Mechanical Systems and Signal Processing 82 (2017) 148–165

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

Chi-squared smoothed adaptive particle-filtering based prognosis ☆, ☆☆

Christopher P. Ley [a,b,*], Marcos E. Orchard [a,c]

[a] Department of Electrical Engineering, Universidad de Chile, Santiago 837-0451, Chile
[b] Commonwealth Science and Industrial Research Organisation (CSIRO) Chile, Avda. Tupper 2007, Santiago 837-0451, Chile
[c] Advanced Mining Technology Center (AMTC), Universidad de Chile, Santiago 837-0451, Chile

ARTICLE INFO

Article history:
Received 26 November 2015
Received in revised form
8 April 2016
Accepted 15 May 2016
Available online 24 May 2016

Keywords:
Particle filtering
Monte-Carlo filter
Failure prognosis
Chi-square smoothing
Artificial evolution
Oil degradation

ABSTRACT

This paper presents a novel form of selecting the likelihood function of the standard sequential importance sampling/re-sampling particle filter (SIR-PF) with a combination of sliding window smoothing and chi-square statistic weighting, so as to: (a) increase the rate of convergence of a flexible state model with artificial evolution for online parameter learning (b) improve the performance of a particle-filter based prognosis algorithm. This is applied and tested with real data from oil total base number (TBN) measurements from three haul trucks. The oil data has high measurement uncertainty and an unknown phenomenological state model. Performance of the proposed algorithm is benchmarked against the standard form of SIR-PF estimation which utilises the Normal (Gaussian) likelihood function. Both implementations utilise the same particle filter based prognosis algorithm so as to provide a common comparison. A sensitivity analysis is also performed to further explore the effects of the combination of sliding window smoothing and chi-square statistic weighting to the SIR-PF.

1. Introduction

Condition Based Maintenance [1] (CBM) is a relative new paradigm that, using a prediction of the degrading system, optimises the maintenance process. In this regard, CBM relies heavily on the concept of prognosis. “Prognosis is the science of predicting the health condition of a system and/or its components based upon knowledge of past usage, current state, and future conditions” [2]. As such, prognosis encapsulates both state estimation and prediction. Recursive Bayesian algorithms are well suited to real-time estimation and prediction since they incorporate process data into prior state estimates by considering the likelihood of measured values [3]. The underlying uncertainty can then be projected for n steps into the future using state models, thus providing a realisation of the future uncertainty of the state [4].

Particle Filters (PFs) or Sequential Monte Carlo (SMC) are a form of recursive Bayesian algorithms that provide a solid and consistent theoretical framework to handle model non-linearities or non-Gaussian process/observation noise. PFs are based on the concept of sequential importance sampling (SIS) and Bayesian Theory, which have been proven in literature to be a

☆ A CORFO Project with funding provided under: Proyecto Innova CSIRO-CHILE 10CEII-9007.
☆☆ This work has been partially supported by FONDECYT Chile Grant no. 1140774 and Advanced Center for Electrical and Electronic Engineering, Basal Project FB0008.
* Corresponding author.
E-mail addresses: christopher.ley@csiro.au (C.P. Ley), morchard@ing.uchile.cl (M.E. Orchard).

http://dx.doi.org/10.1016/j.ymssp.2016.05.015
0888-3270/

highly effective method of online estimation. PFs do not use the assumption of Normally distributed process noise, as in the case of the Kalman Filter, thus allowing a more accurate representation of the true state evolution in non-Gaussian stochastic processes [5–9,4,10,11,2]. The PF framework has also been utilised in literature as a tool for prognosis [6,4,10,11], as the particle population can be projected by the state models. The resulting projection can then be used as a representation of the uncertainty of future state evolution. This is particularly useful in cases where an analytical solution to the future uncertainty probability density function (pdf) is difficult or impossible to calculate.

In general, prognosis algorithms are sensitive to the uncertainty associated with the current state estimate. Prognosis routines utilise a stochastic state model to propagate state estimates into the future, and thus all state predictions can only be more uncertain. In fact, the increase in future uncertainty follows the rules of information *entropy* [12,13] (Shannon entropy), which indicates that if the system includes a stochastic component then *entropy* will only increase. Particle Filtering based Prognosis (PFP) algorithms are no different to any other prognosis routine: uncertainty associated with current estimates can only increase when projected into the future. However, as PFP algorithms project the current likelihood into the future by utilising the state process model, errors in that model exacerbate problems related to prediction uncertainty by introducing additional biases. With this in mind, it is natural to aim at improving the accuracy of PF-based prognostic results by reducing the uncertainty associated with the state estimates that we use as initial condition for prediction algorithms; ensuring that the state transition model accurately reflects the system evolution in time.

A suggested solution to the problem of a biased process model is to introduce an additional parameter that adapts to the observed state evolution, this is often referred to as an Artificial Evolution (AE) [9]. The introduced parameter is shaped by a process resembling simulated annealing [14]. The resulting process model accurately reflects the observed state evolution, but introduces additional hurdles for PFP algorithms. Due to the time associated with the convergence of unknown system parameters, AE naturally incorporates a time delay that constraints the moment in which prognosis can be accurately performed. The convergence of AE can also be a problem given situations of high measurement uncertainty.

The Chi-squared likelihood weighting and smoothing window introduced in this paper serves to alleviate the shortcomings associated with the introduction of the AE parameter, effectively improving the precision of prognosis results. The results are demonstrated with total base number (TBN) oil degradation data in comparison with the traditional PF algorithm with a Normally weighted likelihood function so as to create a benchmark for performance. A sensitivity analysis is also performed to explore the effects of AE in the chi-squared algorithm. The TBN measurements exhibit many of the characteristics that are undesirable in estimation applications, namely high measurement uncertainty and an unknown phenomenological model. A novel performance index is also presented to demonstrate a statistical improvement in the precision of obtained results.

This paper is divided as follows:

- Section 2, which provides the background and previous work in literature utilised as the basis of the Particle Filtering based Prognosis
- Section 3, introduces the case study utilised in this paper as well as the motivation for using this data. This data is haul truck total base number measurements from oil.
- Section 4, introduces the Chi-squared Smoothed Adaptive Particle-Filtering based Prognosis.
- Section 5 analyses the effects of the Chi-squared kernel on various aspects of the algorithm, as well as delivering insight into the effects of selecting certain parameters in the Chi-squared Smoothed Adaptive Particle-Filtering based Prognosis algorithm.
- Section 6 presents the application of the Chi-squared Smoothed Adaptive Particle-Filtering based Prognosis to the TBN case study as well as the Normalised Integral of Precision Index for validating the results.
- Section 7 offers Conclusions

2. Particle-filtering based prognosis

"*Particle Filters* (PF) or *Sequential Monte Carlo* (SMC) methods are a set of simulation-based methods which provide an attractive approach to computing the posterior distributions" [3]. PF are very flexible, parallelisable and applicable to any non-linear or linear system. The advent of cheap and formidable computational power in conjunction with the recent proliferation of scientific papers on PF & SMC have led to their wide spread application to many fields, particularly in the area of control and observation. Several closely related algorithms, under the names of *bootstrap filters, condensation, Monte Carlo filters, interacting particle approximations, Sequential Importance sampling filter* and *survival of the fittest* have appeared in several research fields [3]. This paper focuses on an implementation of a derivative of PF called *Sequential Importance Sampling and Re-sampling filter* (SIR).

We first introduce the general continuous PF to demonstrate the underlying concept and to highlight the flexibility when selecting the transition kernel and the likelihood function that can be applied without changing the underlying assumptions (and the convergence characteristics). This flexibility in the likelihood function selection will be a key aspect exploited later in this paper (Section 4.1)

2.1. Bayesian estimation

Let $X = \{X_t,\ t \in \mathbb{R}\}$ be a $\mathbb{R}^{n_x}$-valued Markov process characterised by both its initial distribution $p(x_0)$ and the transition probability $p(x_t|x_{t-1})$. Furthermore, let $p(x_t|x_{t-1})$ be defined by (1), where $\{\omega_t\}_{t\geq 0}$ is a sequence of independent random variables, not necessarily Gaussian.

$$x_t = f_t(x_{t-1},\ \omega_t) \tag{1}$$

Noisy observations $Y = \{Y_t,\ t \in \mathbb{N}\}$ are assumed to be conditionally independent, given $X = \{X_t,\ t \in \mathbb{N}\}$. Eq. (2) defines the marginal distribution $p(y_t|x_t)$ where $\{v\}_{t\geq 0}$ is a sequence of independent random variables.

$$y_t = g_t(x_t,\ \nu_t) \tag{2}$$

Let $x_{0:t} \triangleq \{x_0,\ \ldots,\ x_t\}$ and $y_{1:t} \triangleq \{y_1,\ \ldots,\ y_t\}$ denote, respectively, the signal and the observations up to time t. It is often desired to estimate the *posterior distribution* $p(x_{0:t}|y_{1:t})$, the marginal distribution $p(x_t|y_{1:t})$, and the expectations (3) for any function $f_t\colon \mathbb{R}^{n_x} \to \mathbb{R}^{n_{ft}}$ integrable with respect to $p(x_{0:t}|y_{1:t})$, [3].

$$I(f_t) = E_{p(x_{0:t}|y_{1:t})}[f_t(x_{0:t})] \triangleq \int f_t(x_{0:t})p(x_{0:t}|y_{1:t})dx_{0:t} \tag{3}$$

Posterior distribution estimation can basically achieved by performing two sequential steps, *prediction* and *correction*.

