagram the course of the median $PH(n)_m$ of two prognosis methods is displayed. Whereas the $PH_{m,15}$ value determined by the mean assesses both methods almost similarly, the weighted mean leads to a better distinctness, as the second method shows no stable improvement of the prognosis horizon.

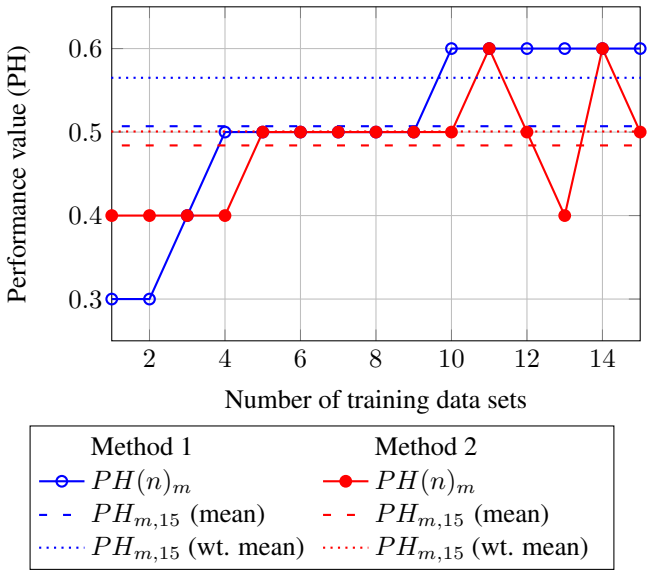

Figure 9. Course of the PH at an increasing number of training data sets (comparison of two prognostic methods)

5. RESULTS AND DISCUSSION

Table 1 summarizes the obtained results of the three prognostic approaches. The performance values are determined by the weighted mean according to the introduced evaluation concept. As we expected, the results show similar values in most categories which is explained by the fact that the three prognostic methods are based on the same regression modelling tool and training data sets. However, since the prognostic approaches manage the trained prognosis models in a different way, it is worth investigating the evolution of the performance according to the available training data sets.

Performance	GPUKF	GPMMM	GPPF
$MAPE_{m,15}$	12.07	13.12	**11.80**
$MAD_{m,15}$	**1.91**	3.41	2.34
$PH_{m,15}$	0.88	0.62	**0.89**
$PP_{m,15}$	0.34	**0.66**	0.45
$MAPE_{d,15}$	31.60	28.62	**27.68**
$MAD_{d,15}$	**10.21**	12.94	11.16
$PH_{d,15}$	0.71	0.71	**0.68**
$PP_{d,15}$	0.80	0.73	**0.64**

Table 1. Enhanced performance metrics of the three prognosis algorithms

Figure 10 shows the improvement of the $MAPE(n)_m$ value over the training data. All prognostic algorithms strongly benefit from the first training data sets and settle down at a similar mean absolute error. It is interesting to note that instead of remaining stable on the achieved performance level, the three methods behave differently with a rising knowledge about the system. Whereas the GPPF is able to further improve the metric, the performance of the GPMMM approach deteriorates. Regarding table 1, this leads to a reduction of $MAPE_{m,15}$ value. The GPMMM also reveals a weakness w.r.t the MAD metric displayed in figure 11. In contrast to the other approaches, the GPMMM exhibits less precision with a raising number of historical data. Given that the MAD value can be considered as an indicator of a prognostic method's tendency towards models, the GPMMM seems to struggle with the selection of a correct prognostic model. Consequently, the MAPE and MAD value of the GPMMM increase. Furthermore, this impact is also observable by looking at the dissatisfactory PH value. One explanation is the applied transition matrix **H** that - in the case at hand - permits a fast transition from model i to another model j so that GPMMM alternates between several models. The increased system knowledge has less influence on the GPUKF, which is reasoned by the manner the training data is stored. Including an additional training data set does not essentially change the basic orientation of the one applied prognosis model.

