Sequential ensemble Monte Carlo sampler for on-line Bayesian inference of time-varying parameter In engineering applications

Lye, Adolphus; Marino, Luca; Cicirello, Alice; Patelli, Edoardo

DOI
10.1115/1.4056934

Publication date
2023

Document Version
Final published version

Published in
ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering

Adolphus Lye
Institute for Risk and Uncertainty,
University of Liverpool,
Chadwick Building, Peach Street,
Liverpool L69 7ZF, UK

Luca Marino
Department of Engineering Science,
University of Oxford,
Parks Road,
Oxford OX1 3PJ, UK

Alice Cicirello
Faculty of Civil Engineering and Geoscience,
Delft University of Technology,
Stevinweg 1,
Delft 2628 CN, Netherlands;
Department of Engineering Science,
University of Oxford,
Parks Road,
Oxford OX1 3PJ, UK;
Institute for Risk and Uncertainty,
University of Liverpool,
Chadwick Building,
Peach Street,
Liverpool L69 7ZF, UK

Edoardo Patelli[1]
Centre for Intelligent Infrastructure,
Department of Civil and
Environmental Engineering,
University of Strathclyde,
James Weir Building,
75 Montrose Street,
Glasgow G1 1XJ, UK
e-mail: edoardo.patelli@strath.ac.uk

Sequential Ensemble Monte Carlo Sampler for On-Line Bayesian Inference of Time-Varying Parameter in Engineering Applications

Several on-line identification approaches have been proposed to identify parameters and evolution models of engineering systems and structures when sequential datasets are available via Bayesian inference. In this work, a robust and "tune-free" sampler is proposed to extend one of the sequential Monte Carlo implementations for the identification of time-varying parameters which can be assumed constant within each set of data collected but might vary across different sequences of datasets. The proposed approach involves the implementation of the affine-invariant Ensemble sampler in place of the Metropolis–Hastings sampler to update the samples. An adaptive-tuning algorithm is also proposed to automatically tune the step-size of the affine-invariant ensemble sampler which, in turn, controls the acceptance rate of the samples across iterations. Furthermore, a numerical investigation behind the existence of inherent lower and upper bounds on the acceptance rate, making the algorithm robust by design, is also conducted. The proposed method allows for the off-line and on-line identification of the most probable models under uncertainty. The proposed sampling strategy is first verified against the existing sequential Monte Carlo sampler in a numerical example. Then, it is validated by identifying the time-varying parameters and the most probable model of a nonlinear dynamical system using experimental data. [DOI: 10.1115/1.4056934]

Keywords: sequence Monte Carlo, model updating, affine-invariant ensemble sampler, time-varying parameter

1 Introduction

In recent years, on-line learning has garnered significant attention for the purpose of parameter identification of engineering systems. On-line parameter identification involves the learning and estimation of the parameter(s) of interest through distinct datasets which are obtained sequentially, contrary to the batch learning approach which requires the availability of the entire dataset to produce estimates [1]. In particular, on-line parameter identification is of value in situations when data is obtained over a period of time. This allows for real-time parameter identification, making it a practical approach for investigating the performance of engineering systems under operating conditions. For example, on-line parameter identification has been applied to identify: moveable mass positions within a 2DoF shear frame [2,3]; mistuning parameters of rotating blisks [4]; modal parameters of a vehicle motion modes to investigate its dominant motion-mode [5]; structural parameters of a nonlinear structural system to update its dynamical response model [6]; terrain parameters of the Wheel-Terrain model for wheeled motion control of mobile robots [7]; and structural parameters of smart building structures for real-time damage detection [8].

The parameters identified through on-line learning are often time-invariant, see, e.g., Refs. [4,5], and [9]. However, in many engineering applications, these parameters vary with time [10] to which examples include: structural modal parameters to study the dynamic response of structures [11,12]; fatigue cracking parameters [13]; stress data for performance prediction of steel bridges [14]; localized impact damage in composite panels based on sensor data [15]; and earthquake ground motion parameters using generalized Kalman filter and structural absolute accelerations data [16].