1. *Prediction*: consists of using both the knowledge of the previous state estimate and the process model to generate the *a priori* state pdf estimate for the next time instant:

$$p(x_{0:t}|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{0:t-1}|y_{1:t-1})dx_{0:t-1} \tag{4}$$

2. *Correction*: generates the *posterior* state pdf using Bayes formula and is implemented by using the recursion formula:

$$p(x_{0:t}|y_{1:t}) \propto p(y_t|x_t)\cdot p(x_t|x_{0:t-1})\cdot p(x_{0:t-1}|y_{1:t-1}) \tag{5}$$

Expressions (3), (4) and (5) do not have analytical solutions in most cases. In this sense, PF & SMC algorithms make their evaluations statistically through effective sampling strategies [3,4]

2.2. The particle filter

Consider a sequence of probability distributions $\pi_t(x)_{t\geq 1}$, where it is assumed that $\pi_t(x_{0:t})$ can be evaluated pointwise up to a normalising constant. PF are a class of algorithms designed to approximately obtain samples sequentially from $\{\pi_t\}$ to generate a collection of $N \gg 1$ weighted random samples $\{w_t^{(i)},\ x_{0:t}^{(i)}\}_{i=1,\ldots,N}$, $w_t^{(i)} \geq 0,\ \ \forall\ t \geq 1$, satisfying [5]:

$$\sum_{i=1}^{N} w_t^{(i)}\varphi_t(x_{0:t}^{(i)}) \xrightarrow[N\to\infty]{} \int \varphi_t(x_{0:t})\pi_t(x_{0:t})dx_{0:t}, \tag{6}$$

where φ_t is any π_t-integrable function. In the particular case of the Bayesian Filtering problem, the *target distribution* $\pi_t(x_{0:t}) = p(x_{0:t}|y_{1:t})$ is the *posterior* pdf of $X_{0:t}$, given a realisation of noisy observations $Y_{1:t} = y_{1:t}$. Using (1) and (2), $\pi_t(x_{0:t})$ may be written as [3]

$$\pi_t(x_{0:t}) = p(x_0)\prod_{k=1}^{t} f_k(x_k|x_{k-1})g_k(y_k|x_k) \tag{7}$$

Let a set of N paths $\{x_{0:t-1}^{(i)}\}_{i=1,\ldots,N}$ be available at time $t-1$. Furthermore, let these paths distribute according to $q_{t-1}(x_{0:t-1})$, also referred to as the *importance* density function at time $t-1$. Then, the objective is to efficiently obtain a set of N new paths (particles) $\{\tilde{x}_{0:t}^{(i)}\}_{i=1,\ldots,N}$ approximately distributed according to $\pi(\tilde{x}_{0:t})$ [5].

For this purpose, the current paths $x_{0:t-1}^{(i)}$ are extended by using the kernel $q_t(\tilde{x}_{0:t}|x_{0:t-1}) = \delta(\tilde{x}_{0:t-1} - x_{0:t-1})\cdot q_t(\tilde{x}_t|x_{0:t-1})$, ie, $\tilde{x}_{0:t} = (x_{0:t-1},\ \tilde{x}_t)$. The *importance sampling* procedure generates consistent estimates for (3), by approximating (7) with the empirical distribution [5]

$$\tilde{x}_t^N(x_{0:t}) = \sum_{i=1}^{N} w_{0:t}^{(i)}\delta(x_{0:t} - \tilde{x}_{0:t}^{(i)}) \tag{8}$$

where $w_{0:t}^{(i)} \propto w_{0:t}(\tilde{x}_{0:t}^{(i)})$ and $\sum_{i=1}^{N} w_{0:t}^{(i)} = 1$.

2.3. Sequential Importance Sampling/Resampling Particle Filter (SIR-PF)

The simplest PF implementation - the SIS particle filter - computes the value of the particle weights $w_{0:t}^{(i)}$, by setting the

importance density function equal to *a priori* pdf for the state, ie, $q_t(\tilde{x}_{0:t}|x_{0:t-1}) = p(\tilde{x}_t|x_{t-1}) = f_t(\tilde{x}_t|x_{t-1})$. In that manner, the weights for the newly generated particles are evaluated from the likelihood of new observations. The efficiency of the procedure improves as the variance of the importance weights is minimised. The choice of the importance density function is critical for the performance of the particle filter scheme and hence, it should be considered in the filter design.

One of the main difficulties that must be addressed in the implementation of SIS particle filters is the problem of *particle degeneracy* [3] since, after a few iterations, all but one particle could have a negligible weight [5,3]. Several methods have been proposed to overcome this problem, one of which is to perform the re-sampling procedure conditionally on a test of the *Effective Sampling Size* (N_{eff}) criterion, given by [15]:

$$N_{eff} = \left(\sum_{i=1}^{N} \left(\omega_t^{(i)} \right)^2 \right)^{-1} \text{ s. t. } \sum_{i=1}^{N} \omega_t^{(i)} = 1 \tag{9}$$

The re-sampling is only implemented when the N_{eff} is below a threshold N_T, e.g. $N_T = N/2$, where N is the size of the particle population. This process avoids *sample impoverishment* and permits the SIR to converge to the optimal filter, in the weak sense, with the rate of $1/\sqrt{N}$.

Considering the latter points, the algorithm for the sampling importance resampling (SIR) particle filter is as follows [3,4]:

Sequential Importance Sampling Resampling (SIR) Particle Filter.

1. Importance sampling:
   - For $i = 1, \ldots, N$, sample $\tilde{x}_t^{(i)} \sim \pi(x_t|\tilde{x}_{0:t-1}^{(i)}, y_{0:t})$ and set $\tilde{x}_{0:t}^{(i)} \triangleq (x_{0:t-1}^{(i)}, \tilde{x}_t^{(i)})$.
   - Evaluate the importance weights:

$$w(\tilde{x}_{0:t}^{(i)}) = w_{0:t-1}^{(i)} \cdot \frac{p(y_t|\tilde{x}_t^{(i)}) p(\tilde{x}^{(i)}|x_{0:t-1}^{(i)})}{q_t(\tilde{x}^{(i)}|x_{0:t-1}^{(i)})} \tag{10}$$

   - Normalise the weights:

$$w_{0:t}^{(i)} = w(\tilde{x}_{0:t}^{(i)}) \cdot \left(\sum_{i=1}^{N} w(\tilde{x}_{0:t}^{(i)}) \right)^{-1} \tag{11}$$

2. Resampling algorithm:
   - If $N_{eff} \geq N_T$ $\breve{x}_{0:t}^{(i)} = \tilde{x}_{0:t}^{(i)}$ for $i = 1, \ldots, N$; otherwise
   - For $i = 1, \ldots, N$, sample an index $j(i)$ distributed according to a discrete distribution satisfying $P(j(i) = l) = w_t^{(l)}$ for $l = 1, \ldots, N$.
   - For $i = 1, \ldots, N$, $\breve{x}_{0:t}^{(i)} = \tilde{x}_{0:t}^{j(i)}$ and $\breve{w}_t^{(i)} = N^{-1}$

   After the resampling procedure, the new particle population $\{\breve{x}_{0:t}^{(i)}\}_{i=1,\ldots,N}$ is an independent and identically distributed (i.i.d) sample of the empirical distribution (12), thus the weights are reset to $\breve{w}_t^{(i)} = N^{-1}$.

$$\breve{\pi}_t^N(x_{0:t}) = \frac{1}{N} \sum_{i=1}^{N} N_t^{(i)} \delta(x_{0:t} - \tilde{x}_{0:t}^{(i)}) = \frac{1}{N} \sum_{i=1}^{N} \delta(x_{0:t} - \breve{x}_{0:t}^{(i)}) \tag{12}$$

2.4. Particle-filtering based prognosis

It is possible to describe the evolution in time of the degradation of the non-linear state equations used in the estimation phase (Section 2.3). By using the state equation (1), it is possible to generate long-term predictions using kernel functions to reconstruct the estimate of the state pdf in future time instants.

The goal is to produce a predicted conditional state pdf $\hat{p}(x_{t+k}^{(i)}|\hat{x}_{t+k-1}^{(i)})$, which describes the state distribution at the future time instant $t + k$, $(k = 1, \ldots, \rho)$ when the particle $\hat{x}_{t+k-1}^{(i)}$ is used as an initial condition. Assuming that the current weights $\{w_t^{(i)}\}_{i=1,\ldots,N}$ are a good representation of the state pdf at time $t + k$, by using the law of total probabilities and the particle weights at time $t + k - 1$, as shown in (13)

$$\begin{aligned} &\hat{p}(x_{t+k}|\hat{x}_{1:t+k-1}) \approx \sum_{i=1}^{N} w_{t+k-1}^{(i)} \cdot \hat{p}(x_{t+k}^{(i)}|\hat{x}_{t+k-1}^{(i)}), \\ &\hat{x}_t^{(i)} = \tilde{x}_t^{(i)}, \quad k = 1, \ldots, \rho \end{aligned} \tag{13}$$

To evaluate (13), the weight of every particle should be modified (at each prediction step) to take into account the fact that

noise and process non-linearities could change the shape of the state pdf as time passes. However, since the weight update procedure is needed as part of a prediction problem, it cannot depend on the acquisition of new measurements. Additionally, before proceeding with the next prediction step, it is necessary to allocate a new set of particles within the domain of the probability distribution (13). To overcome most of these difficulties Orchard et al. [4] proposed a method of using a kernel to represent the uncertainty growth of each prediction step.

Consider a discrete approximation (14) for the predicted state pdf (13), where $K(\cdot)$ is a kernel density function, which may correspond to the process noise pdf, a Gaussian Kernel or a rescaled Epanechnikov Kernel [4].

$$\hat{p}(x_{t+k}|\hat{x}_{1:t+k-1}) \approx \sum_{i=1}^{N} \omega_{t+k-1}^{(i)} K\left(x_{t+k} - E\left[x_{t+k}^{(i)}|\hat{x}_{t+k-1}^{(i)}\right]\right) \tag{14}$$

The kernel method utilised is a modified version of the *Regularised Particle Filter* presented by Musso et al. [8], where they present an optimum rescaled Epanechnikov kernel, given by:

$$A = \left(8c_{n_x}^{-1}(n_x+4)(2\sqrt{\pi})^{n_x}\right)^{1/(n_x+4)} h_{opt} = A \cdot N^{-1/(n_x+4)} K_h = \frac{1}{h^{n_x}} K\left(\frac{x}{h}\right)$$

$$K(x) = \begin{cases} \dfrac{n_x+2}{2c_{n_x}}\left(1 - \| x \|^2\right) & \text{if } \| x \| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

where c_{n_x} is the volume of the unit sphere in $\mathbb{R}^{n_x}$. Rather than simply projecting the expectation of the state variables into the future, the uncertainty of (14) can be represented by generating a population of equally weighted particles at time instant $t+k, 1 \le k \le \rho$, and performing an inverse transform re-sampling procedure for the particle population. The focus is to represent the growth of uncertainty presented in (14). For that, samples distributed according to (14) are obtained by selecting $u^{(i)} = i \cdot (N+1)^{-1}$ for i: 1, …, N and interpolating a value for $\hat{x}_{t+k}^{(i)}$ from the cumulative state distribution

$$F(X_{t+k} \le x_{t+k}) = \int_{-\infty}^{x_{t+k}} \hat{p}(x_{t+k}|\hat{x}_{1:t+k-1}) dx_{t+k} \tag{16}$$

in accordance with $\hat{x}_{t+k}^{(i)} = F^{-1}(u^{(i)})$ [4]. Considering all of the above, the regularisation algorithm [8] is used to perform prognosis by [4]:

1. Apply modified inverse transform resampling procedure for $i = 1, \ldots, N$, where $\omega_{t+k}^{(i)} = N^{-1}$
2. Calculate $\hat{S}_{t+k}$, the empirical covariance matrix of

$$\left\{ E\left[x_{t+k}^{(i)}|\hat{x}_{t+k-1}^{(i)}\right], \omega_{t+k}^{(i)} \right\}_{i=1}^{N} \tag{17}$$

3. Compute $\hat{D}_{t+k}$ such that $\hat{D}_{t+k}\hat{D}_{t+k}^{T} = \hat{S}_{t+k}$
4. For $i = 1, \ldots, N$, 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 calculated as in (15) and $\hat{x}_{t+k}^{(i)*}$ is the regularised particle population.
5. For the next prediction step $\hat{x}_{t+k-1}^{(i)*} = \hat{x}_{t+k-1}^{(i)}$

It is assumed that the state covariance matrix $\hat{S}_{t+k}$ is equal to the empirical covariance of $\hat{x}_{t+k}$ and that a set of equally weighted samples for $\hat{x}_{t+k-1}$ is available. This serves to maximise the efficiency of the Epanechnikov Kernel.