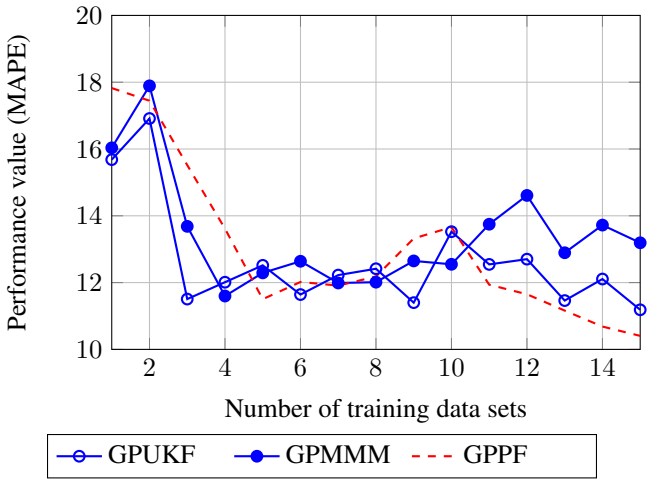

Figure 10. Course of the MAPE at an increasing number of training data sets

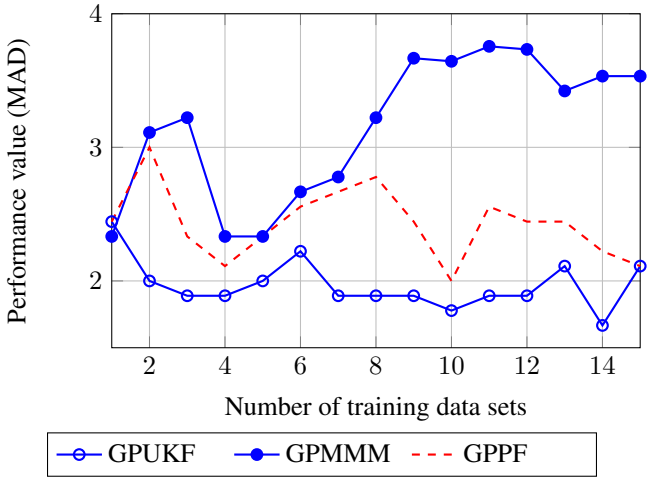

Figure 11. Course of the MAD at an increasing number of training data sets

It is evident that the GPMMM reveals the best PP value, which indicates that the predicted lower and upper RUL limit enter the defined error bound earlier. This is reasoned by the fact that the variance of the forecast is limited artificially (see section 2.3). Thus, the predicted lower and upper limit of the RUL are close. Hence, we learned that the used evaluation concept lacks of a metric which specifies the quality of the predicted error bound. In the current concept, keeping the variance as low as possible will always end in a good performance. A metric which assesses the meaningfulness of the variance is not implemented.

Another aspect of the evaluation is the investigation of the distribution values in Table 1. As described in section 4.3 the values indicate the range within the corresponding metric over the 25 tested UUTs is spreaded. GPPF reaches better results than the other approaches in three of the four criteria.

Nevertheless, there is no noticeable difference to the other methods and all values show that a considerable part of the tested UUTs is predicted with a deviating performance than indicated by the weighted means of the median's course. In other words, whereas the $PH_{m,15}$ values point to a high accuracy of the methods, the $PH_{d,15}$ values reveal that this performance is not always achieved. Especially considering safety relevant systems, this issue should not be neglected.

6. CONCLUSION AND OUTLOOK

In this paper we have presented three data-driven prognosis algorithms. Each algorithm is based on the Gaussian Process to generate prognosis models by means of training data sets. Nevertheless, the way they rate and select suitable models for the estimation of the RUL differs.

One purpose of this paper was to suggest a method to include the training process of a prognosis algorithm in the evaluation process. A simple way is presented to assess the trend of performance metrics at an increasing number of provided training data sets. Moreover, the presented evaluation concept considers the fact that a prognosis method does not constantly reach the same accuracy and precision by testing several UUTs.

