Improved Likelihood Estimation for Noisy Gamma Degradation Processes via Sequential Monte Carlo

Merel Buist[1], Radislav Vaisman[2], Maria Vlasiou*[3]

[1]*Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, Eindhoven, 5600 MB, the Netherlands.*
[2]*Faculty of Science, The University of Queensland, Brisbane, St Lucia 4072, Queensland, Australia.*
[3]*Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, P.O. Box 217, Enschede, 7500 AE, the Netherlands.*

May 15, 2024

Abstract

The use of Gamma processes for modeling various degradation phenomena has recently gained extensive attention. In many cases, the degradation data contain measurement errors and an intractable likelihood phenomenon comes into sight. Therefore, in order to perform efficient statistical inference, one must obtain high-quality estimates of the corresponding likelihood. Our findings indicate that the crude Monte Carlo method, which is the de facto state-of-the-art method, is not adequate in practice for efficient likelihood estimation. To cope with this problem, we propose to employ the sequential Monte Carlo approach, which shows promise for improved reliability compared to the current state of the art. Our approach leads to efficient variance minimization and opens the way for effective and scalable inference procedures.

Keywords: sequential Monte Carlo; degradation process; likelihood estimation

*Corresponding author.
E-mail addresses: merelbuist@gmail.com (M. Buist), Email: r.vaisman@uq.edu.au (R. Vaisman), m.vlasiou@utwente.nl (M. Vlasiou)

1 Introduction

Many complex systems and products we use in our daily activities are subject to degradation. Degradation affects the product's lifetime, the availability of services, and the corresponding safety of usage. A few of the numerous examples of critical systems that undergo degradation over time are nature (e.g., fish populations [1]), the human body (e.g., humans organs [2]), medicine (e.g., vaccines [3]), and hardware (e.g., nuclear reactors [4]). Understanding and estimating these degradation processes as precisely as possible is of great importance, as this can lead to more preservation of nature, improved prediction of biological processes, and better maintenance of infrastructures.

This work aims to provide an important step forward in understanding and managing general degradation processes. Specifically, we concentrate on the problem of the analysis of degradation data. Such data usually contains measurement errors and as a consequence, reliability scientists and practitioners need to deal with intractable likelihood phenomena when performing statistical inference. In this paper, the objective is to compare the estimation of intractable likelihood functions using a particle-based method and a crude Monte Carlo method, assuming a model that incorporates Gamma-distributed degradation and normally-distributed errors. To evaluate the performance of these methods, we test them on both synthetic data and real drug shelf-life data. Our findings demonstrate that the particle-based method offers more reliable likelihood estimates compared to the crude Monte Carlo method. These improved estimates enable more accurate estimation of degradation development. The insights gained from this study contribute to advancing the field of degradation modeling and its practical applications.

Reliability management is a rich field of study (see for instance [5]), where system lifetime data is commonly studied. However, this method can be inefficient and prone to inaccuracies [6]. This is due to the fact that many systems of interest are extremely reliable (for example, nuclear reactors), and the time to failure can be prohibitively large, hence observing the degradation process in the data becomes hard.

In the last three decades, there has been increasing recognition in tackling reliability management in the context of degradation processes [7]. The analysis of degradation processes offers several advantages over lifetime data analysis. It provides more informative insights into highly reliable components with long lifetimes, allowing the assessment of crucial properties like degradation rate and remaining useful life without waiting for actual failure events. Additionally, the analysis of degradation data eliminates the need for complex and expensive accelerated life testing experiments. Moreover, in cases where the available data is censored, degradation analysis becomes crucial for assessing system reliability [8]. Due to the advantages of degradation data analysis over lifetime data analysis, our study aims to analyze degradation processes.

Degradation processes have been mainly studied through two different classes of models: general path models and stochastic models [9]. General path models are simplified models that assume that systems are tested under a homogeneous environment. Examples of general path models are the modeling of drug shelf-life in [10], which models degradation as a linear regression model, and modeling structural deficiencies in buildings through general nonlinear mixed-effects models [11]. Stochastic models, on the other hand, allow for unpredictability in the environmental impact and hence randomness in degradation [9]. Examples of stochastic models are, for instance, Wiener processes used to model the gyroscopic drift in a gyros used for an inertial navigation system [12], and Gamma processes used to model corrosion in a nuclear reactor [4].

Stochastic processes, and in particular Gamma processes, were first introduced to the domain of system reliability in 1975 [13, 14]. Gamma processes were found to be beneficial in recommending

appropriate maintenance and inspection choices. Consequently, it was shown that these procedures were suitable for characterising a range of deterioration scenarios [15, 16]. There are many applications of Gamma processes in reliability literature. For instance, the optimal dike heightening and the best sand nutrition sizes were determined using Gamma processes, see [17, 18]. While there are numerous applications of Gamma processes, [19, 20, 21, 22], for some more recent examples, we refer to remaining useful life prediction [23], wheel wear modeling [24], and space-time-dependent reliability analysis [25].

Our paper models degradation using a Gamma process, which is particularly suitable for systems that show degradation through fatigue. The Gamma process is chosen for its intuitive properties, such as modelling monotone and gradual degradation [26].

Given a model at hand, estimating the parameters of the model is typically done through maximum likelihood estimation. This method requires the computation of a likelihood function. In practice, the computation of the likelihood function is intractable; the method of choice is to estimate the likelihood values numerically. *Crude Monte Carlo* (CMC) is a common method of estimating such an intractable likelihood function. *Sequential Monte Carlo* (SMC), also known as *particle filtering*, was proposed as an alternative in [27], where the authors intended to combat the challenges of non-linearity and non-Gaussian noise in the corresponding model.

In this paper, we compare the performance of CMC to SMC on the likelihood estimation of a parameter set given the observed data and a Gamma degradation process with noisy data. We show both theoretically and numerically that SMC outperforms CMC. Theoretical evidence is provided by establishing the failure of CMC in estimating the likelihood in the asymptotic regime, as proven in Theorem 1. Numerical evidence is provided through experiments on both synthetic and real data. The results are of great theoretical and practical importance since today various estimation algorithms rely on CMC for the corresponding likelihood function estimation. To the best of our knowledge, SMC and CMC have not explicitly been compared for likelihood estimation. However, SMC has shown significantly smaller variances for likelihood estimates compared to other methods in for instance [28]. The results in [29] also demonstrates SMC to have multiple advantages, such as variance reduction and efficiency in long time series. Moreover, SMC and CMC have been explicitly compared for other subjects, for instance for option pricing in [30], which showed a significant reduction in standard deviation for SMC compared to CMC.

The remainder of the paper is organized as follows. In Section 2, we give a brief overview of the degradation problem with measurement errors and the model applied. Section 3 presents the methods applied in this paper. Specifically, we discuss the limitations of Crude Monte Carlo and introduce the Sequential Monte Carlo approach as a viable solution to overcome these limitations. In Section 4, we compare the performance of these algorithms and report our experimental findings. Section 5 compares the methods for a specific case study. Finally, Section 6 summarizes our findings and discusses possible directions for future research.

2 Problem Definition

Following Hazra et al. [31], the degradation model under consideration relies on the Gamma process. Specifically, we consider a system with C components that are subject to degradation. The corresponding degradation of each component $i \in \{1, \ldots, C\}$ is measured at specific times t_j, where $t_j \in \{t_1, \ldots, t_M\}$. By convention we assume that $t_0 = 0$. Each component represents an independent realization of a stochastic Gamma process, such as the degradation of C distinct batteries. As exact

measurements usually require a destruction or dismantling of the system, the observed data contains measurement errors.

To provide a more comprehensive understanding of the model, we introduce a set of random variables that capture the various influences within the degradation process. These variables collectively form the model and shed light on its underlying dynamics. First, we incorporate a term, $Z_i(t_j)$, to account for the measurement error specific to each component i and time t_j. We set $Z_i(t_0) = 0$ for all $i \in \{1, \ldots, C\}$, since we assume there is no measurement at time 0. Second, each component, representing a replication of the system, starts with an initial degradation denoted by A_i. For instance, when examining a battery, it might already have a degradation of 2 percent, resulting in a maximum charging capacity of 98 percent of the original battery life. Finally, for each component there is the degradation that occurs over time, denoted by $X_i(t_j)$. For example, a battery deteriorates over time, the amount it has deteriorated from time zero to time t_j equals $X_i(t_j)$. We take $X_i(t_0) = 0$ for all $i \in \{1, \ldots, C\}$. That is, we assume that there is no incremental degradation for new components. Together, these random variables constitute the complete degradation process:

$$Y_i(t_j) = A_i + X_i(t_j) + Z_i(t_j). \tag{1}$$

Note that, due to the previous assumptions at time 0 and Model Eq. (1), we have that $Y_i(t_0) = A_i$. This framework allows us to holistically investigate the degradation process and its associated components.

In order to characterize the newly introduced random variables, we assign probability distributions to them, based on the motivations discussed earlier in this paper. The initial degradation A_i is an independent and normally-distributed random variable with mean μ_A and variance σ_A^2 for all $i \in \{1, \ldots, C\}$. That is, $A_i \sim N(\mu_A, \sigma_A^2)$. The measurement error $Z_i(t_j)$ is independent of all A_i and all $Z_l(t_j)$ for all $l \neq i$. Also, $Z_i(t_j)$ is a normally-distributed random variable with mean 0 and variance σ_Z^2 for all $i \in \{1, \ldots, C\}$. That is, $Z_i(t_j) \sim N(0, \sigma_Z^2)$. Last, the degradation over time has Gamma-distributed independent increments. In other words,

$$X_i(t_j) - X_i(t_{j-1}) = \Delta X_i(t_j),$$

with $\Delta X_i(t_j)$ being a random variable independent of all A_i, all $Z_i(t_j)$, and all $\Delta X_i(t_j)$ for all $l \neq i$. Moreover, $\Delta X_i(t_j)$ is Gamma distributed with shape parameter $\alpha t_j^\eta - \alpha t_{j-1}^\eta$ and scale parameter β for all $i \in \{1, \ldots, C\}$. That is, $\Delta X_i(t_j) \sim \text{Gamma}(\alpha t_j^\eta - \alpha t_{j-1}^\eta, \beta)$, equivalent to [31]. Since the increments are Gamma distributed, the degradation over a positive time period is always positive, as the probability of $\Delta \mathbf{X}_i$ being smaller than or equal to 0 equals 0. In summary, this paper considers the following model:

Gamma Degradation Model.

$$\begin{aligned} Y_i(t_j) &= A_i + X_i(t_j) + Z_i(t_j), \\ A_i &\sim N(\mu_A, \sigma_A^2), \\ Z_i(t_j) &\sim N(0, \sigma_Z^2), \\ \Delta X_i(t_j) &\sim \text{Gamma}(\alpha t_j^\eta - \alpha t_{j-1}^\eta, \beta). \end{aligned} \tag{2}$$

All realized values of random variables will be denoted in lowercase, vectors and matrices will be represented in boldface. We capture all the parameters of this model by the vector:

$$\boldsymbol{\theta} = (\alpha, \eta, \beta, \mu_A, \sigma_A, \sigma_Z)^\top.$$

For a comprehensive overview of the notation used, please refer to Appendix A for further details.

The chosen shape and scale parameter of the Gamma distribution provide considerable flexibility within the model, allowing for a wide range of possibilities. Moreover, the setup of this model dictates that as the time difference increases, so does the shape parameter of the Gamma process. A higher shape parameter increases the probability of higher degradation rates. Intuitively, it is sensible that larger time differences lead to more significant degradation.

The selection of the parameters of the Gamma process plays a crucial role in determining the applicability of the model to different systems. Observe that the relationship between the variance and the mean is as follows: $Var(\Delta X_i(t_j)) = \mathbb{E}[\Delta X_i(t_j)] \cdot \beta$. As a result, this choice of shape and scale parameters captures well systems where the variance over time remains linear in terms of its mean. For systems where the variance over the degradation should increase more over time, one could introduce an appropriate time factor in the scale parameter.

The objective of this paper is to study methods for estimating the parameter vector $\boldsymbol{\theta}$. To perform parameter inference, it is essential to evaluate the likelihood function $\mathcal{L}(\boldsymbol{\theta}; \mathbf{y})$, which represents the likelihood of the observed data given the parameter $\boldsymbol{\theta}$. In the data collection process, degradation data $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_C)$, with $\mathbf{y}_i = (y_{i,t_1}, \ldots, y_{i,t_M})^\top$, is obtained for all components $i \in \{1, \ldots, C\}$. However, the presence of measurement errors necessitates incorporating not only the probability density function of the Gamma distribution but also the probability of the measurement error values in the likelihood function, resulting in its intractability; see Appendix B for details. This intractability introduces significant variability in the likelihood estimation, thereby complicating the task of finding the maximum likelihood. Unreliable likelihood estimates, characterized by high variance, can harm any reasonable optimization procedure. Consequently, it becomes crucial to employ an estimation method for the likelihood that reduces this variability. In this work, we focus on exploring two estimators of $\mathcal{L}(\boldsymbol{\theta}; \mathbf{y})$: CMC and SMC. Both estimators are employed using N_{part} simulations. We provide more information on CMC and SMC in Section 3.