3. Particle Filtering based Prognosis (PFP) applied to Haul Truck Total Base Number (TBN) measurements

Often creating or utilising a phenomenological state model for system estimation is intractable or undesirable e.g. due to high computational load or observability issues. When those situations arise a generalised numerical model is usually desirable. Normally, a generalised model is an approximation of the underlying state evolution which introduces inaccuracy in the state evolution, the effect of which, can be minimised with corrections from measurements. In the prediction phase of prognosis this is no longer the case, thus errors in the state model introduces bias in the predicted probability density function (pdf) of the state. A solution to the problem of a biased process model as suggested in [9] is to utilise Artificial Evolution (AE).

The Haul Truck total base number (TBN) of engine oil measurements, seen in Fig. 1, are an excellent example of a system whose phenomenological model is infeasible. This is primarily due to the reliance on states that are unobservable with only TBN measurements. Due to these observability constraints, we utilise a general stochastic state model with the Particle Filtering based Prognosis (PFB).

It can be seen in Fig. 1 that the TBN data is a periodic monotonically decreasing function, considering each period a different data set. Furthermore, considering only complete periods of Fig. 1, of which there are 14, each set will have the

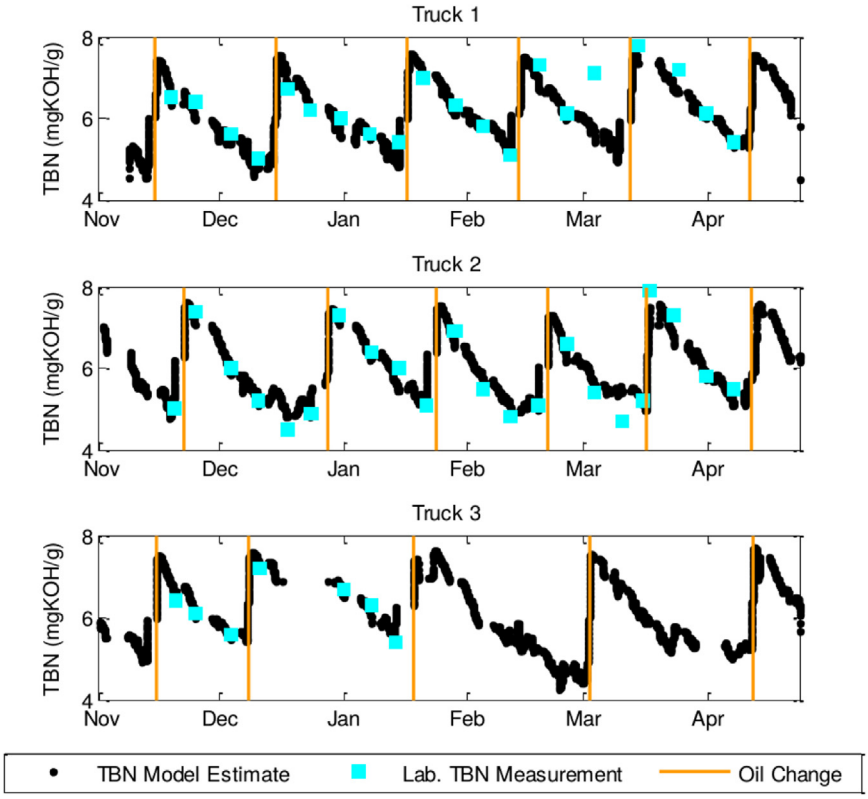

Fig. 1. Example of given TBN truck data [16].

general state model (18).

Our goal is to predict the degradation of each set. The ability to predict when an engine needs its oil changed based on usage, rather than arbitrary time values, will enable the maintenance process to begin to be optimised. The ultimate goal is twofold, predict the local degradation of the oil, and for the algorithm learn the changing parameters of the model as a precursor to an engine age estimation and ultimately an engine degradation prediction.

$$x_1(k+1) = \alpha x_1(k) + w_1(k) \tag{18}$$

We see in (18), $x_1(k)$ is the TBN at time k, w_1 represents uncertainty in the process model, and an estimated degradation constant of α that is deduced from the average degradation of all the sets.

Already it is clear that this model is a loose generalisation as each set has variations in the parameter α. This is can be accounted for by underlying degradation of the haul truck motor over time, from which the oil was taken, as well as variations in the motor age between trucks and their varying usage profiles. We apply the concept of *Artificial Evolution* (AE) [17] by creating an additional state that can be used to correct for parametric errors in the model. This correction is achieved by utilising a process similar to simulated annealing [14] in which process variance for this additional state is reduced artificially over time and a model that simulates *Brownian Motion*, a mathematical model used to describe random movements of particles named after the botanist Robert Brown. This reduction in process covariance, combined with the random search pattern of *Brownian Motion*, and the SIR-PF selecting successful particles, allows the additional state to find an adequate correction for the parametric error in the model. This is described by the state equations:

$$\begin{aligned} x_1(k+1) &= (x_2(k) + \alpha)x_1(k) + w_1(k) \\ x_2(k+1) &= x_2(k) + w_2(k) \end{aligned} \tag{19}$$

where $x_1(k)$ is the TBN at time k, $x_2(k)$ is the introduced state for correcting the parametric error in α and $w_1(k)$, $w_2(k)$ are represented by the normal distributions $N(0, \omega_1^2)$ and $N(0, \omega_2^2)$ respectively. Where x_2 is initially given a relatively large freedom of movement, i.e. a large ω_2^2, so as to represent the uncertainty in the initial knowledge of the degradation of x_1. As the process evolves the process variance is reduced so as to constrict the freedom of movement of x_2 (or the evolution), this ensures the particle system converges on a solution and thereby "learns" the correction of the parametric error of state x_1. The measurement equation in this case is given by

$$y(k) = x_1(k) + \phi(k) \tag{20}$$

where $y(k)$ is the measurements of TBN and $\phi(k)$ represents the noise present in the sensor measurements. Note that $\phi(k)$ is consider white noise and is represented by the normal distribution $N(0, \sigma^2)$.

4. Chi-squared (χ^2) smoothed adaptive particle-filtering

4.1. Artificial evolution, smoothing and the χ^2 kernel

One of the difficulties of using the *Artificial Evolution* (Section 3) approach to learning unknown model parameters arises when the measurements used for estimating the weights are highly noisy; this inhibits the convergence of the PF due to the large posterior uncertainty at each step. In the standard SIR-PF [18] (Section 2.3), measurements are weighted with a importance density function (10). In practice these weights are usually calculated from the error $\epsilon^{(i)}$ of the ith estimation of the measurement (20) and the sensor measurements $\hat{y}(k)$ given by:

$$\epsilon^{(i)}(k) = \hat{y}(k) - y^{(i)}(k) \tag{21}$$

These errors are then weighted utilising a kernel, which is generally a Gaussian Normal kernel given by:

$$w(\tilde{x}_{0:t}^{(i)}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right) \tag{22}$$

where σ^2 is the variance of the measurement noise $\phi(k)$.

A novel approach presented in this paper attempts to assists the convergence of the state introduced by *Artificial Evolution* by modifying the traditional use of the Normal (Gaussian) likelihood, to include knowledge of past measurements, through the use of a *Goodness of fit* [19] measure similar to the Pearson's χ^2 test, thereby reducing the overall uncertainty [20]. The standard chi-square test for goodness of fit is given by [21]

$$\begin{aligned}\chi^2 &\equiv \frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2} + \cdots + \frac{(x_\nu-\mu_\nu)^2}{\sigma_\nu^2} \\ &\equiv \sum_{i=1}^{\nu} \frac{(x_i-\mu_i)^2}{\sigma_i^2}\end{aligned} \tag{23}$$

where the given fluctuations of the values of x_i about the underlying mean values μ_i should vary by $\pm\sigma_i$, thus the ratio of the square error and σ_i^2 will be in the order of unity. Hence if we have chosen the parameters correctly, we expect the value of χ^2 will be approximately equal to ν. If it is, then we may conclude that the data are well described by the estimated states $\{x_i\}_{i=1,\ldots,\nu}$.

We expand this idea and consider that given the correct parameters of the state model (19) that the deviations of the state x_1 (TBN) and therefore (20) should only be the measurement uncertainty given by $\phi(k)$, with the underlying assumption that the measurement error is zero-mean, normally distributed.

This is implemented by the following steps:

1. For a given time k, the state equations (19) are smoothed over M steps and χ^2 error statistic is computed. This is performed by fixing state $x_2(k)$ and computing the jth back propagated estimation of $x_1(k)$, given by:

$$x_1(k-j) = x_1(k)\big(x_2(k)+\alpha\big)^{-j} \tag{24}$$

the χ^2 statistic is then given by the sum of the errors of (24) and the past measurements $y(k-j)$ squared, expressed as:

$$(\chi^2)^{(i)} = \sum_{j=0}^{M} \frac{\left(y(k-j) - \left(x_2^{(i)}(k)+\alpha\right)^{-j} x_1^{(i)}(k)\right)^2}{\sigma^2} \tag{25}$$

where σ^2 is the variance of the measurement noise $\phi(k)$ and $y(k) = x_1(k) + \phi(k)$.