Currently, in time-domain applications, system parameters and system states such as displacements and velocities, are estimated via sequential Bayesian inference using Kalman filters [17,18], Gaussian Sum filters [19,20], and particle filters [21,22]. Kalman filters are computationally less-expensive compared to other filtering techniques. However, they are mostly designed to deal with problems involving a dynamical system identification with a linear state-space and Gaussian "noise" [23]. Extended Kalman filter [24,25] and unscented Kalman filter [6,18] extended the approach to nonlinear state-space and non-Gaussian noise. For instance, Gaussian sum filters utilize weighted Gaussian Models to approximate the predictive and posterior PDFs [19]. Thus, it does not require the analytical form of the aforementioned PDFs. However, like the Kalman filter, Gaussian sum filters become ineffective when the state-space setup becomes highly nonlinear [26]. Particle filters are applicable in both linear and nonlinear state-space setup and do not assume the form of noise [27]. However, they are computationally expensive and perform poorly even with moderately high number of parameters (i.e., above 18 dimensions) [28,29]. Despite this, particle filters are more general and robust in their implementation [27], since the error of the estimates is reducible by obtaining more samples. For more information pertaining to the above approaches, the reader is referred to the recent review paper by Ref. [30].

The sequential Monte Carlo (SMC) sampler, whose concept is based on the particle filter [31,32], is a popular technique when addressing Bayesian inference problems [33]. Currently, it has

[1] Corresponding author.

Manuscript received October 21, 2021; final manuscript received February 12, 2023; published online March 24, 2023. Assoc. Editor: Ioannis A. Kougioumtzoglou.

been implemented under off-line (i.e., batch) and on-line settings toward the parameter identification for numerous problems involving: (1) nonlinear time-series model; (2) nonlinear state-space models; and (3) high-dimensional target distributions [34,35]. At present, the sequential Monte Carlo sampler has not yet been implemented to identify model parameters that are considered constant between observation intervals but might vary across the different observation sequences. Under such settings, the posteriors of such parameters are assumed to be constant between observations while their posteriors (and therefore their estimates) may vary between the different sequences of observations. Moreover, there exist the following short-comings in the SMC algorithm: (1) the choice of proposal distribution can significantly affect the sampler results [32]; (2) it is computationally inefficient because of the number of parameters required to compute and tune [36]; and (3) the moderation of acceptance rates of its samples has not been investigated in detail.

To address these short-comings, a robust and "tune-free" sequential ensemble Monte Carlo (SEMC) sampler is proposed based on the use of the affine-invariant ensemble sampler (AIES) in place of the Metropolis–Hastings (MH) sampler for the MCMC step. AIES has been proved robust in recent implementations for reliability analysis using subset simulation [37] and for model updating via transitional ensemble Monte Carlo sampler [2] thanks to its capability of sampling from highly-skewed and anisotropic distributions (see Sec. 3). In addition, an adaptive-tuning algorithm, inspired from the work by Ref. [38], is developed to provide a robust mechanism to ensure that the acceptance rate values have achieved convergence before moving on to the next sampling iteration. Details on the adaptive-tuning algorithm are provided in Sec. 3.1. Numerical investigation has proven the existence of inherent bounds in the acceptance rates values and shown in Sec. 4. To the best of the authors' knowledge, such investigation and analysis are yet to be presented in existing literature.

The proposed SEMC sampler is first implemented alongside the traditional SMC sampler in a numerical example involving a spring-mass-damper system with two time-varying parameters to which details are provided in Sec. 5. This allows to compare and verify the inference results and to highlight the advantages of the proposed sampler over the traditional SMC sampler.

The proposed sampler is then validated through an experimental example involving a single-storey structure with a Coulomb friction contact whose properties are different during each observation interval under harmonic loading as shown in Sec. 6. Such application example is specifically chosen due to its importance in assessing the dynamic performance of structures and avoiding friction-related failures [39,40]. This example allows: (1) to validate the performance of the SEMC sampler in inferring both time-varying and time-invariant parameters with experimental data; and (2) to demonstrate the capabilities and robustness of the algorithm in identifying the most probable Markov kernel under uncertainty.

2 Review of Sequential Monte Carlo

Sequential Bayesian filtering is a popular technique to address inverse problems and infer model parameters under uncertainty thanks to its ability to combine and update prior knowledge with a sequence of observed data $\boldsymbol{D}$ [41,42]. This yields the posterior $P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$ from which samples are obtained to provide numerical estimates of the inferred parameter(s). One such sampling technique commonly used in literature is the SMC sampler [43,44].