3 Monte Carlo Methods

This section briefly describes the two Monte Carlo methods used in this paper: Crude Monte Carlo and Sequential Monte Carlo. We prove a significant limitation of CMC, which SMC circumvents. Furthermore, we elaborate on the particular SMC method we employ, known as the Bootstrap particle filter, and provide its specifics considering our Gamma Degradation Model (2).

3.1 Failure of the Crude Monte Carlo Method

In this section, we provide an explanation of how CMC operates and discuss its limitations when applied to our specific model. These limitations are further addressed and analyzed in Theorem 1.

CMC estimates an expected value by averaging over numerous runs. Therefore, we express the likelihood as an expected value. To do so, we begin by determining the likelihood per component, denoted by $\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}_i)$. This likelihood per component corresponds to the probability density function of the realized observed degradation $\mathbf{y}_i$ under the parameter set $\boldsymbol{\theta}$, denoted by $f_{\boldsymbol{\theta}}(\mathbf{y_i})$. This expression equals the expected value of the multiplication of the probability of the realized true

degradation during each time period $\Delta x_{i,t_j}$, under the distributions of the initial degradation A_i and the measurement errors $\mathbf{Z}_i$. In other words,

$$f_{\boldsymbol{\theta}}(\mathbf{y_i}) = \mathbb{E}_{A_i,\mathbf{Z}_i}\big[\prod_{j=1}^{M} f_{\boldsymbol{\theta}}(\Delta x_{i,t_j}|A_i = a_i, \mathbf{Z}_i = \mathbf{z}_i)\big].$$

Subsequently, the joint likelihood $\mathcal{L}(\boldsymbol{\theta};\mathbf{y})$ is obtained by multiplying all the likelihoods per component $\mathcal{L}(\boldsymbol{\theta};\mathbf{y}_i)$; that is:

$$\mathcal{L}(\boldsymbol{\theta};\mathbf{y}) = \prod_{i=1}^{C}\mathcal{L}(\boldsymbol{\theta};\mathbf{y}_i) = \mathbb{E}_{A_i,\mathbf{Z}_i}\Bigg[\prod_{i=1}^{C}\prod_{j=1}^{M} f_{\boldsymbol{\theta}}(\Delta x_{i,t_j}|A_i = a_i, \mathbf{Z}_i = \mathbf{z}_i)\Bigg].$$

By performing N_{part} simulations and sampling N_{part} times all initial degradations $\mathbf{a}$ and errors $\mathbf{z}$, we can compute the value in the expectation for each simulation, which yields the CMC likelihood estimator:

$$\hat{\mathcal{L}}(\boldsymbol{\theta};\mathbf{y}) = \frac{1}{N_{\text{part}}}\sum_{n=1}^{N_{\text{part}}}\prod_{i=1}^{C}\prod_{j=1}^{M} f_{\boldsymbol{\theta}}(\Delta x_{i,t_j}|A_i = a_i, \mathbf{Z}_i = \mathbf{z}_i). \tag{3}$$

For more details on the theoretical and estimated likelihood derivation, see Appendix B, which computes the specific likelihoods for Gamma Degradation Model (2) and shows that the estimator in Eq. (3) is unbiased.

In the remainder of this section, we prove that for Gamma Degradation Model (2) the CMC method fails to estimate likelihoods accurately in an asymptotic regime. We define a method as failing if it estimates the likelihood to be exactly zero, as it provides no more information than confirming that those particular parameters are not the true ones. In other words, we show that when the number of simulations N_{part} and the standard deviation of the measurement error $\sigma_Z \neq 0$ remain fixed, for the number of time measurements M or components C tending towards infinity, the probability of CMC failing approaches one. Furthermore, it is generally observed that a larger σ_Z leads to a quicker convergence of the probability of CMC failing towards one. We show this in Theorem 1.

In the case of CMC, failure to estimate the likelihood arises from the inherent methodology employed. CMC relies on sampling an initial degradation and sampling measurement errors. These sampled values are then used to compute the degradation over time. If, as a result of these samplings, the degradation increment turns out to be negative, the likelihood is evaluated to zero. The latter is due to the model assumption that degradation over positive time periods cannot be negative.

Theorem 1. *Given Gamma Degradation Model* (2)*, for a fixed number of simulations* N_{part} *and a fixed standard deviation of the measurement error* $\sigma_Z > 0$*, the probability of CMC failing to estimate the likelihood converges to* 1*, i.e.* $\mathbb{P}(\hat{\mathcal{L}}(\boldsymbol{\theta};\mathbf{y}) = 0) \to 1$*, for:*

1. *number of time measurements* $M \to \infty$*.*
2. *number of components* $C \to \infty$*,*

Proof. To prove this result, we show that the the likelihood estimator $\hat{\mathcal{L}}(\boldsymbol{\theta};\mathbf{y})$ goes to zero in probability for $M \to \infty$ or $C \to \infty$. To do so, we derive $\hat{\mathcal{L}}(\boldsymbol{\theta};\mathbf{y})$, which is a sum of terms. We show that these terms converge to zero in probability for Gamma Degradation Model (2), which implies $\hat{\mathcal{L}}(\boldsymbol{\theta};\mathbf{y})$

converges to zero in probability. The probability of the terms equalling zero requires knowledge of the distribution of $\Delta X_i(t_j)$. We lower bound this probability, and show that the lower bound converges to 1, which leads to the conclusion that the probability converges to 1 and hence $\hat{\mathcal{L}}(\boldsymbol{\theta};\mathbf{y})$ goes to zero in probability.

To this end, recalling that the true degradation over time $\Delta X_i(t_j)$ is Gamma distributed, Eq. (3) reduces to:

$$\hat{\mathcal{L}}(\boldsymbol{\theta};\mathbf{y}) = \frac{1}{N_{\text{part}}} \sum_{n=1}^{N_{\text{part}}} \prod_{i=1}^{C} \prod_{j=1}^{M} \frac{(\Delta x_{i,t_j}^n)^{(\alpha t_j^\eta - \alpha t_{j-1}^\eta - 1)} \exp(\frac{-\Delta x_{i,t_j}^n}{\beta}) \mathbb{1}\{\Delta x_{i,t_j}^n \geq 0\}}{\Gamma(\alpha t_j^\eta - \alpha t_{j-1}^\eta)\beta^{\alpha t_j^\eta - \alpha t_{j-1}^\eta}}. \tag{4}$$

The indicator function in the right-hand side of Eq. (4) causes the failure, i.e. the likelihood estimator $\hat{\mathcal{L}}(\boldsymbol{\theta};\mathbf{y})$ to be zero in certain cases. Namely, if for all $n \in \{1,\dots,N_{\text{part}}\}$ there exists an indicator function $\mathbb{1}\{\Delta x_{i,t_j}^n \geq 0\}$ that is zero, then the estimated likelihood value $\hat{\mathcal{L}}(\boldsymbol{\theta};\mathbf{y})$ is zero. Hence, we investigate the probability of there existing a component $i \in \{1,\dots,C\}$ and a time measurement $j \in \{1,\dots,M\}$ such that $\mathbb{1}\{\Delta X_i(t_j) \geq 0\}$ equals zero. This probability, which is the probability of a single term in the sum of Eq. (4) equalling zero, is written as follows:

$$\begin{aligned} \mathbb{P}(\exists_{i,j} \text{ s.t. } \Delta X_i(t_j) \leq 0) &= \mathbb{P}(\exists_{i,j} \text{ s.t. } \Delta X_i(t_j) \leq 0) = 1 - \mathbb{P}(\forall_{i,j} \text{ s.t. } \Delta X_i(t_j) \geq 0), \\ &= 1 - \prod_{i=1}^{C} \mathbb{P}(\forall_j \ \Delta X_i(t_j) \geq 0). \end{aligned} \tag{5}$$

To assess the probability in Eq. (5), it is crucial to determine the distribution of $\Delta X_i(t_j)$. We have observed the data $Y_i(t_j)$ for $i \in \{1,\dots,C\}$ and $j \in \{1,\dots,M\}$. At time zero ($j = 0$), the values are as follows: $Y_i(t_0) = A_i$, $X_i(t_0) = 0$, and $Z_i(t_0) = 0$. Consequently, we can discern two cases for the distribution of $\Delta X_i^n(t_j)$:

- $j = 1$: $\Delta X_i(t_1) = \Delta Y_i(t_1) - \Delta Z_{i,t_1} = y_{i,t_1} - A_i - Z_i(t_1) \sim \mathcal{N}(y_{i,t_1} - \mu_A, \sigma_A^2 + \sigma_Z^2)$,
- $j \geq 2$: $\Delta X_i(t_j) = \Delta y_{i,t_j} - \Delta Z_{i,t_j} \sim \mathcal{N}(\Delta y_{i,t_j}, 2\sigma_Z^2)$.

Using this knowledge on the distribution of $\Delta X_i(t_j)$, we can derive an upper bound of the probability $\mathbb{P}(\forall_j \ \Delta X_i(t_j) \geq 0)$, and hence a lower bound of the probability in Eq. (5). To do so, we define:

$$|\overline{\Delta Y}| = |max_{i,j}\{\Delta Y_i(t_j)\}|.$$

Using this maximum, the upper bound is derived as follows:

$$\mathbb{P}(\forall_j \ \Delta X_i(t_j) \geq 0) \overset{*}{\leq} \mathbb{P}\Big(\bigcap_{\frac{j}{2}=1}^{\frac{M}{2}} \Delta X_i(t_j) \geq 0\Big) = \prod_{\frac{j}{2}=1}^{\frac{M}{2}} \mathbb{P}(\Delta X_i(t_j) \geq 0) = \prod_{\frac{j}{2}=1}^{\frac{M}{2}} \Phi\left(\frac{\Delta Y_i(t_j)}{\sqrt{2}\sigma_Z}\right) \leq \Phi\left(\frac{|\overline{\Delta Y}|}{\sqrt{2}\sigma_Z}\right)^{\frac{M}{2}}.$$

In the above derivation, we upper bound the probability by considering only half of the events inside the probability, i.e. $\{\Delta X_i(t_j) \geq 0\}$ for all measurements j. This step is taken to remove the dependence between the events inside the probability. Namely, degradation increments $\Delta X_i(t_j)$ and $\Delta X_i(t_k)$ are independent if $k < j-1$ or $k > j+1$. Thus, at *, we specifically focus on cases where the degradation increments are two time periods apart, ensuring their independence.

The previous upper bound leads to a lower bound of the probability in Eq. (5):

$$\begin{aligned}\mathbb{P}(\exists_{i,j} \text{ s.t. } \Delta X_i(t_j) \leq 0) &= 1 - \prod_{i=1}^{C} \mathbb{P}(\forall_j \ \Delta X_i(t_j) \geq 0) \geq 1 - \prod_{i=1}^{C} \Phi\left(\frac{|\overline{\Delta Y}|}{\sqrt{2}\sigma_Z}\right)^{\frac{M}{2}}, \\ &= 1 - \Phi\left(\frac{|\overline{\Delta Y}|}{\sqrt{2}\sigma_Z}\right)^{\frac{C \cdot M}{2}}.\end{aligned} \tag{6}$$

One can easily see that the last expression in Eq. (6) converges to 1 for $\sigma_Z \neq 0$ fixed and either the number of components $C \to \infty$ or the number of time measurements $M \to \infty$. As the lower bound converges to 1, $\mathbb{P}(\exists_{i,j} \text{ s.t. } \Delta X_i(t_j) \leq 0) = \mathbb{P}(\exists_{i,j} \text{ s.t. } \mathbb{1}\{\Delta X_i(t_j) \geq 0\} = 0)$ converges to 1 as well. Therefore, the probability that the estimator equals 0, i.e. $\mathbb{P}(\hat{\boldsymbol{\mathcal{L}}}(\boldsymbol{\theta}; \mathbf{y}) = 0)$, converges to 1 for N_{part} fixed. □

Note also that one can induce CMC to fail, by letting $\sigma_Z \to \infty$, as then

$$\mathbb{P}(\exists_{i,j} \text{ s.t. } \Delta X_i(t_j) \leq 0) \xrightarrow{\sigma_Z \to \infty} 1 - \left(\frac{1}{2}\right)^{\frac{C \cdot M}{2}}.$$

Which approaches 1 for the data set reasonably big.

In order to avoid the failure in likelihood estimation, we propose using a different Monte Carlo algorithm, namely Sequential Monte Carlo. The SMC method is discussed in Section 3.2.