2. The likelihood function then becomes the χ^2 probability density function and the particle weight $\omega^{(i)}(k)$ is computed by

$$w(\tilde{x}_{0:t}^{(i)}) = \frac{\left((\chi^2)^{(i)}\right)^{(\nu-2)/2} e^{-(\chi^2)^{(i)}/2}}{2^{\nu/2}\Gamma(\nu/2)} \tag{26}$$

where $\nu = M$ which is the degrees of freedom of the χ^2 pdf and $\Gamma(\cdot)$ is the gamma function. Please note that a value of $M = 10$ is used for all cases unless otherwise specified.

3. During the initial stages of the algorithm where there is less data than the desired degrees of freedom (i.e. $k < M$) a reduced order back propagation should be used until there is sufficient data. For example during the second step, $k = 2$, a second order back propagation would be used, which implies $M = 2$ in the case of (25). Note that when $M = 1$ this reduces to the Gaussian weighted case.

We will see in the following sections that the smoothed χ^2 weighted window alleviates the learnt parameter x_2 sensitivity to high measurement noise which impedes convergence on the underlying degradation of the state x_1, as well as reduces the estimation variance for each time step, together these factors serve to improve the prognosis accuracy.

5. Choosing parameters and their effects: a sensitivity analysis

The *Chi-squared Smoothed Adaptive Particle-Filtering* (χ^2-PF) was introduced in Section 4 in order to improve the rate of convergence and precision of the AE parameter. With the introduction of a new kernel comes additional parametric freedom that has yet to be explored, namely the degrees of freedom ν in equations (25)–(26). The choice of degrees of freedom, or what we will call backsteps in our implementation, as well as the annealing rate of the AE parameter will directly affect the performance of the χ^2-PF algorithm. By intuition we expect there to be a relationship between the annealing rate, backsteps, measurement noise, and the amount of parametric error in the process model, which we will call drift for the purposes of this paper. Considering all these variables we will explore the statistical performance of both the χ^2-PF algorithm and the standard Normally weighted SIR algorithm. Our goal ultimately, is to apply the optimum version of the algorithm to the TBN data presented in Section 3. It is important to note that, when considering precision, it is important to have a ground truth value to compare results. With this in mind it is proposed that we construct a synthetic set, similar to that found in the TBN data set, where by the parameters can be controlled in order to properly analyse their effects in relation to algorithmic performance.

5.1. The synthetic data set

The synthetic data set is generated from a monotonically decreasing function (27) that follows the form of (19):

$$x_{syn}(k+1) = (\alpha + \beta)x_{syn}(k), \tag{27}$$

where the degradation constant $\alpha = 0.992$ and the drift $\{\beta \in \mathbb{R} | -0.002 \leq \beta \leq 0.002\}$ will be fixed with each separate data set. The initial conditions for $k=0$ will be $x_{syn}(0) = 7.5$. Similarly, the measurement equation (28) will follow that same form as (20):

$$y_{syn}(k) = x_{syn}(k) + N(0, \sigma_{syn}^2) \tag{28}$$

where $\sigma_{syn} = 0.05 \times 10^l$: $\{l \in \mathbb{R} | 0 \leq l \leq 1\}$ and l will be fixed with each separate data set. Both β and σ where divided into 11 linearly spaced sets within their respective ranges totalling a combination of $11 \times 11 = 121$ separate, distinct sets.

Each set of data was then used as measurement data for a SIR Particle Filter routine for $k = 1, \ldots, 50$ time steps. Every distinct set of data was used with every possible combination of parameter variation desired to be analysed. This total routine was then evaluated 60 times, this was performed mainly to eliminate the possibility of anomalous results (or outliers) due to the stochastic nature of the PF routine.

The SIR Particle Filter with AE was chosen to perform the estimation so as to mimic the application to the TBN data case. All parameters that where not part of study where fixed so as to eliminate any biased towards any particular configuration. The parameters varied in the study consisted of:

- Backsteps: Consists of the set $\{M \in \mathbb{N} | 1 \leq M \leq 10\}$ that indicates the number of degrees of freedom used for the χ^2 kernel as well as the number times each particle is propagated backwards in (25) and (26) respectively. Please note that when $M=1$ the Gaussian kernel was used.
- Annealing rate: For each algorithm the AE parameter x_2, as seen in (19), follows a *Brownian Motion* model

$$x_2(k+1) = x_2(k) + N(0, \omega(k)^2) \tag{29}$$

where, given its initial value and its initial uncertainty $\omega(0)^2$, the variance $\omega(k)^2$ of the *Brownian Motion* model is then reduced. The model used to implement the simulated annealing is given by

$$\begin{aligned} \eta &= \eta + 2^{-\xi} \\ \omega(k) &= \omega(0) \times 10^{-\eta} \end{aligned} \tag{30}$$

where the annealing rate $\{\xi \in \mathbb{N} | 0 \leq \xi \leq 4\}$, $\{\eta \in \mathbb{R} | 0 \leq \eta \leq 5\}$ and when $k=0$, we set $\eta = 0$. The annealing rate ξ is fixed for each simulation.

In summary, the sensitivity analysis consists of 60 solutions of every possible combination of parameters (that were studied) and variations in the configuration of the measurement data, totalling 60 iterations $\times$ 10 backsteps $\times$ 5 annealing rates $\times$ 121 data sets $=$ 363000 results.

5.2. Performance evaluation

In order to evaluate the performance of the algorithms with the various configurations, we must first qualify what it is that constitutes an improvement of one result over another. If we consider that the goal of the introduction of the χ^2 kernel was to improve the precision, accuracy as well as the rate in which the learning parameter x_2 in (19) compensates for the parametric error, or "learns" the drift. Then we should consider a metric that evaluates over time both the spread of the x_2 particles (precision), and the bias of the estimation of x_2 with respect to the ground truth (accuracy). With this in mind we introduce a modification on a Precision Index introduced in [22]. The Precision Index (PI) was introduced to measure both the spread and bias simultaneously of a Time-of-Failure pdf but the same concept can be applied to the x_2 parameter. The modified PI (31) lies in the set $\{I \in \mathbb{R} | 0 \leq I \leq 1\}$.

$$I(k) = \exp\left(\frac{-\left| \sup(x_2(k)) - \inf(x_2(k)) \right|}{\left| E[x_2(k)] - \beta \right|} \right) \tag{31}$$

For (31) $\sup(x_2(k))$ and $\inf(x_2(k))$ are the supremum and infimum respectively of the x_2 population at time k, and $E[x_2(k)]$ is the expectation of x_2 population at time k. A $I(k)$ closer to 1 indicates a more precise particle population of x_2. We can also consider how fast $I(k)$ approaches one, a faster convergence to 1 would imply the algorithm becomes accurate quicker. With the aforementioned considerations, coupled with the fact that there is a large quantity of data to compare, we will compute the centroid of the PI evolution in time [23].

Calculating the centroid [24] of a bounded region of $1 - I(k)$, seen in (32) and (33), we have quantified the performance of the complete evolution of $I(k)$ with two centroids for all k (where $k \in \mathbb{N}$, $k \in [1, 50]$).

$$\bar{k} = \frac{1}{A_I} \int_1^{50} k \cdot [1 - I(k)] dk \tag{32}$$

$$\bar{I} = \frac{1}{A_I} \int_1^{50} \frac{\left[1 - I(k)\right]^2}{2} dk \tag{33}$$

$$\frac{1}{A_I} = \int_1^{50} [1 - I(k)] dk$$

A centroid $(\bar{k}, \bar{I})$ closer to the origin $(0, 0)$ indicates a faster more accurate algorithm.

5.3. Centroid analysis and the precision of AE parameter

Given the sheer quantity of results generated we need to consider ways of reducing the dimensionality of the results. We can immediately see that if we take the mean of any variable with identical configurations we reduce the volume of results by a factor of 60 as well as eliminate any anomalous outliers in the data.

5.3.1. Annealing rate

If we fix for example σ_{syn}, as in Fig. 2, we can see the effects of other factors on the centroid movement. We can clearly see in Fig. 2 that if we increase the Annealing Rate $2^{-\xi}$ we shift the centroid $\bar{I}$ towards 0, this implies that a larger Annealing Rate (within the range we have tested) improves precision of the x_2 population. We can see in Table 1 that this trend applies to the larger set of data. Interestingly, the Annealing Rate seems to have little or no impact on the centroid $\bar{k}$, implying it does not have an impact on how quickly the population of x_2 becomes precise.

5.3.2. Gaussian Kernel and χ^2 backsteps

Fixing σ_{syn} for various values and this time analysing the affect of the Gaussian kernel and the various χ^2 kernel configurations with respect to Backsteps (M) or Degrees of Freedom (ν), this is also the same configuration as our applied case presented in Section 3. We can see in Fig. 3 that the results do indeed confirm our intuition that the χ^2 kernel, regardless of the configuration, equals or improves the $\bar{k}$ values. This trend implies that the χ^2 kernel improves the rate in which the x_2 population become precise. Investigating Table 2 we see that this trend applies to the larger data set. Interestingly the improvement of $\bar{k}$ is not as straight forward as to imply more degrees of freedom always performs better, we see in fact that there is a deeper seemingly inverse relationship between the number of Backsteps M required and the quantity of measurement noise, which should be further explored in the future. For now our empirical results suggests for the applied case a $M = 10$ should be adequate. Irrespective of this deeper relationship between the χ^2 kernel's degree's of freedom and quantity of noise, we have strong evidence suggesting a blanket improvement of the AE precision with the χ^2 kernel.