Another purpose was to investigate the training process of three data-driven prognosis methods. The results show that the prognosis methods do not automatically benefit from more knowledge about the degradation processes of a system. A particularly motivation was, whether a single GP approach or a Multiple Model Method is preferable when training an arbitrary number of training data sets. The obtained results indicate that GPUKF reaches slightly better performance, especially since the applied GPMMM approach reveals a weakness by managing a high number of prognosis models. In contrast, the single GP approach converges towards a constant performance. Combining the Gaussian Process with a Particle Filter shows the best results and provides a more straight forward possibility to handle the model uncertainties in comparison to the UKF. Of course, the conclusion are strongly depending on the chosen data pool and evaluation concept.

As a result effort is going to put to an enhancement of the mathematical degradation model and thus to generate various data pools to obtain a more comprehensive fundament for the evaluation. We plan to enhance the presented model by an additional load input and to replace the fixed failure threshold by a hazard model, which simulates varying failure limits of UUTs. Furthermore, in order to increase the informative value of the results, other regression modelling concepts like Relevance Vector Machines etc. will be examined under same conditions.

We do not claim the presented evaluation concept near complete, since there is still enough room for improvement. Focus of future work is to include more performance metrics. Especially, a metric to determine the quality of the predicted uncertainty is required. An emerged drawback is that the assessment of a prognosis method suffers from the increased number of available performance indicators, since one metric value is replaced by two. A further bottleneck of the described evaluation concept is that a comprehensive data pool is necessary. Using simulated data this should be no problem. However, generating such a data pool by means of real data is a long and costly work.

ABBREVIATIONS

CBM	Condition Based Maintenance
ESS	Effective Sample Size
GP	Gaussian Process
IMM	Interacting Multiple Model
MAD	Mean Absolute Deviation
MAPE	Mean Average Percentage Error
MMM	Multiple Model Method
PDF	Probability Density Function
PF	Particle Filter
PH	Prognosis Horizon
PP	Prognosis Precision
RUL	Remaining Useful Lifetime
UKF	Unscented Kalman Filter
UUT	Unit Under Test

REFERENCES

Anger, C., Schrader, R., & Klingauf, U. (2012). Unscented kalman filter with gaussian process degradation model for bearing fault prognosis. *Proceedings of First European Conference of the Prognostics and Health Management Society 2012*, 202–213.

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188. doi: 10.1109/78.978374

Ferris, B., Hähnel, D., & Fox, D. (2006). Gaussian processes for signal strength-based location estimation. In *In proc. of robotics science and systems.*

Julier, S. J. (2002). The scaled unscented transformation. In *American control conference, 2002. proceedings of the 2002* (Vol. 6, pp. 4555–4559).

Ko, J., Klein, D. J., Fox, D., & Haehnel, D. (2007). Gp-ukf: Unscented kalman filters with gaussian process prediction and observation models. In *Intelligent robots and systems, 2007. iros 2007. ieee/rsj international conference on* (pp. 1901–1907).

Li, X. R., & Jilkov, V. P. (2003). Survey of maneuvering target tracking. part i. dynamic models. *Aerospace and Electronic Systems, IEEE Transactions on*, *39*(4),

1333–1364.

Orchard, M. E., & Vachtsevanos, G. J. (2009). A particle-filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*, *31*(3-4), 221–246. doi: 10.1177/0142331208092026

Rasmussen, C. E. (2006). Gaussian processes for machine learning.

Saha, B., Goebel, K., & Christophersen, J. (2009). Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Transactions of the Institute of Measurement and Control*, *31*(3-4), 293–308. doi: 10.1177/0142331208092030

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for evaluating performance of prognostic techniques. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–17).

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2009). Evaluating algorithm performance metrics tailored for prognostics. In *Aerospace conference, 2009 ieee* (pp. 1–13).