3.2 Sequential Monte Carlo

While CMC may fail due to the possibility of estimating negative true degradation over time $\Delta\mathbf{x}$, SMC is designed to prevent such breaches of model assumptions and subsequent failures. In CMC, each simulation involves sampling measurement errors $\mathbf{z}$ and computing the true degradation over time $\Delta\mathbf{x}$ using these errors. In contrast, SMC samples the true degradation over time $\Delta\mathbf{x}$, ensuring that this value is positive by construction and cannot violate the model assumptions. SMC samples $\Delta\mathbf{x}$ through the use of a proposal kernel M_{t_j}. The algorithm resamples between all runs N_{part} the occurred degradation until that time step through the use of weights $w_{t_j,n}$, computed by the use of a weight function G_{t_j}. Due to SMC's resilience to the described failure, next to its improved accuracy, we employ SMC in this paper as a superior alternative to CMC.

It is important to note that SMC can be computationally costly if resampling between simulations is performed at every time step. To address this, adaptive resampling was developed, where resampling is done only when the Effective Sample Size (ESS) drops below a threshold; see Chopin et al. [32] for details. The ESS, based on sample weights, reflects the quality of the particle set. In this paper, we employ a common threshold of $\frac{N_{\text{part}}}{2}$, ensuring resampling occurs when the number of effective samples decreases significantly.

For more detailed information on SMC and the ESS, please refer to [32, Ch. 10] and [32, Sect. 8.6] respectively. We have customized their generic SMC algorithm including adaptive resampling to suit our specific setup. The pseudocode for our customized algorithm can be found in Appendix C. The choice of distributions of the proposal kernel M_{t_j} and the weight function G_{t_j} depends on the specific particle filter in SMC being applied.

In this paper we apply the Bootstrap particle filter, which is a particle filter that is quite commonly used in Sequential Monte Carlo. For Gamma Degradation Model (2), the choice of the proposal kernel M_{t_j} and weight function G_{t_j} in this filter is as follows:

$$\begin{aligned} M_{t_0}(\Delta d_{i,t_0}) &= N(\mu_A, \sigma_A^2), \\ G_{t_0}(\mathbf{D}_n(t_0)) &= 1, \\ M_{t_j}(D_{i,k(n)}(t_{j-1}), \Delta d_{i,t_j}) &= \text{Gamma}(\alpha t_j^\eta - \alpha t_{j-1}^\eta, D_{i,k(n)}(t_{j-1}), \beta), \\ G_{t_j}(\mathbf{D}_{k(n)}(t_{j-1}), \mathbf{D}_n(t_j)) &= \prod_{i=1}^{C} \frac{1}{\sigma_Z \sqrt{2\pi}} \exp\big(- \frac{(y_{i,t_j} - D_{i,n}(t_j))^2}{2\sigma_Z^2}\big). \end{aligned}$$

The algorithm for the Bootstrap filter of [32] tailored to our Gamma Degradation Model (2), can be found in Appendix D.

The goal of the paper is to estimate the likelihood. To achieve this using Bootstrap, we use the output weights $w_{t_j,n}$ of the algorithm for all times $t_j \in \{t_1, \ldots, t_M\}$ and runs $n \in \{1, \ldots, N_{\text{part}}\}$. The estimator of the likelihood is given by:

$$\hat{\mathcal{L}}(\boldsymbol{\theta}; \mathbf{y}) = \prod_{j=1}^{M} \hat{\mathcal{L}}_{t_j}(\boldsymbol{\theta}; \mathbf{y}),$$

which is the product of the estimated likelihood values for each time step. The estimation of the likelihood values for each time step is as follows:

$$\hat{\mathcal{L}}_{t_j}(\boldsymbol{\theta}; \mathbf{y}) = \begin{cases} \dfrac{1}{N_{\text{part}}} \displaystyle\sum_{n=1}^{N_{\text{part}}} w_{t_j,n} & \text{if resampling occurred at time } t_j, \\ \dfrac{\sum_{n=1}^{N_{\text{part}}} w_{t_j,n}}{\sum_{n=1}^{N_{\text{part}}} w_{t_j-1,n}} & \text{otherwise.} \end{cases}$$

For further details, please refer to [32].

The application of the Bootstrap filter eliminates the possibility of failure. In the remainder of the paper, we apply CMC and the Bootstrap filter to both synthetic data as well as real data; see Section 4 for the results. Our findings demonstrate that the Bootstrap filter provides a more accurate estimation of the likelihood. As such, the Bootstrap filter emerges as a more reliable method compared to CMC.

4 Experiments

This section presents experimental results that focus on comparing the estimated likelihood values of the Bootstrap filter to Crude Monte Carlo. The experiments are conducted on synthetic data, following a similar approach as [31]. Specifically, we use the parameter set $\boldsymbol{\theta} = (\alpha, \eta, \beta, \mu_A, \sigma_A, \sigma_Z)^\top = (2, 2.5, 0.01, 0.5, 0.01, 0.01)$, consider time periods $\{2, 4, 6\}$ and set the number of components $C = 10$. Moreover, we examine three different numbers of unknown parameters: one unknown parameter, two unknown parameters and six unknown parameters. We apply the same prior distributions as in [31] to the unknown parameters in our research.

In these experiments, we compare the variance of the estimators of the two methods. The results reveal that CMC estimates exhibit a significantly higher relative variance than Bootstrap. Therefore,

the estimator of CMC demonstrates lower reliability in contrast to the estimator of Bootstrap. The relative errors associated with CMC are in fact so substantial, that the mean likelihood values for all different parameter settings, within 95% certainty, could as well be estimated to be zero, which results in not estimating higher likelihoods towards true parameter values. Bootstrap, on the other hand, demonstrates increasing likelihoods towards the true parameter setting, accompanied by distinct confidence intervals for parameter settings that are sufficiently apart from each other. These findings imply that when using Bootstrap, maximum likelihood estimation using different parameter settings is more likely to approach the true parameter setting. Consequently, the Bootstrap filter is considered a more reliable method for likelihood estimation given Gamma Degradation Model (2).

To assess the variance of the estimated likelihood value in the Bootstrap algorithm, we need a parameter denoted by N_{sim}, which represents the number of times the likelihood is estimated. Consequently, both the CMC and the Bootstrap filter methods are executed N_{sim} times, and the mean and variance of the estimates across these N_{sim} simulations are analyzed. All results are obtained with the number of particles $N_{\text{part}} = 10000$ and the number of times the likelihood is estimated $N_{\text{sim}} = 1000$. For the first experiment involving one unknown parameter, we conduct a thorough analysis of the results. However, for the experiments with two and six unknown parameters, we provide a more concise summary since the interpretation aligns closely with the results obtained for one unknown parameter.

4.1 One Unknown Parameter

The first considered case is that of leaving only one parameter unknown, which is α. Equivalent to [31], we adopt the prior distribution $U[0, 10]$ for α.

We begin by analyzing the mean estimated likelihood value along with the corresponding 95% confidence interval (CI), which is computed by assuming a normal distribution for the estimates. These results are illustrated in Figure 1(a) and Figure 1(b). Figure 1(a) shows all estimates for different parameter settings, ranging from the true parameter values to settings that deviate significantly from the true values. To generate the parameter values range, a drift ratio r_c is randomly added or subtracted from the true parameter value, with the drift direction reversed if the resulting value exceeds the bounds of the prior distribution. Specifically, for α with a prior distribution $U[w, y]$ and a true value of α_{true}, we generate values for α as follows:

$$\alpha = \alpha_{\text{true}} + r_c \cdot (y - w) \cdot random\{-1, 1\} \cdot \big[- \mathbb{1}\{\text{value outside prior distribution bounds}\}\big],$$

where $random\{-1, 1\}$ is a random integer of the set $\{1, -1\}$. Figure 1(b) provides a zoomed-in view of the y-axis, allowing for better visualization of the CMC values in proximity to the true parameter setting.

Upon examining Figure 1(a), we observe that for the Bootstrap filter a substantial difference in likelihood estimates between parameter values deviating less than a ratio of 0.02 from the true values and values with larger deviations. In fact, the mean estimate deviates approximately 21 times its 95% half-width from zero, which is about the same distance from the mean likelihood estimates of parameters with a deviation ratio of 0.05 or more to the true parameter setting. Hence, maximum likelihood estimation (MLE) will, most likely, result in a parameter estimate with a deviation ratio of less than 0.02 from the true parameter.

Figure 1(b) demonstrates the uncertainty in the CMC method due to its the high relative variance and hence relatively large CIs. The 95% CIs are overlapping for all different parameter values,

implying that parameter values with a high deviation are more likely to have the highest estimated likelihood when using CMC compared to Bootstrap. This suggests that MLE is less likely to approach the true parameter setting, indicating the lower reliability of CMC.

The significant reliability disparity between CMC and the Bootstrap filter stems from the high relative errors (REs) observed for CMC; See Kroese et al. [33] for details on the relative error. Table 1 provides a summary of the REs, excluding parameter values with a change above 0.050, as their likelihood estimates are too small to assess the RE accurately or they are affected by machine error in the RE assessment. While CMC displays extremely large REs, the Bootstrap filter exhibits significantly smaller REs. This stark contrast shows that the Bootstrap filter provides more accurate likelihood estimation and hence MLE of the parameters. Therefore, the Bootstrap filter clearly outperforms CMC in this case.

Method \ r_c	0.000	0.001	0.010	0.020
CMC	0.9708	0.9476	0.9894	0.8921
Bootstrap	0.0240	0.0254	0.0314	0.0431

Table 1: The RE for both estimation methods over the different parameter inputs for one unknown.

We also assess the sensitivity of the likelihood function around the true parameter value for both methods, as this is crucial for MLE. Table 2 presents the relative change in mean likelihood estimates compared to the estimate for the true parameter. Thus, the true parameters correspond to a value of 100%. If, for instance, a random parameter change of 0.001 results in a mean likelihood estimate twice the size of the one of the true parameters, the relative value is 200%. This analysis assesses the sensitivity of the likelihood estimate to variations in parameter values.

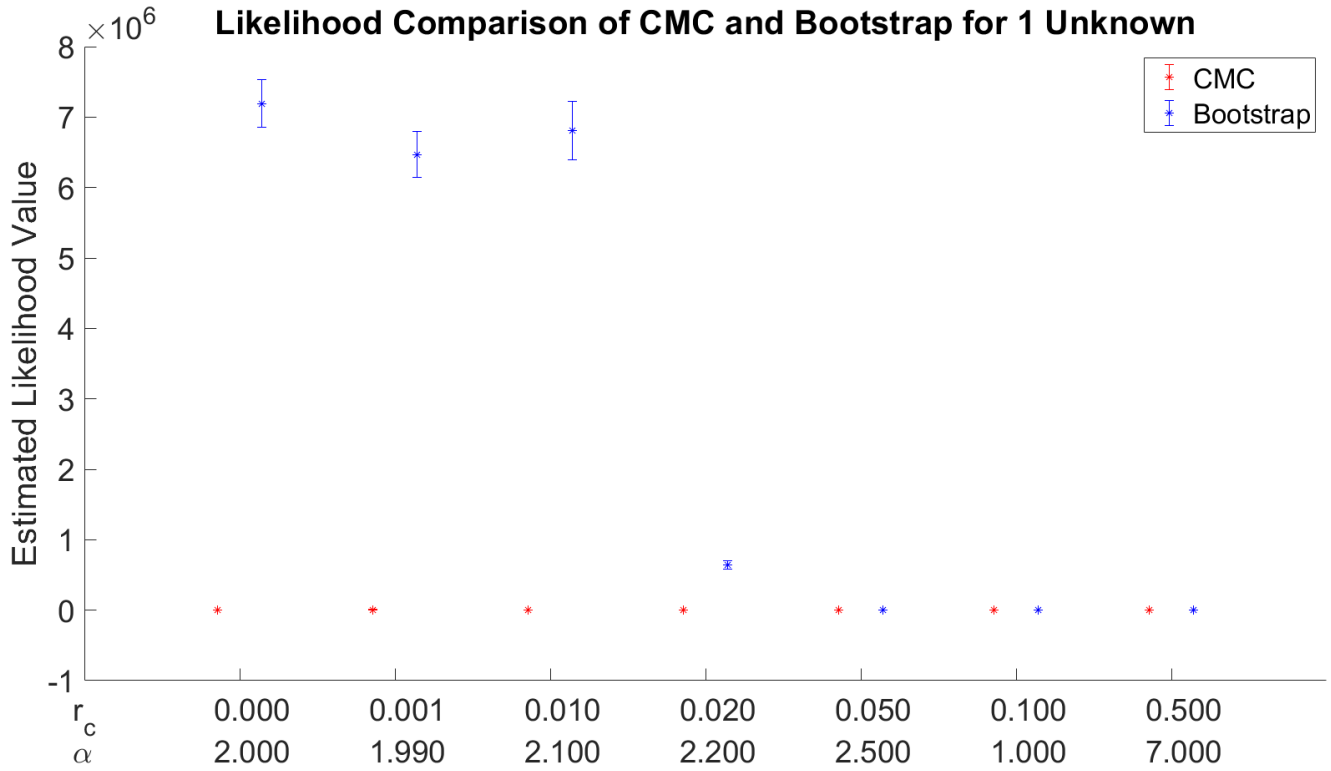
In the table, 'Random' indicates the random change as displayed in Figure 1. 'Positive' indicates all changes are positive and vice versa. The lower bound of the CI for the value of r_c indicates the lowest value in the 95% likelihood CI for that parameter divided by the highest value in the 95% likelihood interval of the true parameter. The upper bound is the highest value within the 95% interval for that parameter divided by the lowest value in the interval of the true parameter. This leads to an approximately 90% CI.