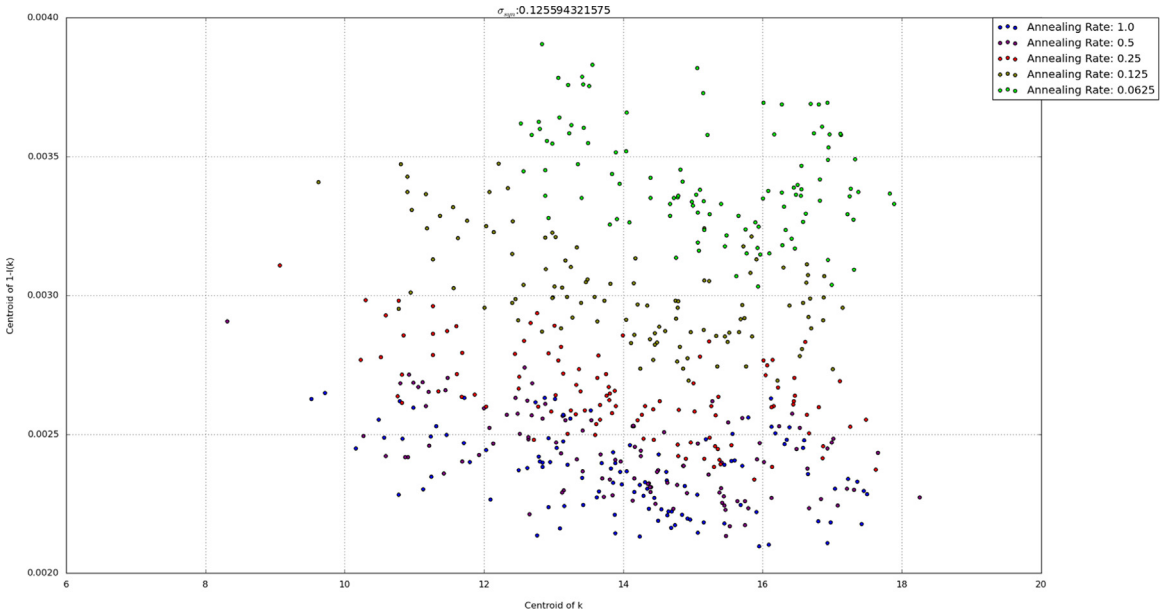

Fig. 2. Mean centroids when $\sigma_{syn} \approx 0.125$, which are coloured according to their to annealing rate ($2^{-\xi}$).

Table 1
Minimum and maximum centroid values sorted by annealing rate (ξ).

Sigma	Annealing rate $2^{-\xi}$	$A=0.994$			
		$\bar{I}_{min}$	$\bar{I}_{max}$	$\bar{k}_{min}$	$\bar{k}_{max}$
0.05	0.0625	2.53×10^{-3}	3.18×10^{-3}	12.05	14.88
	0.125	2.29×10^{-03}	2.66×10^{-3}	10.23	13.97
	0.25	2.00×10^{-03}	2.41×10^{-3}	9.39	12.57
	0.5	1.85×10^{-03}	2.13×10^{-3}	9.14	13.63
	1	1.75×10^{-03}	2.20×10^{-3}	8.12	12.85
0.125	0.0625	3.25×10^{-3}	3.53×10^{-3}	15.66	16.94
	0.125	2.83×10^{-03}	2.98×10^{-3}	14.11	16.58
	0.25	2.55×10^{-03}	2.98×10^{-3}	13.45	16.04
	0.5	2.27×10^{-03}	2.54×10^{-3}	13.72	15.50
	1	2.22×10^{-03}	2.39×10^{-3}	13.15	15.71
0.5	0.0625	4.47×10^{-3}	5.15×10^{-3}	15.14	18.12
	0.125	3.75×10^{-03}	4.27×10^{-3}	14.67	17.91
	0.25	3.18×10^{-03}	3.71×10^{-3}	14.95	17.81
	0.5	3.02×10^{-03}	3.38×10^{-3}	15.22	17.64
	1	2.99×10^{-03}	3.36×10^{-3}	14.92	17.97

6. Application to the TBN data set

In order to predict the oil degradation, recall our motivation in Section 3, we must first introduce indicators to predict. Specifically we require an indicator that transforms a prediction with respect to state into a prediction with respect to time. A time based prediction is more applicable to an applied case than a state prediction but usually involves a transformation that is often intractable. We also wish to provide further evidence of an improvement on the existing state of the art Particle Filtering based Prognosis utilising the SIR-PF than was seen in Section 5. We will compare our performance of these prediction indicators on the set of real total base number (TBN) of oil data seen in Fig. 1 presented in Section 3.

6.1. The Just in Time Point (JITP)

Firstly we consider a key indicator in prognosis, namely the Just in Time Point [25] (*JITP*). In order to define the *JITP* we first consider a limit, in this case a TBN limit, which we refer to as the Safe Limit [4]. This is an approximation of the optimum TBN level that the oil should be changed at, this limit in more critical system would be considered a failure

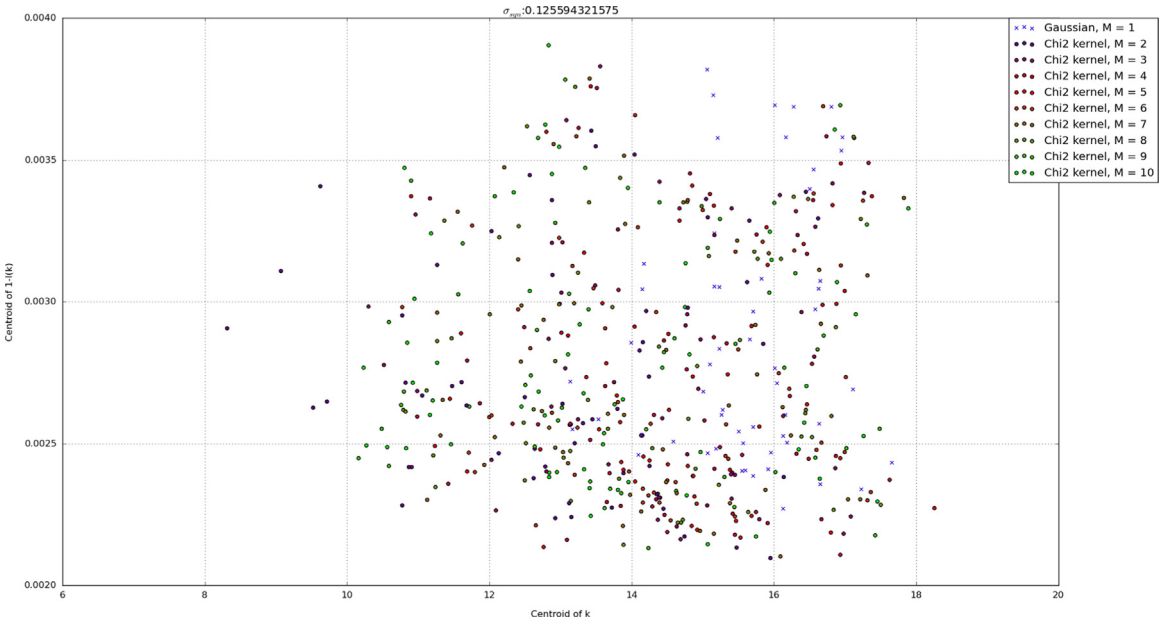

Fig. 3. Mean centroids when $\sigma_{syn} \approx 0.125$, coloured with respect to Backsteps (M).

Table 2
Minimum and maximum centroid values sorted by Gaussian and χ^2 kernel configurations.

Sigma	Backsteps	$A=0.994$			
		I_{min}	I_{max}	k_{min}	k_{max}
0.05	Gaussian kernel	2.13×10^{-3}	3.18×10^{-3}	12.47	14.87
	χ^2 kernel, $M=2$	1.92×10^{-03}	2.72×10^{-3}	10.02	14.18
	χ^2 kernel, $M=3$	1.75×10^{-03}	2.61×10^{-3}	11.16	13.88
	χ^2 kernel, $M=5$	1.92×10^{-03}	2.61×10^{-3}	10.11	13.19
	χ^2 kernel, $M=7$	2.00×10^{-03}	2.53×10^{-3}	9.07	12.76
	χ^2 kernel, $M=9$	1.99×10^{-03}	2.56×10^{-3}	8.98	12.33
	χ^2 kernel, $M=10$	1.96×10^{-03}	2.60×10^{-3}	8.12	12.04
0.125	Gaussian kernel	2.39×10^{-3}	3.53×10^{-3}	15.50	16.94
	χ^2 kernel, $M=2$	2.24×10^{-03}	3.29×10^{-3}	13.15	15.66
	χ^2 kernel, $M=3$	2.23×10^{-03}	3.38×10^{-3}	14.37	16.08
	χ^2 kernel, $M=5$	2.31×10^{-03}	3.34×10^{-3}	14.17	16.82
	χ^2 kernel, $M=7$	2.32×10^{-03}	3.36×10^{-3}	13.98	16.48
	χ^2 kernel, $M=9$	2.32×10^{-03}	3.35×10^{-3}	13.86	16.01
	χ^2 kernel, $M=10$	2.32×10^{-03}	3.25×10^{-3}	13.88	15.94
0.5	Gaussian kernel	2.99×10^{-3}	4.47×10^{-3}	17.45	18.12
	χ^2 kernel, $M=2$	3.02×10^{-03}	4.84×10^{-3}	14.67	15.22
	χ^2 kernel, $M=3$	3.14×10^{-03}	4.87×10^{-3}	15.30	15.93
	χ^2 kernel, $M=5$	3.11×10^{-03}	5.01×10^{-3}	16.63	17.31
	χ^2 kernel, $M=7$	3.32×10^{-03}	4.90×10^{-3}	17.10	17.77
	χ^2 kernel, $M=9$	3.33×10^{-03}	5.04×10^{-3}	16.97	17.50
	χ^2 kernel, $M=10$	3.26×10^{-03}	5.15×10^{-3}	16.96	17.76

threshold or in general a critical point in the system evolution.

For the purposes of the set of data analysed in this section, we have simply selected a limit (*Safe Limit*=5.6) that is applicable to the entire set. If applied in operation, a safe limit for TBN measurements would be based on existing expert knowledge, such as information provided by equipment manufacturers, or given a large enough data set, statistics.

Conditional on the selected limit we can define a probability density function (pdf) that the state has degraded past the defined limit which we denote the End of Life [25] (*EOL*) pdf. Where

$$Pr\{EOL = eol\} = \int_{-\infty}^{Safe\ Limit} f_{Y(eol)}(y(eol))d(y(eol)) \tag{34}$$

and as in [25] the expected *EOL* and the *JITP* is given by

$$\widehat{EOL} \triangleq E\{k|E\{y(k)\} = Safe\ Limit\} \tag{35}$$

$$JITP_{\beta\%} = \underset{k}{\operatorname{argmin}}(Pr\{EOL \leq eol\} \geq \beta\%) \tag{36}$$

The *EOL* expectation $\hat{EOL}$ is interpreted as the most likely time a given system will pass the safe limit. In contrast the $JITP_{\beta\%}$ considers the tail of the *EOL* pdf and gives a likely time of reaching a certain percentage $\beta\%$ likelihood of the *EOL* pdf. For example given a $\beta\%=5\%$ implies that the $JITP_{5\%}$ will be the point two standard deviations before the expectation of the $\hat{EOL}$ if we assume the *EOL* pdf is a Normal distribution. *JITP* incorporates the concept of acceptable risk in the measure.

We consider the $JITP_{5\%}$ (where $\beta\%=5\%$) of 60 realisations of the Particle Filtering based Prognosis (PFP) for both the *Chi-Squared weighted PFP* and the *Normally weighted PFP*, where

- *Chi-Squared weighted PFP*; refers to an estimation stage that utilises the *Sequential Importance Sampling/Re-sampling Particle Filter* (SIR-PF) presented in Section 2.3 using Chi-Squared χ^2 weighted smoothing window (25) and (26) as the likelihood function and a prediction stage that utilises the PFP presented in Section 2.4
- *Normally weighted PFP*; refers to an estimation stage that utilises the *Sequential Importance Sampling/Re-sampling Particle Filter* (SIR-PF) presented in Section 2.3 using the standard Normally weighted (22) likelihood function and a prediction stage that utilises the PFP presented in Section 2.4

As both forms of the PFP algorithms are stochastic processes this implies that the $JITP_{5\%}$ are random numbers, so in order to make statistically valid comparison we utilise a Monte Carlo process to effectively sample the pdf of the $JITP_{5\%}$ for both processes. We perform 60 realisations of the $JITP_{5\%}$ estimation for both the *Chi-Squared weighted PFP* & *Normally weighted PFP* for every (complete) set available, which totals 14 distinct sets. Fig. 4 demonstrates an example of the pdf of the $JITP_{5\%}$ for the 60 realisations superimposed on a single realisation of the prognosis (for demonstration) for both the *Chi-Squared weighted PFP* & *Normally weighted PFP* of a single set (Truck 1: Set 1).

In Fig. 4 the prediction point is fixed in time for visualisation purposes, all realisations of *JITP* calculations are computed at every point up to the Safe Limit. The state prediction behaviour seen in Fig. 4 is common for all of this remaining section, it is only the transformation to the time domain that varies.

Fixing an arbitrary prediction point such as Fig. 4 it is difficult to visually distinguish which algorithm is better. One fixed point realisation is also an oversimplification of the results obtained as this a multidimensional, time varying problem. We require a metric that can reduce the dimensionality of the problem, account for the entire time evolution of the predictions, as well as encapsulates a measure of the precision of the prediction.

6.2. Remaining useful life precision index

We consider the original implementation of the PI we introduced in Section 5, as it was implemented in [22]. The original application of the PI is a metric to compare the precision of a Confidence Interval (*CI*). This index considers the relative width of the 95% points of the confidence interval (CI_t), computed at time t, when compared to the expected time-of-failure (*ToF*)

$$I_1(t) = e^{-\left(\frac{\sup(CI_t)-\inf(CI_t)}{E_t\{ToF\}-t}\right)} \tag{37}$$

where $E_t\{ToF\}$ is the expectation of the time-of-failure, conditional to the measurements up to time t. The Precision Index (*PI*) quantifies the concept that the more data an algorithm has processed, the more precise the prognosis should be. From (37) we can see an index value of $I_1(t) \approx 1$ is considered a precise prognosis result.

From one realisation of the *PI*, as seen in Fig. 5, it is difficult to distinguish a clear improvement, this coupled with the fact that each calculation of the *EOL* pdf of both algorithms is simply one realisation of a stochastic process. We wish to understand the general behaviour of each algorithm and eliminate the possibilities of outliers. With respect to the previous considerations we define a modification to the precision index so as to highlight any improvement, we denote this as the Normalised Integral of Precision (NIP).

6.3. Normalised Integral of Precision (NIP)

The NIP should emphasise the aim of a better prognosis algorithm, which is to have an accurate prognosis result as quickly as possible. We can consider a *PI* where $I(t) \to 1$ when $t \ll ToF$ as the best possible result. With this in mind we define the Integral of the Precision

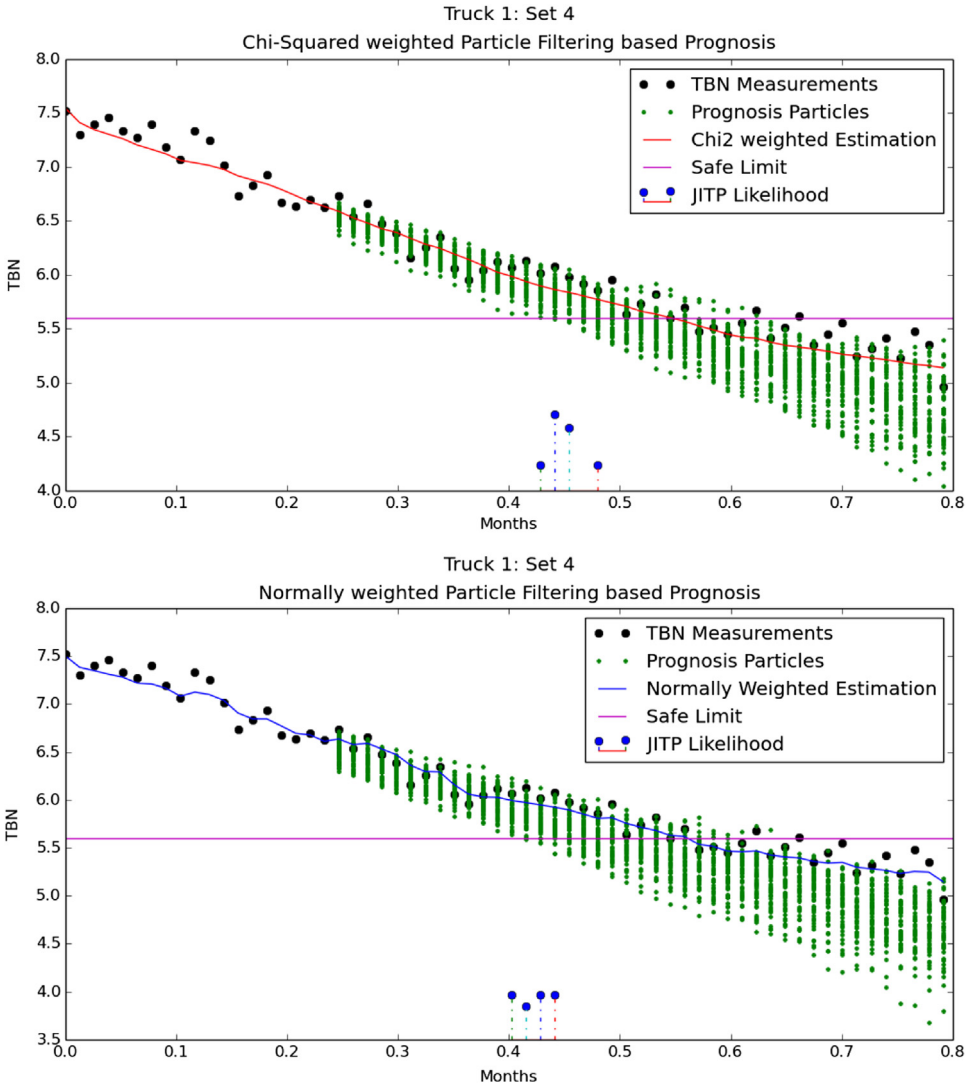

Fig. 4. Estimation and prognosis comparison showing Just in Time Point (JITP) statistics (60 realisations).

$$\hat{N}(\tau) = \int_0^{\tau} e^{-\left(\frac{\sup(CI_t)-\inf(CI_t)}{E_t\{ToF\}-t}\right)} dt, \tag{38}$$

where τ is the point in time in which the prognosis is performed and $\tau \leq ToF$. The Normalised Integral of Precision (*NIP*) is defined as:

$$N(\tau) = \frac{\hat{N}(\tau)}{\sup(\hat{N}(\tau))} \tag{39}$$

We can see in Fig. 6 that now it is much easier to distinguish an improvement in the precision of the prognosis, we could simply take the difference of the *NIP* for the χ^2 algorithm with the *NIP* of the Normally weighted algorithm, e.g.

$$\Delta(\tau) = N_{\chi^2}(\tau) - N_N(\tau) \tag{40}$$

where $N_{\chi^2}(\tau)$ is the *NIP* of the χ^2 Smoothed Particle Filter and $N_N(\tau)$ is the *NIP* of the Normally weighted SIR Particle Filter. $\Delta > 0$ implies the χ^2 Smoothed Particle Filter is an improvement over the Normally weighted SIR Particle Filter and visa versa. Given that each *NIP* is simply one realisation of a stochastic process, $N(\tau)$ is a random variable, which implies the difference of one realisation of two random variables, which does not have much statistical significance.

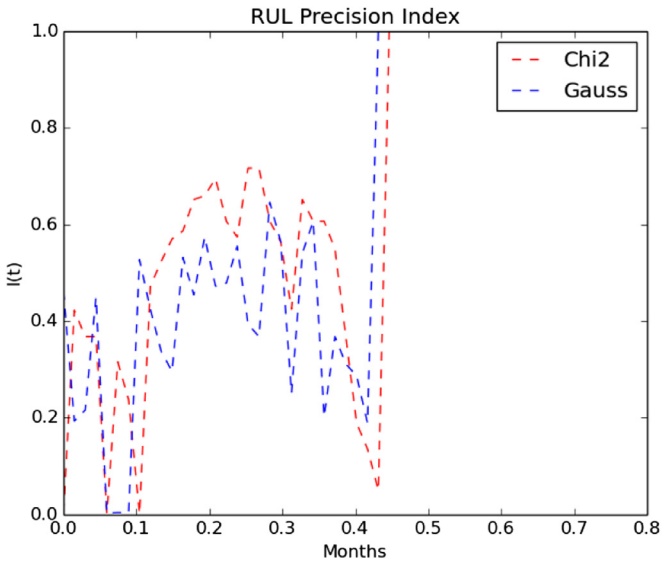

Fig. 5. Precision index of truck 1, Set 1. Note $t=0.45$ is the *ToF*, after which the PI has no significance.