CMC demonstrates high elasticity with small random parameter changes, with an increase in mean likelihood for nearby non-true parameters. In Bootstrap, the likelihood shows less elasticity for small random parameter changes, while for larger parameter changes the estimates decrease more rapidly. The Bootstrap filter results are more favorable for MLE. However, the CMC results are very uncertain, due to their large relative variance.

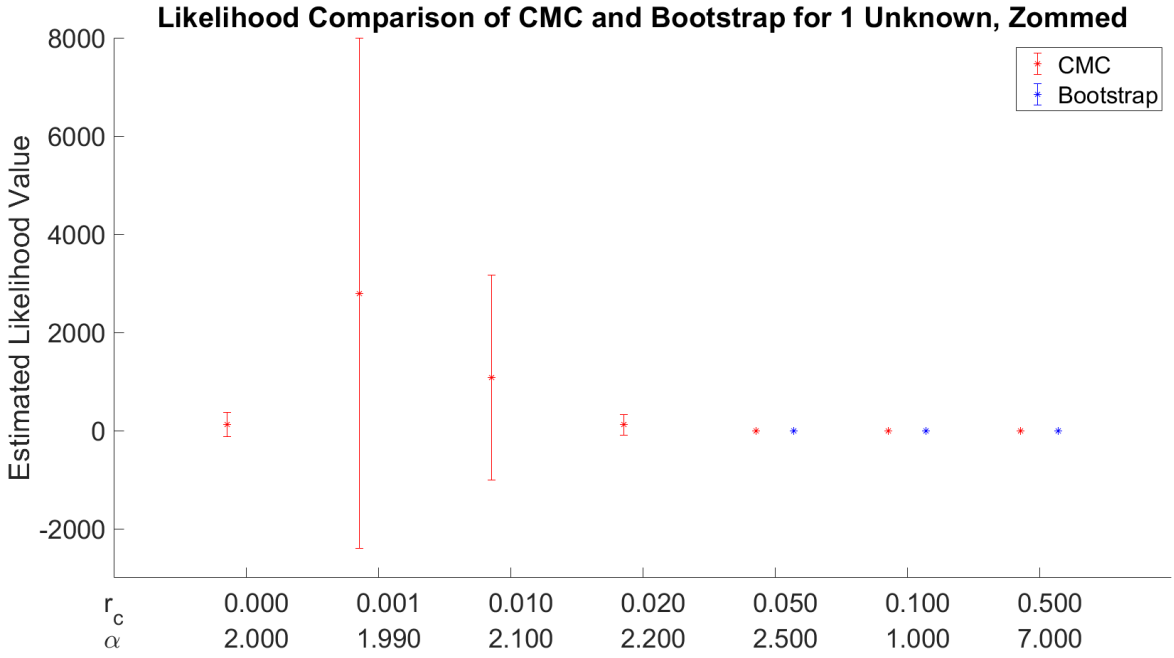
While many likelihoods of parameters for higher r_c being zero percent of the original likelihood, in Table 3 we observe the likelihoods are still descending in size. In other words, for parameter settings closer to the true one, the likelihoods are generally larger.

Method \ r_c	0.000	0.001	0.010	0.020	0.050	0.100	0.500
CMC	4.9	7.9	7.0	4.8	$-3.5 \cdot 10$	$-2.4 \cdot 10^2$	$-3.0 \cdot 10^3$
Bootstrap	$1.6 \cdot 10$	$1.6 \cdot 10$	$1.6 \cdot 10$	$1.3 \cdot 10$	-7.0	$-2.4 \cdot 10^2$	$-7.7 \cdot 10^3$

Table 3: The log transformed likelihood estimate for one unknown parameter.



(a) Including the mean estimated likelihood value and CI for different parameter settings of the one unknown.



(b) Including the mean estimated likelihood value and CI for different parameter settings of the one unknown with a zoomed in y-axis.

Figure 1: Likelihood Comparison for 1 Unknown Parameter.

Method \ r_c		0.000	0.001	0.010	0.020	0.050	0.100	0.500
CMC	Random	100.0%	2158.4% [0; Inf]	830.9% [0; Inf]	92.4% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0;Inf]
	Positive	100.0%	3418.7% [0; Inf]	830.9% [0; Inf]	92.4% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]
	Negative	100.0%	2158.4% [0; Inf]	1.7% [0; Inf]	29.3% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	
Bootstrap	Random	100.0%	90.0% [81.6; 99.1]	94.6% [84.8; 105.4]	8.9% [7.8; 10.2]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]
	Positive	100.0%	109.0% [99.7; 119.2]	91.2% [82.5; 100.7]	9.5% [8.3; 10.8]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]
	Negative	100.0%	90.0% [81.6; 99.1]	10.2% [9.3; 11.3]	0.1% [0.1; 0.1]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	

Table 2: Relative change in likelihood over a certain change in parameter values for one unknown parameter.

To summarize and assess the performance of each technique, we check for distinct CIs and descending values over the range of parameter values, through the log-transformed values and RE. Distinct CIs result in a much more reliable MLE. Through an educated guess we take a RE of 0.8 for CMC and 0.05 for Bootstrap. Denoting the likelihood estimate as l, the CI bounds are computed as follows:

$$\textbf{Bound} = log(l) + log(1 \pm 1.96 \cdot RE).$$

The value inside the lower bound becomes negative we simply regard it as 0, as the likelihood cannot go beneath zero.

The CIs for CMC are all overlapping, while for the Bootstrap filter for $r_c > 0.01$ none of the CIs overlap. Thus, considering the log mean estimates, within the 95% CI, the mean likelihood for the Bootstrap filter is increasing towards the true parameter value for $r_c \geq 0.02$, which is not the case for CMC. To conclude, the Bootstrap filter is more reliable than CMC.

4.2 Two Unknown Parameters

The second case considered is that of leaving two parameters unknown, which are η and μ_A. Again equivalent to [31], we adopt the prior distributions $U[0, 5]$ for η and $U[0, 1]$ for μ_A.

The results highlight again the superior reliability of the Bootstrap filter compared to CMC in maximum likelihood estimation. Table 6 presents the relative change in mean likelihood estimates compared to the estimate for the true parameters. The table set-up is similar to that of Table 2.

The Bootstrap filter exhibits significantly smaller REs and increasing likelihood estimates towards the true parameter values, as shown in Figure 2, Table 4, and Table 5. Therefore, there are distinct 95% CIs of the Bootstrap filter for parameter values with ratio of change $r_c < 0.010$ and $r_c > 0.020$. In fact, assuming a 0.04 RE, none of the CIs overlap and they are increasing towards the true parameter values from $r_c > 0.001$, which further demonstrate its reliability. On the other hand, CMC shows overlapping CIs due to its higher RE. Therefore, the Bootstrap filter is significantly more reliable for MLE.

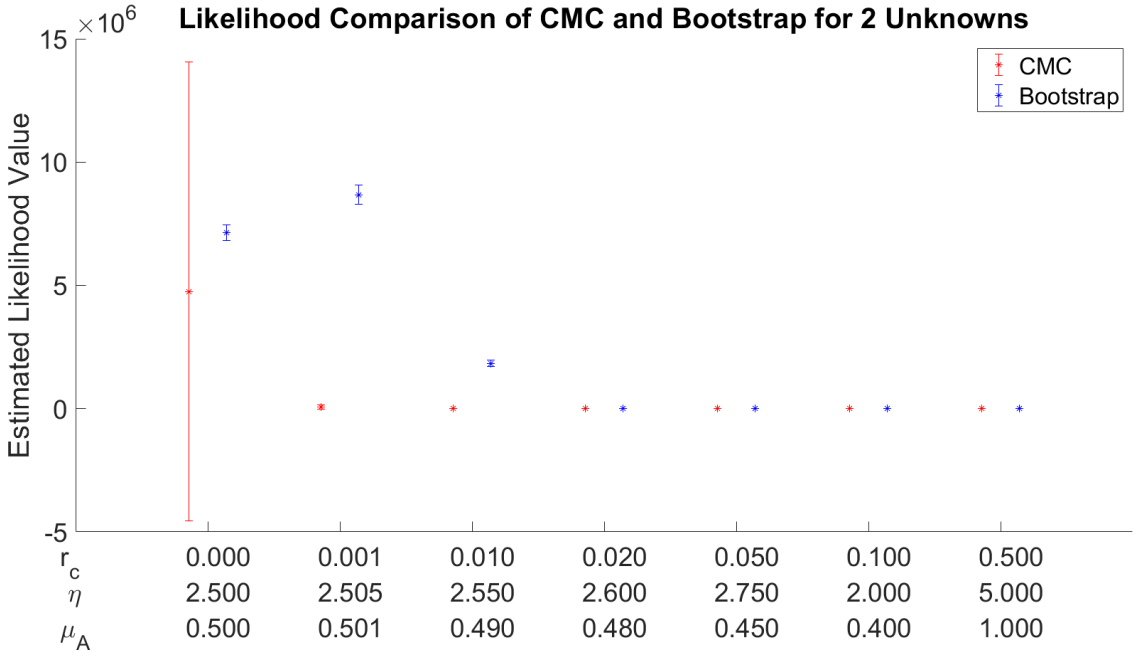


Figure 2: Likelihood comparison, including the mean estimated likelihood value and CI for different parameter settings of the two unknowns.

Method \ r_c	0.000	0.001	0.010
CMC	0.9982	0.7752	0.9974
Bootstrap	0.0229	0.0228	0.0367

Table 4: The RE for both estimation methods over the different parameter inputs for two unknowns.

Method \ r_c	0.000	0.001	0.010	0.020	0.050	0.100	0.500
CMC	$1.5 \cdot 10$	$1.1 \cdot 10$	6.9	-9.3	$-1.7 \cdot 10^2$	$-4.0 \cdot 10^2$	$-1.0 \cdot 10^{4*}$
Bootstrap	$1.6 \cdot 10$	$1.6 \cdot 10$	$1.4 \cdot 10$	4.0	$-1.8 \cdot 10^2$	$-4.3 \cdot 10^2$	$-1.2 \cdot 10^7$

Table 5: The log transformed likelihood estimate for two unknown parameters.
* This is the lower bound for CMC, considered zero.

Method \ r_c		0.000	0.001	0.010	0.020	0.050	0.100	0.500
CMC	Random	100.0%	1.3% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]
	Positive	100.0%	1.3% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]
	Negative	100.0%	0.3% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]
Bootstrap	Random	100.0%	121.7% [111.2; 133.1]	25.7% [22.8; 28.8]	0.0% [0; 0]	0.0% [0; 0]	0.0% [0; 0]	0.0% [0; 0]
	Positive	100.0%	121.7% [111.2; 133.1]	13.3% [11.5; 15.2]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]
	Negative	100.0%	77.2% [70.2; 84.9]	0.2% [0.2; 0.2]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	

Table 6: Relative change in likelihood over a certain change in parameter values for two unknown parameters.

Method \ r_c	0.000	0.001
CMC	0.7581	0.9577
Bootstrap	0.0234	0.0386

Table 7: The RE for both estimation methods over the different parameter inputs for six unknowns.

4.3 Six Unknown Parameters

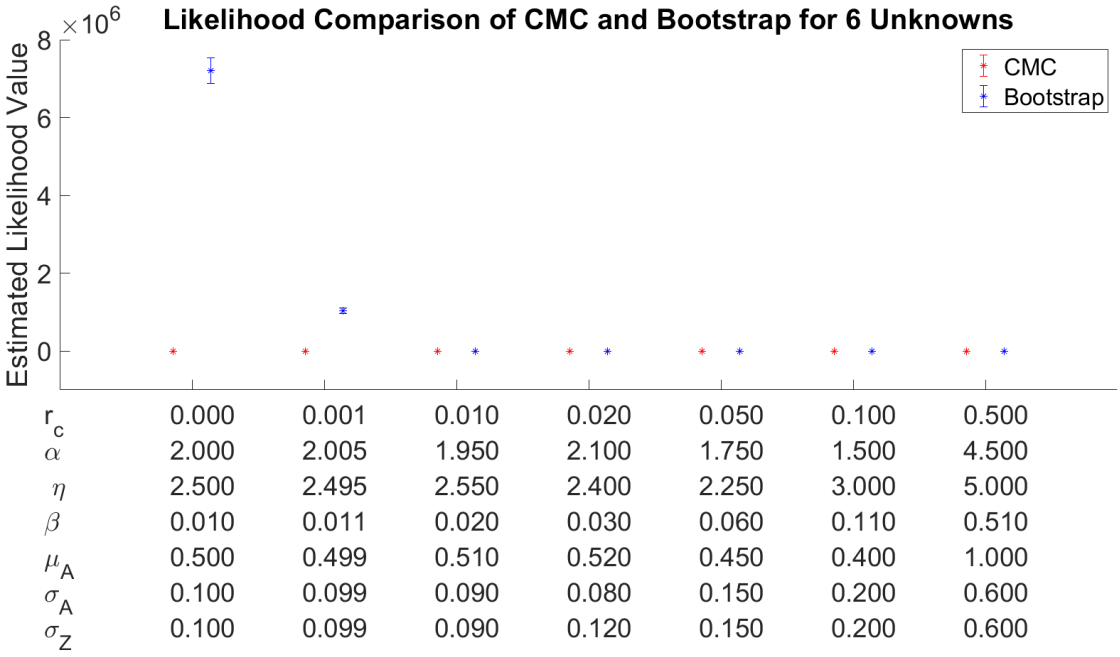
Last, we consider the case with all six parameters left unknown. Equivalent to [31], the prior distributions for the parameters are $U[0, 5]$ for α and η, and $U[0, 1]$ for μ_A, β, σ_A and σ_Z.