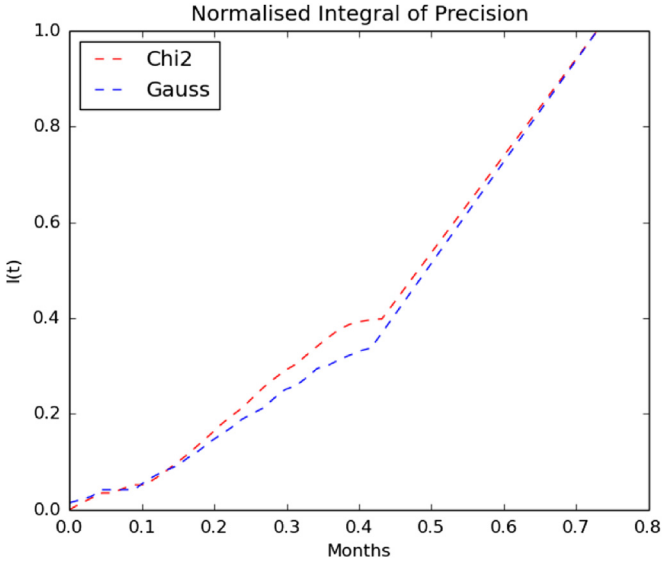

Fig. 6. Normalised integral of the precision index with respect to time for truck 1, set 1. Note $t=0.45$ is the *ToF* after which, the NIP has no significance.

6.4. Monte Carlo sampling and convolution of NIP

Considering $N(\tau)$ as a time varying random variable we can consider each *NIP* as simply one realisation of that random variable. Therefore in order for $\Delta(\tau)$ to have statistical significance we will consider the convolution of N_{χ^2} and $-N_N$, defined as

$$\Delta(\tau) \equiv (N_{\chi^2} * (-N_N))(\tau). \tag{41}$$

$\Delta(\tau)$ is the solution of the joint pdf of the random variables N_{χ^2} and $-N_N$. One method of calculating the solution to the convolution would be to find an analytical solution to both the random variables then calculate the solution to the joint pdf. Using this method could possibly be mathematically intractable and it cannot be guaranteed that a solution could be found. An alternate method, and the one utilised, is a Monte Carlo method whereby the underlying pdf π_τ can approximated by sampling $N(\tau)$, n times, where the $\lim_{n\to\infty} N^n(\tau) = \pi_\tau$. From these discrete approximations of the pdf of both random variables, $\pi_{\chi^2,\tau}[n]$ and $\pi_{N,\tau}[n]$, we can calculate the discrete convolution [26]

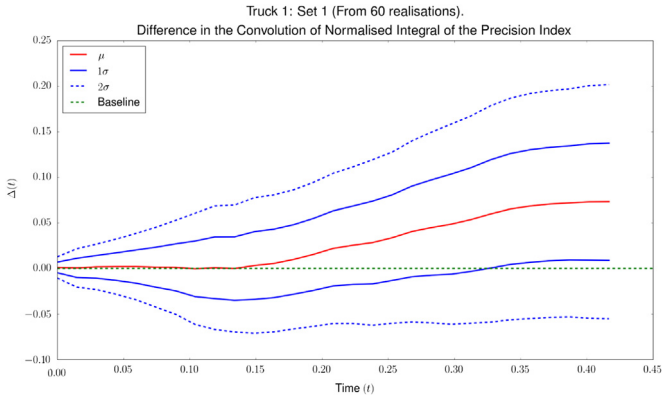

Fig. 7. Convolution of *NIP*.

$$\Delta(\tau) \triangleq (\pi_{\chi^2,\tau}{}^* - \pi_{N,\tau})[n] = \sum_{m=-\infty}^{\infty} \pi_{\chi^2,t}[m](-\pi_{N,t}[n-m]) \tag{42}$$

For ease of computation we chose to sample each distribution π_τ^n 60 times; i.e., $n=60$. The choice to use 60 simulations was one based on experience. As a general rule, if Monte Carlo simulations provide a coarse set of statistics you should increase the quantity of simulations. However, in the case of the results presented in this paper, only negligible differences could be observed in terms of the proposed performance measures when using more than 60 simulations. We then convolute the results and calculate the first and second moments of the distribution as seen in Fig. 7. For Fig. 7 we see that the convolution is only calculated until the *ToF* (for Truck 1, Set 1 *ToF*=0.43), this is the case for all sets. This is a direct result of the fact that the PI has no significance after it has reached the *ToF*. From Fig. 7 we can now say with confidence that statistically the *Chi-Squared weighted PFP* improves the precision of the prognosis. We define some features in order to tabulate the results for all sets. We firstly consider in Table 3:

- the total percentage that the mean of the resultant pdf $\Delta(\tau)$ is above the Baseline $(\Delta(\tau) > 0)$
- the percentage of the mean of the resultant pdf $\Delta(\tau)$ for each Quartile is above the Baseline. A Quartile is defined as the results within each quarter of the total time of the results e.g. the 1st Quartile is the first quarter of the results with respect to time. Each Quartile contains 25% of the total results for each convolution $\Delta(\tau)$

We consider in Tables 4 5 the maximum and minimum standard deviation σ for each quartile for $\Delta\tau$ of each set. We tabulate these results in order to demonstrate how likelihood evolves over time for each $\Delta(\tau)$. We consider the $\sigma(\Delta(\tau))$ as an indicator of the confidence we have as time progresses that the *Chi-Squared weighted PFP* is in fact more precise.

6.5. Discussion of results

Ultimately the aim was to show an improvement in the precision of the prognosis of the *Chi-Squared weighted PFP* so that

Table 3
Table of convolution results.

Truck: set	% μ Above baseline				
	Total (%)	1st **Quartile(%)**	2nd **Quartile (%)**	3rd **Quartile (%)**	4th **Quartile (%)**
Truck 1: Set 1	93.10	100.0	71.43	100.0	100.0
Truck 1: Set 2	86.05	40.00	100.0	100.0	100.0
Truck 1: Set 3	34.04	81.82	0.00	0.00	50.00
Truck 1: Set 4	90.91	63.64	100.0	100.0	100.0
Truck 1: Set 5	97.37	88.89	100.0	100.0	100.0
Truck 2: Set 1	78.26	20.00	80.00	100.0	100.0
Truck 2: Set 2	66.67	50.00	0.00	100.0	100.0
Truck 2: Set 3	42.86	85.71	0.00	0.00	85.71
Truck 2: Set 4	77.27	100.0	0.00	100.0	100.0
Truck 2: Set 5	52.50	20.00	0.00	90.00	100.0
Truck 3: Set 1	56.67	0.00	14.29	100.0	100.0
Truck 3: Set 2	86.67	45.45	100.0	100.0	100.0
Truck 3: Set 3	58.33	83.33	16.67	33.33	100.0
Truck 3: Set 4	69.67	77.78	0.00	100.0	100.0

Table 4
Minimum standard deviation (σ) for each Quartile.

Truck: Set	**min(σ) per Quartile**			
	1st **Quartile**	2nd **Quartile**	3rd **Quartile**	4th **Quartile**
Truck 1: Set 1	5.84×10^{-3}	3.05×10^{-2}	4.12×10^{-2}	5.69×10^{-2}
Truck 1: Set 2	3.29×10^{-3}	3.80×10^{-2}	7.12×10^{-2}	8.28×10^{-2}
Truck 1: Set 3	9.05×10^{-3}	5.00×10^{-2}	7.99×10^{-2}	6.91×10^{-2}
Truck 1: Set 4	5.75×10^{-3}	4.06×10^{-2}	8.44×10^{-2}	9.20×10^{-2}
Truck 1: Set 5	1.10×10^{-2}	4.65×10^{-2}	9.32×10^{-2}	9.45×10^{-2}
Truck 2: Set 1	3.04×10^{-3}	1.05×10^{-2}	2.35×10^{-2}	2.87×10^{-2}
Truck 2: Set 2	9.70×10^{-3}	3.45×10^{-2}	5.95×10^{-2}	7.87×10^{-2}
Truck 2: Set 3	9.99×10^{-3}	3.05×10^{-2}	6.52×10^{-2}	5.69×10^{-2}
Truck 2: Set 4	7.99×10^{-3}	2.40×10^{-2}	3.91×10^{-2}	4.55×10^{-2}
Truck 2: Set 5	6.54×10^{-3}	5.72×10^{-2}	8.66×10^{-2}	7.22×10^{-2}
Truck 3: Set 1	1.35×10^{-2}	4.64×10^{-2}	7.65×10^{-2}	9.23×10^{-2}
Truck 3: Set 2	1.82×10^{-2}	7.96×10^{-2}	1.67×10^{-1}	8.35×10^{-2}
Truck 3: Set 3	3.25×10^{-3}	1.15×10^{-2}	2.17×10^{-2}	2.72×10^{-2}
Truck 3: Set 4	5.47×10^{-3}	2.49×10^{-2}	5.17×10^{-2}	7.63×10^{-2}

Table 5
Maximum standard deviation (σ) for each Quartile.

Truck: Set	**max(σ) per Quartile**			
	1st **Quartile**	2nd **Quartile**	3rd **Quartile**	4th **Quartile**
Truck 1: Set 1	2.57×10^{-2}	3.95×10^{-2}	5.49×10^{-2}	6.43×10^{-2}
Truck 1: Set 2	3.36×10^{-2}	6.86×10^{-2}	9.74×10^{-2}	9.58×10^{-2}
Truck 1: Set 3	4.41×10^{-2}	9.12×10^{-2}	9.41×10^{-2}	8.98×10^{-2}
Truck 1: Set 4	3.53×10^{-2}	8.03×10^{-2}	9.51×10^{-2}	1.01×10^{-1}
Truck 1: Set 5	4.28×10^{-2}	9.00×10^{-2}	1.01×10^{-1}	1.03×10^{-1}
Truck 2: Set 1	8.48×10^{-3}	2.13×10^{-2}	2.76×10^{-2}	3.47×10^{-2}
Truck 2: Set 2	2.98×10^{-2}	5.55×10^{-2}	7.51×10^{-2}	8.52×10^{-2}
Truck 2: Set 3	2.75×10^{-2}	6.00×10^{-2}	7.68×10^{-2}	8.41×10^{-2}
Truck 2: Set 4	2.15×10^{-2}	3.94×10^{-2}	4.31×10^{-2}	5.60×10^{-2}
Truck 2: Set 5	4.97×10^{-2}	9.66×10^{-2}	9.57×10^{-2}	9.64×10^{-2}
Truck 3: Set 1	4.20×10^{-2}	7.24×10^{-2}	9.23×10^{-2}	9.67×10^{-2}
Truck 3: Set 2	6.84×10^{-2}	1.58×10^{-1}	2.03×10^{-1}	1.65×10^{-1}
Truck 3: Set 3	1.08×10^{-2}	2.00×10^{-2}	2.79×10^{-2}	3.20×10^{-2}
Truck 3: Set 4	2.22×10^{-2}	4.80×10^{-2}	7.51×10^{-2}	8.32×10^{-2}

a case could be made for its use over the classical *Normally weighted PFP*. We expected an improvement when applied to a real world application, in this case the prediction of the TBN oil degradation, which was shown in Fig. 1 and also motivated in Section 3. Two aspects of the algorithm where investigated in order to demonstrate an improvement of the *Chi-Squared weighted PFP*.

The first aspect of the algorithm investigated was the fact that χ^2 kernel improves the rate in which the AE becomes precise. This improvement is demonstrated in Fig. 3 and Table 2 as the centroid with respect to time $\bar{k}$, is closer to the origin for all χ^2 cases. We see in Table 2 the most significant results:

- The χ^2 kernel improves the centroid when $\sigma_{syn} = 0.05$ from $\bar{k}_{min} = 12.47$ for the Gaussian Kernel to $\bar{k}_{min} = 8.12$ for a χ^2 kernel with $M=10$. We can also see an improvement in this case of the centroid from $\bar{I}_{min} = 2.13 \times 10^{-3}$ for the Gaussian Kernel to $\bar{I}_{min} = 1.96 \times 10^{-3}$ for the χ^2 kernel with $M=10$.
- When $\sigma_{syn} = 0.125$ we have an improvement of the centroid from $\bar{k}_{min} = 15.50$ for the Gaussian Kernel to $\bar{k}_{min} = 13.15$ for a χ^2 kernel with $M=2$.
- When $\sigma_{syn} = 0.5$ we have an improvement of the centroid from $\bar{k}_{min} = 17.45$ for the Gaussian Kernel to $\bar{k}_{min} = 14.67$ for a χ^2 kernel with $M=2$.

There was an improvement for all configurations of the χ^2 kernel, although some configurations performed better than others, depending on measurement characteristics such as σ_{syn}. We assert that this strong evidence that the χ^2 kernel improves the performance of AE.

The second aspect investigated was an improvement on the precision of the predicted *EOL* pdf. In Section 6.3 we presented a measure that demonstrates an improvement of *Chi-Squared weighted PFP*'s over *Normally weighted PFP* with respect to the precision of the predicted *EOL* pdf. From the *NIP* results from Tables 3–5 we can conclude that:

1. 70.59% of sets are above the baseline (on average) for more than 50% of the time
2. 1st Quartile 52.94% of sets are ≥50%
3. 2nd Quartile 35.29% of sets are ≥50%
4. 3rd Quartile 64.70% of sets are ≥50%
5. 4th Quartile 100.0% of sets are ≥50%

Given the definition of *NIP* in Section 6.3 we assert that from our results we have provided strong evidence that the *Chi-Squared weighted PFP*'s prediction is more precise and therefore an improvement on the classical *Normally weighted PFP* when used for prediction.

Given the large volume of statistics generated we can conclude with some certainty that we have shown that the χ^2 kernel has improved the performance of both SIR-PF and PFP.

7. Conclusion

This paper presents an adaptive Particle-Filtering based Prognosis (PFP) with a smoothing window and a Chi-squared weighted likelihood function, which aims at improving the convergence of a parameter learning framework called Artificial Evolution (AE), as well as improving the precision of the prognosis results. Strong evidence is shown that the framework presented in this paper is an improvement of the classical PFP which uses a Normally weighted likelihood function.

Statistical evidence was provided as proof that the new framework is in fact an improvement. Both algorithms were applied to a series of real world data which involves total base number measurements from three mining haul trucks as well as a series of synthetically generated data with varying uncertainty and parametric configurations. The use of synthetically generated data demonstrated that the χ^2 kernel improved the performance of AE in the estimation phase. With the real data sets, prognosis performance was compare as a measure of both algorithms in parallel. These algorithms were applied to a novel measure we call the Normalised Integral of Precision (NIP). 60 separate realisations were performed of each measure, in order to generate statistics of the results in a *Monte Carlo* fashion. The need to generate various realisations of each measure was due to the stochastic nature of the indices.

One important outcome of the introduction of the χ^2 kernel, mentioned in Section 5.3.2, is the indication of a deeper seemingly inverse relationship between the number of backsteps M used in (25) and the measurement noise ϕ in (20). M is indeed a key variable in the performance of the Chi-squared weighted Particle Filter and a solution to the optimum M (or an optimum relationship with ϕ) is still an open issue that can be explored in future work.

From our results we claim that we have shown that statistically the Chi-squared Smoothed Adaptive Particle-Filtering based Prognosis has improved the convergence of the AE framework and the precision of the prognosis results.

The application of the generalised model framework with AE was used to model a series of data that did not have a concrete phenomenological model.We demonstrated that the Chi-squared Smoothed Adaptive Particle-Filtering assisted in the convergence of the parameter that was required to be learned by the AE framework in a first order system. Theoretically the idea could be expanded to encompass a system of higher order generalised model with multiple parameters to be learnt and we propose to explore this idea in future work.

References

[1] C.S. Byington, M.J. Roemer, T. Galie, Prognostic enhancements to diagnostic systems for improved condition-based maintenance [military aircraft], in: Aerospace Conference Proceedings, 2002, IEEE, vol. 6, IEEE, 2002, pp. 6–2815.
[2] B. Saha, K. Goebel, S. Poll, J. Christophersen, Prognostics methods for battery health monitoring using a bayesian framework, Instrum. Meas. IEEE Trans. 58 (2) (2009) 291–296.
[3] A. Doucet, N. De Freitas, N. Gordon, Sequential Monte Carlo Methods in Practice, Springer, 2001.
[4] M.E. Orchard, G.J. Vachtsevanos, A particle-filtering approach for on-line fault diagnosis and failure prognosis, Transactions of the Institute of Measurement and Control.
[5] C. Andrieu, A. Doucet, E. Punskaya, Sequential monte carlo methods for optimal filtering, in: Sequential Monte Carlo Methods in Practice, Springer, 2001, pp. 79–95.
[6] M. Dalal, J. Ma, D. He, Lithium-ion battery life prognostic health management system using particle filtering framework, Proc. Inst. Mech. Eng. O: J. Risk Reliab. 225 (1) (2011) 81–90.
[7] Z. Di, M. Yan, B. Qing-Wen, Estimation of lithium-ion battery state of charge, in: Control Conference (CCC), 2011 30th Chinese, IEEE, 2011, pp. 6256–6260.
[8] C. Musso, N. Oudjane, F. Le Gland, Improving regularised particle filters, in: Sequential Monte Carlo Methods in Practice, Springer, 2001, pp. 247–271.
[9] M.E. Orchard, P. Hevia-Koch, B. Zhang, L. Tang, Risk measures for particle-filtering-based state-of-charge prognosis in lithium-ion batteries, Ind. Electron. IEEE Trans. 60 (11) (2013) 5260–5269.
[10] D. Pola, H.F. Navarrete, M.E. Orchard, R.S. Rabie, M. Cerda, B.E. Olivares, J.F. Silva, P. Espinoza, A. Pérez, et al., Particle-filtering-based discharge time prognosis for lithium-ion batteries with a statistical characterization of use profiles.
[11] B. Saha, K. 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, 2009, pp. 2909–2924.
[12] T.M. Cover, J.A. Thomas, Entropy, relative entropy and mutual information, Elem. Inf. Theory (1991) 12–49.
[13] M.E. Orchard, M.S. Lacalle, B.E. Olivares, J.F. Silva, R. Palma-Behnke, P.A. Estévez, B. Severino, W. Calderon-Munoz, M. Cortes-Carmona, Information-theoretic measures and sequential monte carlo methods for detection of regeneration phenomena in the degradation of lithium-ion battery cells.
[14] J. Kennedy, J.F. Kennedy, R.C. Eberhart, Y. Shi, Swarm Intelligence, Morgan Kaufmann, 2001.
[15] T. Li, S. Sun, T.P. Sattar, J.M. Corchado, Fight sample degeneracy and impoverishment in particle filters: a review of intelligent approaches, Expert Syst. Appl. 41 (8) (2014) 3944–3954.
[16] C.S. Byington, Introduction to improved real-time mechanical system diagnostics and prognostics, Impact Technologies for Tutorials Session: PHM Society (September 2012).
[17] J. Liu, M. West, Combined parameter and state estimation in simulation-based filtering, in: Sequential Monte Carlo Methods in Practice, Springer, 2001, pp. 197–223.
[18] N.J. Gordon, D.J. Salmond, A.F. Smith, Novel approach to nonlinear/non-gaussian bayesian state estimation, in: IEE Proceedings F (Radar and Signal Processing), vol. 140, IET, 1993, pp. 107–113.
[19] D.S. Moore, M.C. Spruill, Unified large-sample theory of general chi-squared statistics for tests of fit, Ann. Stat. (1975) 599–616.
[20] E.T. Jaynes, Probability Theory: the Logic of Science, Cambridge University Press, 2003.
[21] R.L. Plackett, Karl pearson and the chi-squared test, International Statistical Review/Revue Internationale de Statistique (1983) 59–72.
[22] M.E. Orchard, L. Tang, K. Goebel, G. Vachtsevanos, A novel rspf approach to prediction of high-risk, low-probability failure events, in: Annual Conference of the Prognostics and Health Management Society (PHM09), San Diego, CA, 2009.
[23] A. Saxena, J. Celaya, B. Saha, S. Saha, K. Goebel, Evaluating algorithm performance metrics tailored for prognostics, in: Aerospace Conference, 2009 IEEE, IEEE, 2009, pp. 1–13.
[24] M.H. Protter, C.B. Morrey, College Calculus with Analytic Geometry, Addison-Wesley, 1977.
[25] B.E. Olivares, C. Munoz, M.E. Orchard, J.F. Silva, Particle-filtering-based prognosis framework for energy storage devices with a statistical characterization of state-of-health regeneration phenomena, Instrum. Meas. IEEE Trans. 62 (2) (2013) 364–376.
[26] S.B. Damelin, W. Miller Jr, The Mathematics of Signal Processing, vol. 48, Cambridge University Press, 2011.