The results reinforce the superior reliability of the Bootstrap filter over CMC in maximum likelihood estimation. The Bootstrap filter exhibits significantly lower REs, as evidenced in Table 7. Table 9 presents the relative change in mean likelihood estimates compared to the estimate for the true parameters. The table set-up is similar to that of Table 2.

In Table 8, it is apparent that the mean likelihood estimates for the Bootstrap filter increase towards the true parameter values, indicating accurate estimation. The small REs further ensure that likelihoods are estimated in the correct order, favoring parameters close to the true values. Conversely, the large REs for CMC introduce uncertainty in the estimation order, as even high ratios of change r_c may yield high likelihood estimates. While the Bootstrap filter remains superior, it is important to note that both methods are sensitive to small ratio changes in parameter values, as shown in Table 9. This sensitivity complicates MLE for both the Bootstrap filter and CMC.

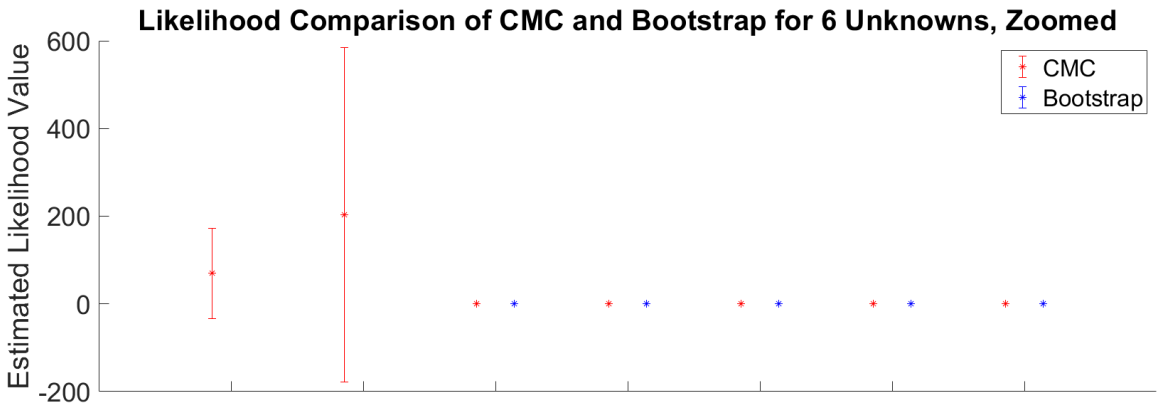
Method \ r_c	0.000	0.001	0.010	0.020	0.050	0.100	0.500
CMC	4.2	5.3	$-3.6 \cdot 10^2$	$-4.7 \cdot 10^2$	$-4.1 \cdot 10^2$	$-6.1 \cdot 10^3$	$-1.0 \cdot 10^{4*}$
Bootstrap	$1.6 \cdot 10$	$1.4 \cdot 10$	$-1.3 \cdot 10^3$	$-1.6 \cdot 10^3$	$-2.0 \cdot 10^3$	$-1.2 \cdot 10^5$	$-4.4 \cdot 10^9$

Table 8: The log transformed likelihood estimate for six unknown parameters.
* This is the lower bound for CMC, considered zero.



r_c	0.000	0.001	0.010	0.020	0.050	0.100	0.500
α	2.000	2.005	1.950	2.100	1.750	1.500	4.500
η	2.500	2.495	2.550	2.400	2.250	3.000	5.000
β	0.010	0.011	0.020	0.030	0.060	0.110	0.510
μ_A	0.500	0.499	0.510	0.520	0.450	0.400	1.000
σ_A	0.100	0.099	0.090	0.080	0.150	0.200	0.600
σ_Z	0.100	0.099	0.090	0.120	0.150	0.200	0.600

(a) Including the mean estimated likelihood value and CI for different parameter settings of the six unknowns.



r_c	0.000	0.001	0.010	0.020	0.050	0.100	0.500
α	2.000	2.005	1.950	2.100	1.750	1.500	4.500
η	2.500	2.495	2.550	2.400	2.250	3.000	5.000
β	0.010	0.011	0.020	0.030	0.060	0.110	0.510
μ_A	0.500	0.499	0.510	0.520	0.450	0.400	1.000
σ_A	0.100	0.099	0.090	0.080	0.150	0.200	0.600
σ_Z	0.100	0.099	0.090	0.120	0.150	0.200	0.600

(b) Including the mean estimated likelihood value and CI for different parameter settings of the six unknowns with a zoomed in y-axis.

Figure 3: Likelihood Comparison for 6 Unknown Parameters.

Method \ r_c		0.000	0.001	0.010	0.020	0.050	0.100	0.500
CMC	Random	100.0%	291.9% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]
	Positive	100.0%	18.7% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]
	Negative*	100.0%	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]	0.0% [0; Inf]		
Bootstrap	Random	100.0%	14.4% [12.7; 16.2]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]
	Positive	100.0%	3.9% [3.4; 4.5]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]
	Negative*	100.0%	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]	0.0% [0.0; 0.0]		

Table 9: Relative change in likelihood over a certain change in parameter values for six unknown parameters.
* With β kept constant after 0.001 negative change.

Overall, the experiments conducted in this section have provided compelling evidence that the Bootstrap filter outperforms CMC in terms of reliable and accurate likelihood estimation, assuming that Gamma Degradation Model (2) effectively models degradation. The poor performance of CMC might explain the efficiency issues of the likelihood-based method in [31], where CMC is employed for the likelihood computation. In addition to our experiments, we present a case study in the following section in which the performance of the two Monte Carlo methods is compared using real-life data.

5 Case Study

Shelved drugs are susceptible to degradation, resulting in a decline in their effectiveness over time due to chemical processes. In this section, we conduct a case study using real drug shelf-life data from Chow and Shao [10]. The data includes measurements of drug potency in percentage of claimed potency at different time periods (after 0, 1, 2, and 3 years) for 24 drug batches. The study assumes a linear degradation model and performs WLS. Our paper however, assumes the data to be better represented by a Gamma-distributed degradation model. Given Gamma Degradation Model (2), we compare the performance of CMC with the Bootstrap filter in estimating likelihood values of parameter inputs, aiming to find the most reliable method in estimating this drug shelf-life degradation. The detailed data set can be found in the Appendix, Table 11.

Our findings clearly demonstrate that the Bootstrap filter outperforms CMC in terms of reliability, offering lower relative variance and more accurate estimates. However, both methods require an extensive number of simulations, leading to a high relative error and overlapping confidence intervals in the results. Additionally, the likelihoods obtained from both approaches are remarkably small, likely due to the sensitivity of the likelihood estimates. This section serves to highlight that, despite both methods performing poorly for this case study, the Bootstrap filter still outperforms CMC.

To apply our specific CMC and the Bootstrap filter algorithm to the drug degradation data, a transformation is performed to align the data with the Gamma Degradation Model (2). The data set from [10] namely represents the drug potency of batch i at time t_j, rather than the observed

degradation of the drug as required by our model. Therefore, we define the observed degradation to be:

$$y_{i,t_j} = 105 - \text{drug potency of batch } i \text{ at time } t_j, \tag{7}$$

where time 0 is omitted based on the model assumptions, and we assume the initial potency to be 105 percent of the claimed potency. It is important to note that this last assumption merely affects the mean of the initial degradation μ_A by a constant.

Our study employs similar prior distributions to [31], with the exception of σ_Z, for which we make an adjustment based on indications in [10] that this parameter may have a larger value than the upper bound used in [31]. Altering the prior distributions for the other parameters does not significantly improve the results. Therefore, we integrate the following prior distributions: $U[0, 50]$ for α, $U[0, 5]$ for η and μ_A, $U[0, 1]$ for β and σ_A, and $U[0, 5]$ for σ_Z.

In Section 4.3, the likelihood estimate shows a rapid decline even for small deviations r_c, indicating that many parameter settings for the real-life data may yield a numerically zero estimated likelihood. To address this, extensive testing across a wide range of parameter settings is necessary to obtain non-zero likelihood estimates for numerical MLE.

To illustrate the challenge, let us consider the scenario where we want all parameters to be within a 0.01 ratio of change. This results in a mean likelihood of order 10^{-157} for CMC and even smaller for the Bootstrap filter in the six unknowns case for synthetic data. The probability of drawing such a parameter setting in one draw is estimated to be $6.4 \cdot 10^{-11}$ using uniform draws. Consequently, the simulation requires an enormous amount of draws, say for instance $N_\theta = 10^{11}$, to be likely to approach the true values, which is computationally infeasible.

To mitigate this issue, we explore a total of $N_\theta = 18000$ different parameter settings in our analysis. Additionally, we employ $N_{\text{part}} = 10000$ particles and perform $N_{\text{sim}} = 100$ simulations to ensure the accuracy of our analysis.

Table 10 shows the RE for both estimation methods over the different parameter inputs for the ten highest likelihoods and the overall mean of all REs excluding 0 and NaN values. Values correspond to the x highest likelihood or the mean. The results of CMC are visualized in Figure 4. These results display remarkably low mean likelihood estimates compared to those in Section 4.3. While this could suggest a potential mismatch between the model and real data, a more plausible explanation is the inadequate approximation of the true parameter values due to the limited number of explored parameter settings N_θ. It is important to note that low likelihood values themselves are not necessarily problematic if the mathematical software can handle them and if the estimates still provide the correct indication towards the true parameter values. However, the 95% CIs in Figure 4 predominantly overlap, indicating inaccurate results and suggesting that the estimates may not be increasing towards the true parameter values. These wide CIs are caused by the high RE observed in Table 10, with a mean of 0.8473, rendering the method unreliable for MLE.

The results of Bootstrap, depicted in Figure 5, exhibit similarly small likelihood estimates as CMC, likely due to the limited parameter search. Additionally, the figure displays overlapping 95% CIs, indicating unreliability. By examining the REs of the Bootstrap filter in Table 10, we observe substantially higher values compared to the synthetic data study, with a mean RE of 0.3220. This could be partly caused to to the smaller number of simulations N_{sim}. Since a reasonable RE should be below 0.10, this higher RE renders the method in this case unreliable. Nevertheless, the Bootstrap filter exhibits a significantly smaller RE compared to CMC, establishing the Bootstrap filter as the more reliable and accurate method for MLE.

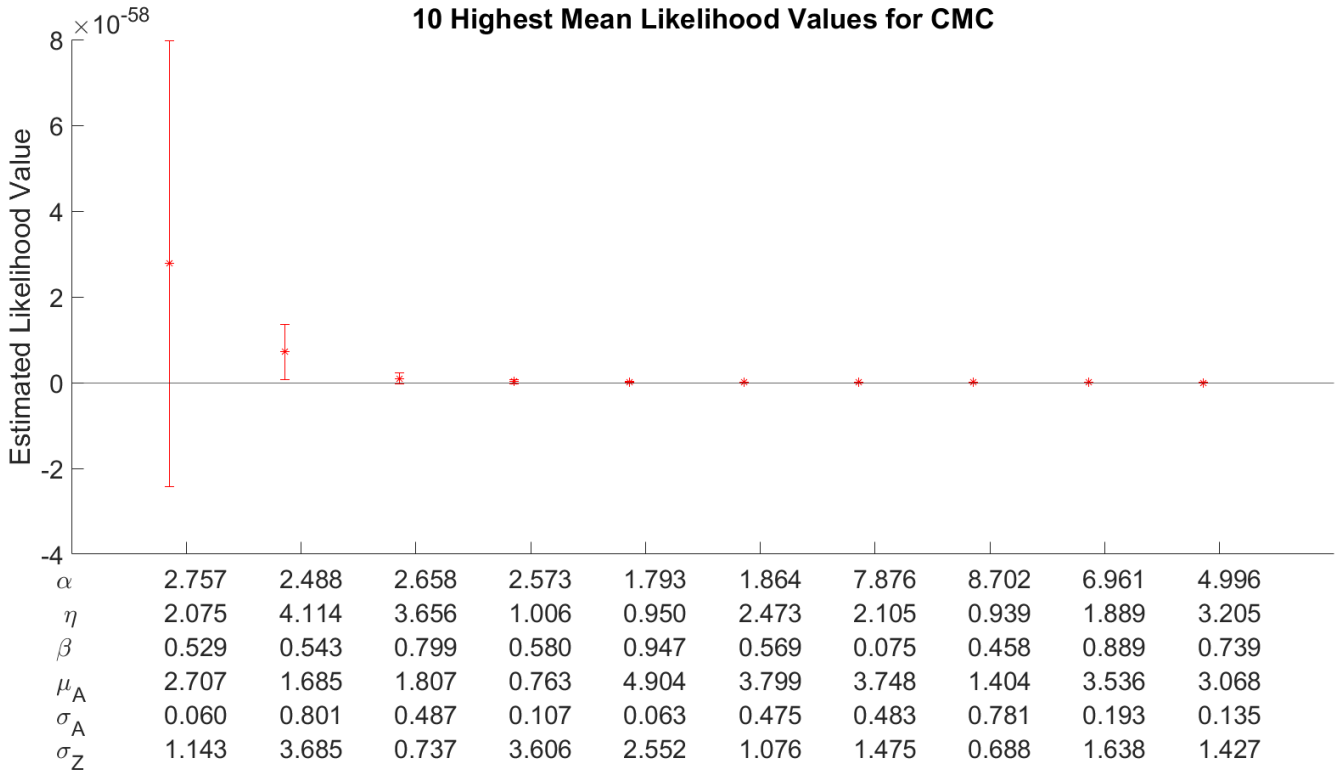


Figure 4: The 10 highest likelihood values of 18000 different parameter settings of CMC for the real data set, including the line indicating $y = 0$.

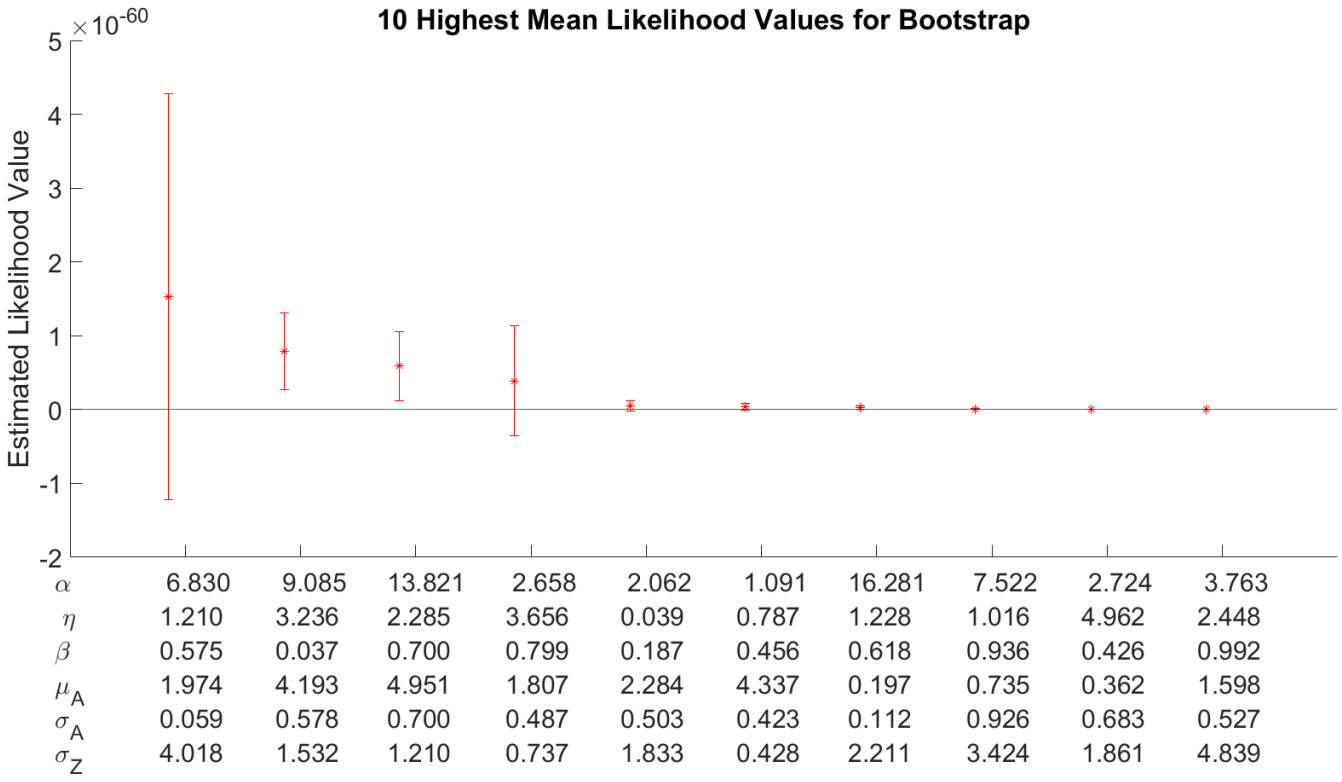


Figure 5: The 10 highest likelihood values of 18000 different parameter settings of the Bootstrap filter for the real data set, including the line indicating $y = 0$.

Method \ x	1	2	3	4	5	6	7	8	9	10	mean
CMC	0.9575	0.4554	0.7063	0.9612	0.9962	1.0000	0.9744	0.6981	0.1408	0.6556	0.8473
Bootstrap	0.9177	0.3367	0.4040	0.9740	0.7304	0.5741	0.4412	0.1462	0.3872	0.4092	0.3220

Table 10: Relative errors for CMC and Bootstrap methods.

6 Conclusions

This paper demonstrates the superiority of the Bootstrap filter over CMC in several aspects. First, in the analytical study we proved that for Gamma Degradation Model (2), Crude Monte Carlo fails to estimate the likelihood for a too large data set. This is attributed to CMC's inherent structure, which allows for a violation of the model assumptions, specifically the estimation of negative degradation over time. In contrast, SMC, and hence Bootstrap, avoids failure by not allowing such model violations.

Second, our experiments using synthetic data demonstrate that the Bootstrap filter is significantly more reliable than CMC. The Bootstrap filter provides accurate likelihood estimates without overlapping confidence intervals for reasonably distant parameters. This is attributed to the significantly lower relative errors of the Bootstrap filter compared to CMC. The likelihood estimates of the Bootstrap filter are predominantly increasing towards the true parameter values and have substantial differences for different parameter inputs. This further reduces the chances of overlapping confidence intervals for Bootstrap, increasing its probability of correctly indicating the true direction of the parameters. Overall, the Bootstrap filter outperforms CMC in likelihood estimation given Gamma Degradation Model (2), and is due to its low relative error is in general a well-performing method for accurately estimating likelihoods.

Third, in our case study, we find that the Bootstrap filter exhibits a significantly lower relative error compared to CMC, making it a more reliable method for maximum likelihood estimation despite the unfavorable high relative error in both methods. The relative error of the Bootstrap filter is generally almost 3 times smaller than that of CMC.

The difference in relative error between our case study and our experiments using synthetic data may have various explanations. One possibility is that the model does not fit accurately, potentially due to the presence of a hidden stochastic component in the true model. Conducting a composite hypothesis test on a normally- plus Gamma-distributed random variable, along with access to more data, could shed light on the presence of such hidden stochasticity. Additionally, the larger relative errors could be attributed to machine errors arising from dealing with very small values. Increasing the number of simulations N_{sim} could potentially reduce the relative error, as it was ten times smaller in the real data study compared to the synthetic data study for computational efficiency.

In our case study, approaching the true parameter values requires an enormous number of parameter settings N_θ. To improve the parameter search, we recommend combining the Bootstrap particle filter with an optimization method. This approach, utilizing a policy for parameter selection and a stopping criteria, can converge more efficiently to the true parameter values compared to the random sampling approach used in this case study. Genetic algorithms are suitable optimization methods for complex likelihoods, such as the one considered here. Applying maximum likelihood estimation using particle filtering and a genetic algorithm is expected to provide a more accurate and reliable estimation of the parameters. For further details on genetic algorithms in maximum likelihood estimation, refer to [34].

Another, more general, improvement to our method is applying the auxiliary particle filter instead of the Bootstrap particle filter. As discussed in Section 1, the auxiliary particle filter has shown to provide more accurate likelihood estimates.

Additionally, it would be interesting to explore the performance of Markov Chain Monte Carlo (MCMC). While SMC might be more robust and flexible for complex likelihood functions, MCMC has the potential to improve computational efficiency and enhance the estimation of the degradation system's state.

References

[1] JR Karr and LA Toth. Fish communities of midwestern rivers: a history of degradation. *BioScience*, 35(2):90–95, 1985.

[2] T.S. Dharmarajan. Physiology of aging. In *Geriatric Gastroenterology*, chapter 6. Springer, New York, 2021.

[3] Amgad Muneer, Suliman Mohamed Fati, Nur Arifin Akbar, David Agustriawan, and Setyanto Tri Wahyudi. iVaccine-deep: Prediction of COVID-19 mRNA vaccine degradation using deep learning. *Journal of King Saud University - Computer and Information Sciences*, 34(9):7419–7432, 10 2022.

[4] Dongliang Lu, Mahesh D. Pandey, and Wei Chau Xie. An efficient method for the estimation of parameters of stochastic gamma process from noisy degradation measurements. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 227(4):425–433, 8 2013.

[5] William Denson. The history of reliability prediction. *IEEE Transactions on Reliability*, 47(3):321–328, 1998.

[6] W Kahle, S Mercier, and C Paroissin. *Degradation Processes in Reliability*. John Wiley & Sons, Inc, 2016.

[7] Jin Lu. *Degradation processes and related reliability models*. PhD thesis, McGill University, 1995.

[8] Michael S. Hamada, Alyson G. Wilson, C. Shane Reese, and Harry F. Martz. *Bayesian Reliability*. Springer Series in Statistics. Springer New York, New York, NY, 2008.

[9] Ding-Geng Chen, Yuhlong Lio, Hon Keung Tony Ng, and Tzong-Ru Tsai. *Statistical Modeling for Degradation Data*. Springer, Singapore, 2017.

[10] Shein-Chung Chow and Jun Shao. Estimating drug shelf-life with random batches. *Biometrics*, 47(3):1071–1079, 1991.

[11] C. Joseph Lu and William Q. Meeker. Using degradation measures to estimate a time-to-failure distribution. *Technometrics*, 35(2):161, 5 1993.

[12] Xiao Sheng Si, Wenbin Wang, Chang Hua Hu, Mao Yin Chen, and Dong Hua Zhou. A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation. *Mechanical Systems and Signal Processing*, 35(1-2):219–237, 2 2013.

[13] Mohamed Abdel-Hameed. A gamma wear process. *IEEE Transactions on Reliability*, R-24(2):152–153, 1975.

[14] J.M. van Noortwijk. A survey of the application of gamma processes in maintenance. *Reliability Engineering & System Safety*, 94(1):2–21, 2009.

[15] Rommert Dekker. Applications of maintenance optimization models: a review and analysis. *Reliability Engineering & System Safety*, 51(3):229–240, 1996.

[16] Rommert Dekker and Philip A. Scarf. On the impact of optimisation models in maintenance decision making: the state of the art. *Reliability Engineering & System Safety*, 60(2):111–119, 1998.

[17] Jan M. van Noortwijk and E. Bart Peerbolte. Optimal sand nourishment decisions. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 126(1):30–38, 2000.

[18] Lennaert J. P. Speijker, Jan M. van Noortwijk, Matthijs Kok, and Roger M. Cooke. Optimal maintenance decisions for dikes. *Probability in the Engineering and Informational Sciences*, 14(1):101–121, 2000.

[19] M.J. Kallen and J.M. van Noortwijk. Optimal maintenance decisions under imperfect inspection. *Reliability Engineering & System Safety*, 90(2):177–185, 2005.

[20] Jan M. van Noortwijk and Pieter H.A.J.M. van Gelder. Optimal maintenance decisions for berm breakwaters. *Structural Safety*, 18(4):293–309, 1996.

[21] Martin Crowder and Jerald Lawless. On a scheme for predictive maintenance. *European Journal of Operational Research*, 176(3):1713–1722, 2007.

[22] J.M van Noortwijk and H.E Klatter. Optimal inspection decisions for the block mats of the Eastern-Scheldt barrier. *Reliability Engineering & System Safety*, 65(3):203–211, 1999.

[23] Kai Song and Lirong Cui. A common random effect induced bivariate gamma degradation process with application to remaining useful life prediction. *Reliability Engineering & System Safety*, 219:108200, 2022.

[24] Xinliang Dai, Sheng Qu, Hao Sui, and Pingbo Wu. Reliability modelling of wheel wear deterioration using conditional bivariate gamma processes and Bayesian hierarchical models. *Reliability Engineering & System Safety*, 226:108710, 2022.

[25] Zhao-Hui Lu, Hao-Peng Qiao, Xuan-Yi Zhang, and Yan-Gang Zhao. An innovative method for space-time-dependent reliability analysis. *Structural Safety*, 102:102326, 2023.

[26] Nima Gorjian, Lin Ma, Murthy Mittinty, Prasad Yarlagadda, and Yong Sun. A review on degradation models in reliability analysis. *Engineering Asset Lifecycle Management - Proceedings of the 4th World Congress on Engineering Asset Management, WCEAM 2009*, pages 369–384, 2009.

[27] N. J. Gordon, D. J. Salmond, and A. F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings, Part F: Radar and Signal Processing*, 140(2):107–113, 4 1993.

[28] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 72(3):269–342, 6 2010.

[29] Edward L. Ionides, Anindya Bhadra, Yves Atchadé, and Aaron King. Iterated filtering. *The Annals of Statistics*, 39(3):1776–1802, 6 2011.

[30] Deborshee Sen, Ajay Jasra, and Yan Zhou. Some contributions to sequential Monte Carlo methods for option pricing. *Journal of Statistical Computation and Simulation*, 87(4):733–752, 3 2016.

[31] Indranil Hazra, Mahesh D. Pandey, and Noldainerick Manzana. Approximate Bayesian computation (ABC) method for estimating parameters of the gamma process using noisy data. *Reliability Engineering and System Safety*, 198, 6 2020.

[32] Nicolas Chopin and Omiros Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer Series in Statistics, 2020.

[33] Dirk P. Kroese, Thomas Taimre, and Zdravko I. Botev. *Handbook of Monte Carlo Methods*. Wiley Blackwell, 9 2011.

[34] K. C. Sharman and G. D. McClurkin. Genetic algorithms for maximum likelihood parameter estimation. *IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings (ICASSP)*, 4:2716–2719, 1989.

Appendices

A Notation

The observed degradation of component i is

$$Y_i(t) = A_i + X_i(t) + Z_i(t),$$

measured at times $0 \leq t_1 \leq t_2 \leq \ldots \leq t_M$.

Below we present a list of definitions. All realizations of random variables are represented by a lowercase letter. All boldfaced characters are vectors.

1. $C \in \mathbb{N}$ – the number of identical components.
2. $M \in \mathbb{N}$ – the number of measurements performed.
3. $Y_i(t) \in \mathbb{R}$ – the observed degradation measurement of the ith component at time t.
4. $A_i \in \mathbb{R}$ – the initial degradation of the ith component.
5. $X_i(t) \in \mathbb{R}^+$ – the true degradation of the ith component at time t (without the initial degradation A_i).
6. $Z_i(t) \in \mathbb{R}$ – the time-invariant measurement error for the ith component at time t.
7. $D_i(t) = A_i + X_i(t) \in \mathbb{R}$ – the total true degradation until now.
8. $y_{i,t}, a_i, x_{i,t}, z_{i,t}, d_{i,t}$ – the realized values of the above mentioned random variables.
9. $\mathbf{Y_i} = (Y_i(t_1), \ldots, Y_i(t_M))^\top$ – the observed degradation data for the ith component as a column vector.
10. $\mathbf{Y} = (\mathbf{Y_1}, \ldots, \mathbf{Y_C})$ – the observed degradation data with measurement errors as a matrix, with column i the observed degradation for component i for time periods $t \in \{t_1, \ldots, t_M\}$.
11. $\Delta Z_i(t_j) = Z_i(t_j) - Z_i(t_{j-1})$ – the measurement error increments of the ith component for $1 \leq j \leq M$.
12. $\Delta X_i(t_j) = X_i(t_j) - X_i(t_{j-1})$ – the true degradation increments of the ith component for $1 \leq j \leq M$.
13. $\Delta Y_i(t_j) = Y_i(t_j) - Y_i(t_{j-1})$ – the observed increments of the ith component for $1 \leq j \leq M$.
14. $\mathbf{\Delta Y_i} = (\Delta Y_i(t_1), \ldots, \Delta Y_i(t_M))^\top$ – the observed increments of the ith component.
15. $\mathbf{\Delta Y} = (\mathbf{\Delta Y_1}, \ldots, \mathbf{\Delta Y_C})^\top$ – the observed increments.
16. $a(t) = \alpha t^\eta$ – the time-dependent shape function (corresponds to the Gamma PDF shape parameter).
17. $\beta > 0$ – the scale parameter of the Gamma PDF.

18. $X \sim N(\mu, \sigma)$ – univariate Gaussian $f_N(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), x \in \mathbb{R}$.

19. $X \sim \text{Gamma}(\alpha, \beta)$ – $f_G(x; \alpha, \beta) = \frac{\beta^{-\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}} \mathbb{1}_{\{x \geq 0\}}$, where $\mathbb{1}$ is the indicator function.

20. $\boldsymbol{\theta} = (\alpha, \eta, \beta, \mu_A, \sigma_A, \sigma_Z)^\top$ – the set of process parameters.

21. $f_{\boldsymbol{\theta}}(\mathbf{y_i})$ – the pdf of the realized observed data, given parameters $\boldsymbol{\theta}$ for the model.

22. $\mathcal{L}(\boldsymbol{\theta}; \mathbf{y})$ – the likelihood function

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}) = \prod_{i=1}^{C} \mathcal{L}(\boldsymbol{\theta}; \mathbf{y}_i).$$

23. $\mathcal{L}_t(\boldsymbol{\theta}; \mathbf{y})$ – the likelihood for time t.

The following definitions are introduced for numerical estimation through (Sequential) Monte Carlo simulation.

24. $N \in \mathbb{N}$ – the number of runs performed.

25. $\mathbf{w}_t = \{w_{t,h}\}_{h=1}^{N_{\text{part}}}$ – the weights for all N_{part} runs at time t.

26. $\mathbf{W}_t = \{W_{t,h}\}_{h=1}^{N_{\text{part}}}$ – the normalized weights for all N_{part} runs at time t.

27. $ESS_t(\mathbf{w}_t)$ – the effective sample size at time t (given the weights).

28. $M_{t_0}(\Delta d_{i,t_0})$ – the importance sampling function at time 0.

29. $M_{t_j}(D_{i,h}(t_j), \Delta d_{i,t_j})$ – the importance sampling function at time t for run h.

30. $G_{t_0}(\mathbf{D}_h(t_0))$ – the observation probability function at time 0 for run h.

31. $G_{t_j}(\mathbf{D}_{k(h)}(t_{j-1}), \mathbf{D}_h(t_j))$ – the observation probability function at time t for run h.

B Likelihood Derivations

Lemma 1. *Given Gamma Degradation Model* (2)*, the component likelihood equals:*

$$f_{\boldsymbol{\theta}}(\mathbf{y_i}) = \mathbb{E}_{A_i, \mathbf{Z}_i}\Big[\prod_{t_j=t_1}^{t_M} \frac{\Delta x_{i,l}^{(\alpha t_j^\eta - \alpha t_{j-1}^\eta - 1)} \exp(\frac{-\Delta x_{i,l}}{\beta}) \mathbb{1}\{\Delta x_{i,l} \geq 0\}}{\Gamma(\alpha t_j^\eta - \alpha t_{j-1}^\eta) \beta^{\alpha t_j^\eta - \alpha t_{j-1}^\eta}} \Big].$$

Proof. Note that $\Delta y_{i,t_j} = \Delta x_{i,t_j} + \Delta z_{i,t_j}$. The component likelihood derivation is then as follows:

$$
\begin{aligned}
&f_{\boldsymbol{\theta}}(\mathbf{y_i}) = f_{\boldsymbol{\theta}}(y_{i,t_1}, \dots, y_{i,t_M}),\\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}} f_{\boldsymbol{\theta}}(y_{i,t_1}, \dots, y_{i,t_M}|A_i = a_i, \mathbf{z}_i = \mathbf{z}_i) f(a_i) f(\mathbf{z}_i) da_i d\mathbf{z}_i,\\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}} f_{\boldsymbol{\theta}}(y_{i,t_1} - y_{i,t_0}, \dots, y_{i,t_M} - y_{i,t_{M-1}}|A_i = a_i, \mathbf{z}_i = \mathbf{z}_i) f(a_i) f(\mathbf{z}_i) da_i d\mathbf{z}_i,\\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}} f_{\boldsymbol{\theta}}(\Delta x_{i,t_1} + \Delta z_{i,t_1}, \dots, \Delta x_{i,t_M} + \Delta z_{i,t_M}|A_i = a_i, \mathbf{z}_i = \mathbf{z}_i) f(a_i) f(\mathbf{z}_i) da_i d\mathbf{z}_i,\\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}} f_{\boldsymbol{\theta}}(\Delta x_{i,t_1}, \dots, \Delta x_{i,t_M}|A_i = a_i, \mathbf{z}_i = \mathbf{z}_i) f(a_i) f(\mathbf{z}_i) da_i d\mathbf{z}_i,\\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}} \prod_{t_j=t_1}^{t_M} f_{\boldsymbol{\theta}}(\Delta x_{i,l}|A_i = a_i, \mathbf{z}_i = \mathbf{z}_i) f(a_i) f(\mathbf{z}_i) da_i d\mathbf{z}_i,\\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}} \prod_{t_j=t_1}^{t_M} \frac{\Delta x_{i,l}^{(\alpha t_j^\eta - \alpha t_{j-1}^\eta - 1)} \exp(\frac{-\Delta x_{i,l}}{\beta}) \mathbb{1}\{\Delta x_{i,l} \geq 0\}}{\Gamma(\alpha t_j^\eta - \alpha t_{j-1}^\eta)\beta^{\alpha t_j^\eta - \alpha t_{j-1}^\eta}} f(a_i) f(\mathbf{z}_i) da_i d\mathbf{z}_i,\\
&= \mathbb{E}_{A_i, \mathbf{Z}_i}\Big[\prod_{t_j=t_1}^{t_M} \frac{\Delta x_{i,l}^{(\alpha t_j^\eta - \alpha t_{j-1}^\eta - 1)} \exp(\frac{-\Delta x_{i,l}}{\beta}) \mathbb{1}\{\Delta x_{i,l} \geq 0\}}{\Gamma(\alpha t_j^\eta - \alpha t_{j-1}^\eta)\beta^{\alpha t_j^\eta - \alpha t_{j-1}^\eta}} \Big].
\end{aligned}
$$

We have substituted the specific Gamma probability density function in the derivation above, thus that applies specifically to our model. □

Lemma 2. *Given Gamma Degradation Model* (2)*, the joint likelihood equals:*

$$
\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}) = \mathbb{E}_{A_i, \mathbf{Z}_i}\Big[\prod_{i=1}^{C} \prod_{t_j=t_1}^{t_M} \frac{\Delta x_{i,t_j}^{(\alpha t_j^\eta - \alpha t_{j-1}^\eta - 1)} \exp(\frac{-\Delta x_{i,t_j}}{\beta}) \mathbb{1}\{\Delta x_{i,t_j} \geq 0\}}{\Gamma(\alpha t_j^\eta - \alpha t_{j-1}^\eta)\beta^{\alpha t_j^\eta - \alpha t_{j-1}^\eta}} \Big].
$$

Proof. Given the component likelihood in Lemma 1, the joint likelihood can be derived as follows:

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}) &= \prod_{i=1}^{m} \mathcal{L}(\boldsymbol{\theta}; \mathbf{y}_i) = \prod_{i=1}^{C} f_{\boldsymbol{\theta}}(\mathbf{y_i})\\
&= \prod_{i=1}^{C} \mathbb{E}_{A_i, \mathbf{Z}_i}\Big[\prod_{t_j=t_1}^{t_M} \frac{\Delta x_{i,t_j}^{(\alpha t_j^\eta - \alpha t_{j-1}^\eta - 1)} \exp(\frac{-\Delta x_{i,t_j}}{\beta}) \mathbb{1}\{\Delta x_{i,t_j} \geq 0\}}{\Gamma(\alpha t_j^\eta - \alpha t_{j-1}^\eta)\beta^{\alpha t_j^\eta - \alpha t_{j-1}^\eta}} \Big],\\
&\overset{(*)}{=} \mathbb{E}_{A_i, \mathbf{Z}_i}\Big[\prod_{i=1}^{C} \prod_{t_j=t_1}^{t_M} \frac{\Delta x_{i,t_j}^{(\alpha t_j^\eta - \alpha t_{j-1}^\eta - 1)} \exp(\frac{-\Delta x_{i,t_j}}{\beta}) \mathbb{1}\{\Delta x_{i,t_j} \geq 0\}}{\Gamma(\alpha t_j^\eta - \alpha t_{j-1}^\eta)\beta^{\alpha t_j^\eta - \alpha t_{j-1}^\eta}} \Big].
\end{aligned}
$$

The equality $(*)$ holds because of the independence of the terms inside the expectation. Again, the probability density function of the Gamma is specific to our model, for other distributions of the true degradation other pdf's can be substituted. □

Remark 1.1. *The theoretical joint likelihood of 2 yields the following CMC estimator:*

$$
\hat{\mathcal{L}}(\boldsymbol{\theta}; \mathbf{y}) = \frac{1}{N_{\text{part}}} \sum_{n=1}^{N_{\text{part}}} \prod_{i=1}^{C} \prod_{t_j=t_1}^{t_M} \frac{(\Delta x_{i,t_j}^{j})^{(\alpha t_j^\eta - \alpha t_{j-1}^\eta - 1)} \exp(\frac{-\Delta x_{i,t_j}^{j}}{\beta}) \mathbb{1}\{\Delta x_{i,t_j}^{j} \geq 0\}}{\Gamma(\alpha t_j^\eta - \alpha t_{j-1}^\eta)\beta^{\alpha t_j^\eta - \alpha t_{j-1}^\eta}}.
$$

Lemma 3. *The likelihood estimator*

$$\hat{\mathcal{L}}(\boldsymbol{\theta};\mathbf{y}) = \frac{1}{N_{\text{part}}} \sum_{n=1}^{N_{\text{part}}} \prod_{i=1}^{C} \prod_{t_j=t_1}^{t_M} \frac{(\Delta x_{i,t_j}^{j})^{(\alpha t_j^{\eta} - \alpha t_{j-1}^{\eta} - 1)} \exp(\frac{-\Delta x_{i,t_j}^{j}}{\beta}) \mathbb{1}\{\Delta x_{i,t_j}^{j} \geq 0\}}{\Gamma(\alpha t_j^{\eta} - \alpha t_{j-1}^{\eta}) \beta^{\alpha t_j^{\eta} - \alpha t_{j-1}^{\eta}}}$$

is an unbiased estimator.

Proof. In the following derivation we show that this estimator is an unbiased estimator:

$$\begin{aligned}
\mathbb{E}\Big[\hat{\mathcal{L}}(\boldsymbol{\theta};\mathbf{y})\Big] &= \mathbb{E}\Big[\frac{1}{N_{\text{part}}} \sum_{n=1}^{N_{\text{part}}} \prod_{i=1}^{C} \prod_{t_j=t_1}^{t_M} \frac{(\Delta x_{i,t_j}^{n})^{(\alpha t_j^{\eta} - \alpha t_{j-1}^{\eta} - 1)} \exp(\frac{-\Delta x_{i,t_j}^{n}}{\beta}) \mathbb{1}\{\Delta x_{i,t_j}^{n} \geq 0\}}{\Gamma(\alpha t_j^{\eta} - \alpha t_{j-1}^{\eta}) \beta^{\alpha t_j^{\eta} - \alpha t_{j-1}^{\eta}}}\Big] \\
&= \frac{1}{N_{\text{part}}} \sum_{n=1}^{N_{\text{part}}} \mathbb{E}\Big[\prod_{i=1}^{C} \prod_{t_j=t_1}^{t_M} \frac{(\Delta x_{i,t_j}^{n})^{(\alpha t_j^{\eta} - \alpha t_{j-1}^{\eta} - 1)} \exp(\frac{-\Delta x_{i,t_j}^{n}}{\beta}) \mathbb{1}\{\Delta x_{i,t_j}^{n} \geq 0\}}{\Gamma(\alpha t_j^{\eta} - \alpha t_{j-1}^{\eta}) \beta^{\alpha t_j^{\eta} - \alpha t_{j-1}^{\eta}}}\Big] \\
&\overset{(*)}{=} \frac{1}{N_{\text{part}}} \sum_{n=1}^{N_{\text{part}}} \prod_{i=1}^{C} \prod_{t_j=t_1}^{t_M} \mathbb{E}\Big[\frac{(\Delta x_{i,t_j}^{n})^{(\alpha t_j^{\eta} - \alpha t_{j-1}^{\eta} - 1)} \exp(\frac{-\Delta x_{i,t_j}^{n}}{\beta}) \mathbb{1}\{\Delta x_{i,t_j}^{n} \geq 0\}}{\Gamma(\alpha t_j^{\eta} - \alpha t_{j-1}^{\eta}) \beta^{\alpha t_j^{\eta} - \alpha t_{j-1}^{\eta}}}\Big] \\
&= \frac{1}{N_{\text{part}}} \sum_{n=1}^{N_{\text{part}}} \prod_{i=1}^{C} \prod_{t_j=t_1}^{t_M} f_{\boldsymbol{\theta}}(y_{i,t_j}) \\
&= \frac{1}{N_{\text{part}}} \sum_{n=1}^{N_{\text{part}}} \prod_{i=1}^{C} f_{\boldsymbol{\theta}}(\mathbf{y_i}) \\
&= \frac{1}{N_{\text{part}}} \sum_{n=1}^{N_{\text{part}}} \prod_{i=1}^{m} \mathcal{L}(\boldsymbol{\theta};\mathbf{y}_i) \\
&= \frac{1}{N_{\text{part}}} \sum_{n=1}^{N_{\text{part}}} \mathcal{L}(\boldsymbol{\theta};\mathbf{y}) \\
&= \mathcal{L}(\boldsymbol{\theta};\mathbf{y})
\end{aligned}$$

where (*) is due to independence of the terms inside the expectation.

□

C SMC algorithm

The main idea behind SMC is to compute weights $w_{t_j,n}$ for all $\mathbf{x}_{t_j,n} + \mathbf{a}_n$, where $n \in \{1, \ldots, N_{\text{part}}\}$, for all time steps t_j, $j \in \{1, \ldots, M\}$, based on the likelihood of the values $\mathbf{z}_{t_j,n} = \mathbf{y}_{t_j} - \mathbf{x}_{t_j,n} - \mathbf{a}_n$. In the next time step, an index resampling takes place, where index n becomes $k(n)$ according to the weights $\mathbf{w}_{t_j}$. Subsequently, the new values $x_{i,t_{j+1},n} + a_{i,n}$ are computed as $x_{i,t_j,k(n)} + a_{i,k(n)} + \Delta x_{i,t_{j+1},n}$, where $\Delta x_{i,t_{j+1},n}$ is a number drawn from the Gamma distribution described in Gamma Degradation Model (2). Consequently, the algorithm progresses with values that are more likely [32].

Remark 1.2 (Vector notation in algorithms)**.** *To simplify the notation in Algorithm 1 and Algorithm 2, we sometimes use a vectorizerd operation. For example, on line 8 in Algorithm 1,*

$$\mathbf{W}_{t_0} = \frac{\mathbf{w}_{t_0}}{\sum_{h=1}^{N_{\text{part}}} w_{t_0,h}}$$

indicates that the operation is performed for each element of the vectors $\mathbf{W}_{t_j}$ *and* $w_{t_0,n}$*. Specifically, in this case, it holds that*

$$W_{t_0,n} = \frac{w_{t_0,n}}{\sum_{h=1}^{N_{\text{part}}} w_{t_0,h}} \text{ for } n = 1, \ldots, N.$$

The same vectorization logic applies to Algorithm 1 on lines 12, 14, 15, 24 and Algorithm 2 on lines 12, 14, 15, 17, and 25.

Algorithm 1: Sequential Monte Carlo, Generic

1 **for** *all runs* n **do**
2 **for** *all components* i **do**
3 Draw C from a distribution $M_{t_0}(\Delta d_{i,t_0})$
4 Initialize $D_{i,n}(t_0) = m$
5 **end**
6 Initialize $w_{t_0,n} = G_{t_0}(\mathbf{D}_n(t_0))$
7 **end**
8 Initialize $\mathbf{W}_{t_0} = \frac{\mathbf{w}_{t_0}}{\sum_{h=1}^{N_{\text{part}}} w_{t_0,h}}$
9 **for** *all time periods* t_j **do**
10 **if** $ESS_{t_{j-1}}(\mathbf{W}_{t_j-1}) = \frac{1}{\sum_{n=1}^{N_{\text{part}}} (W_{t_j-1,n})^2} < N/2 = ESS_{min}$ **then**
11 Draw a vector $\mathbf{k}$ with N_{part} elements drawn from $\{1, \ldots, N_{\text{part}}\}$ where number n has probability $W_{t_j-1,n}$ of being drawn
12 $\hat{\mathbf{w}}_{t_j-1} = \mathbf{1}$
13 **else**
14 $\mathbf{k} = [1, \ldots, N_{\text{part}}]$
15 $\hat{\mathbf{w}}_{t_j-1} = \mathbf{w}_{t_j-1}$
16 **end**
17 **for** *all runs* n **do**
18 **for** *all components* i **do**
19 Draw C from distribution $M_{t_j}(D_{i,k(n)}(t_{j-1}), \Delta d_{i,t_j})$
20 $D_{i,n}(t_j) = m$
21 **end**
22 $w_{t_j,n} = \hat{w}_{t_j-1,n} G_{t_j}(\mathbf{D}_{k(n)}(t_{j-1}), \mathbf{D}_n(t_j))$
23 **end**
24 $\mathbf{W}_{t_j} = \frac{\mathbf{w}_{t_j}}{\sum_{n=1}^{N_{\text{part}}} w_{t_j,n}}$
25 **end**

D Bootstrap algorithm

With these choices of M_{t_j} and G_{t_j}, the algorithm for Bootstrap is defined, which can be observed in Algorithm 2. Bootstrap is intuitively quite easy to understand, which might also be a reason why

it is a commonly used filter. Intuitively namely, the particles are a Markov Chain $\{\mathbf{D}_n(t_j)\}$ where if resampling occurs, one draws a chain from all runs $n \in \{1, \ldots, N_{\text{part}}\}$ based on their probabilities of being given the observed degradation data. Subsequently, one continues with this drawn chain in run n and extends it with a new value $\mathbf{D}_n(t_{j+1})$ by drawing an increase in degradation from $\mathbf{D}_n(t_j)$ from the Gamma distribution. Hence, one keeps continuing with the most likely chains $\{\mathbf{D}_n(t_j)\}$ given the observed degradation data $(\mathbf{y}_1, \ldots, \mathbf{y}_C)$.

Algorithm 2: Bootstrap Filter

for *all runs* n **do**
  for *all components* i **do**
    Draw C from $N(\mu_A, \sigma_A^2)$
    Initialize $D_{i,n}(t_0) = m$
  end
  Initialize $w_{t_0,n} = 1$
  Initialize $W_{t_0,n} = \frac{1}{N_{\text{part}}}$
end
for *all time periods* t_j **do**
  if $ESS_{t_j-1}(\mathbf{W}_{t_j-1}) = \frac{1}{\sum_{n=1}^{N_{\text{part}}}(W_{t_j-1,n})^2} < N/2 = ESS_{min}$ **then**
    Draw a vector $\mathbf{k}$ with N_{part} elements drawn from $\{1, \ldots, N_{\text{part}}\}$ where number n has probability $W_{t_j-1,n}$ of being drawn
    $\hat{\mathbf{w}}_{t_j-1} = \mathbf{1}$
  else
    $\mathbf{k} = [1, \ldots, N_{\text{part}}]$
    $\hat{\mathbf{w}}_{t_j-1} = \mathbf{w}_{t_j-1}$
  end
  $\mathbf{w}_{t_j} = \hat{\mathbf{w}}_{t_j-1}$
  for *all runs* n **do**
    for *all components* i **do**
      Draw Δx from Gamma$(\alpha t_j^{\eta} - \alpha t_j - 1^{\eta}, \beta)$
      $D_{i,n}(t_j) = \Delta x + D_{i,k(n)}(t_{j-1})$
      $w_{t_j,n} = w_{t_j,n} \frac{1}{\sigma_Z \sqrt{2\pi}} \exp\big(-\frac{(y_{i,t_j} - D_{i,n}(t_j))^2}{2\sigma_Z^2}\big)$
    end
  end
  $\mathbf{W}_{t_j} = \frac{\mathbf{w}_{t_j}}{\sum_{n=1}^{N_{\text{part}}} w_{t_j,n}}$
end

E Real Degradation Data

Batch	0 years	1 year	2 years	3 years
1	105	104	101	98
2	106	102	99	96
3	103	101	98	95
4	105	101	99	95
5	104	102	100	96
6	102	100	100	97
7	104	103	101	97
8	105	104	101	100
9	103	101	99	99
10	103	102	97	96
11	101	98	93	91
12	105	102	100	98
13	105	104	99	95
14	104	103	97	94
15	105	103	98	96
16	103	101	99	96
17	104	102	101	98
18	106	104	102	97
19	105	103	100	99
20	103	101	99	95
21	101	101	94	90
22	102	100	99	96
23	103	101	99	94
24	105	104	100	97

Table 11: Drug Potency Degradation Data, percentage of claimed potency.