2.1 Advantages and Limitations of Current SMC Sampler Implementations. One key advantage of the SMC sampler lies in its ability to sequentially compute the evidence term $P(\boldsymbol{D}^{1:s}|M)$ corresponding to the posterior $P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$ at any given sth time-step sequence [32,44]. The metric $P(\boldsymbol{D}^{1:s}|M)$ also quantifies how well a given model class M describes the available set of data $\boldsymbol{D}^{1:s}$ as well as the time-evolution of $\boldsymbol{\theta}$. This makes the SMC sampler well-suited to addressing problems regarding the model class selection of M (see, e.g., Refs. [44–47], and [48]). $P(\boldsymbol{D}^{1:s}|M)$ can be estimated by the product of the mean of the nominal weights w_i^s at any given time-step sequence $s \geq 1$ [32]

$$P\left(\boldsymbol{D}^{1:s}|M\right) \approx \frac{1}{N}\prod_{m=1}^{s}\sum_{i=1}^{N} w_i^m \qquad (1)$$

Another characteristic of the SMC sampler is the flexibility in the choice of auxiliary parameters of the algorithm such as the scaling parameter of the covariance matrix and the Markov kernel of the time-varying parameter (i.e., see Eq. (A2)) [49,50]. This is because even if the samples do not follow the true distribution, the weighting process and the conditional resampling step (i.e., the Bootstrapping with replacement), with which the algorithm will correct and move the samples closer to their true distribution. To illustrate the resampling concept, a numerical example involving a mixture of two bivariate Gaussian distributions based on work in Ref. [51] is presented. Here, we shall consider a two-dimensional uniform prior such that: $P(\theta_1) \sim U(-2,2)$ and $P(\theta_2) \sim U(-2,2)$. The likelihood function is defined as a mixture of two bivariate Gaussian functions with means centered about $\{\theta_1, \theta_2\} = \{0.5, 0.5\}$ and $\{-0.5, -0.5\}$ with covariance matrix $0.1 \cdot \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The standard SMC sampler is implemented with $N = 10{,}000$ samples and the prior samples with their associated normalized weights $\hat{w}_i^s$ are presented in Fig. 1. In the figure, the samples which fall in the regions of the posterior indicated in yellow and light blue are resampled with a higher probability than those which fall outside the regions of the posterior indicated by the darker shaded regions. This allows for the Markov chains in the MCMC step to initiate within the support of the posterior and removes the need to consider the burn-in.

However, there are still limits to such flexibility as situations could arise whereby nearly all the samples could still fall outside the true distribution even after the conditional resampling step. This leads to nominal weights being close to 0 and the resampling step would fail to converge the samples to the true distribution [52].

One key problem in the SMC sampler is that there exists no universal choice of proposal distribution $q(\boldsymbol{\theta}_i^{s*}|\boldsymbol{\theta}_i^s)$ to generate candidate samples $\boldsymbol{\theta}_i^{s*}$ [32]. This creates a significant degree of model uncertainty in deciding an appropriate distribution for $q(\boldsymbol{\theta}_i^{s*}|\boldsymbol{\theta}_i^s)$. In practice, a Normal distribution is used to ease the computation of the acceptance ratio α_i due to the symmetric nature of the distribution [53,54]. However, the ideal choice of $q(\boldsymbol{\theta}_i^{s*}|\boldsymbol{\theta}_i^s)$ is one that is "optimally" scaled to the current posterior $P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$ [32,54,55]. Practically speaking, this would be difficult for two reasons: (1) there is a lack of knowledge over the analytical form of the true posterior itself; and (2) it is difficult to determine that "optimal" scale, especially in the case where the random-walk MH algorithm is used as the MCMC kernel, although the problem can be addressed through the use of the independent MH MCMC kernel in Ref. [31] which eliminates the need for parameter tuning.

Another challenge is the computational cost of the SMC sampler. In particular, when the dimensionality of the problem increases, a higher computation cost is incurred in computing the covariance matrix of $q(\boldsymbol{\theta}_i^{s*}|\boldsymbol{\theta}_i^s)$ at each sth time-step sequence, which can be the case for both the independent MH [31] and random-walk MH algorithms.

2.2 Sequential Monte Carlo Variants. Numerous MCMC move kernels within the SMC sampler have been considered such as: (1) particle evolution metropolis (i.e., PEM-SMC) [33,56], (2) Gibbs sampler, i.e., SMC for vector autoregressions with stochastic volatility (VAR-SV) [57], and (3) target-invariant MCMC mutation kernel, i.e., SMC for high-dimensional inverse problems [58].

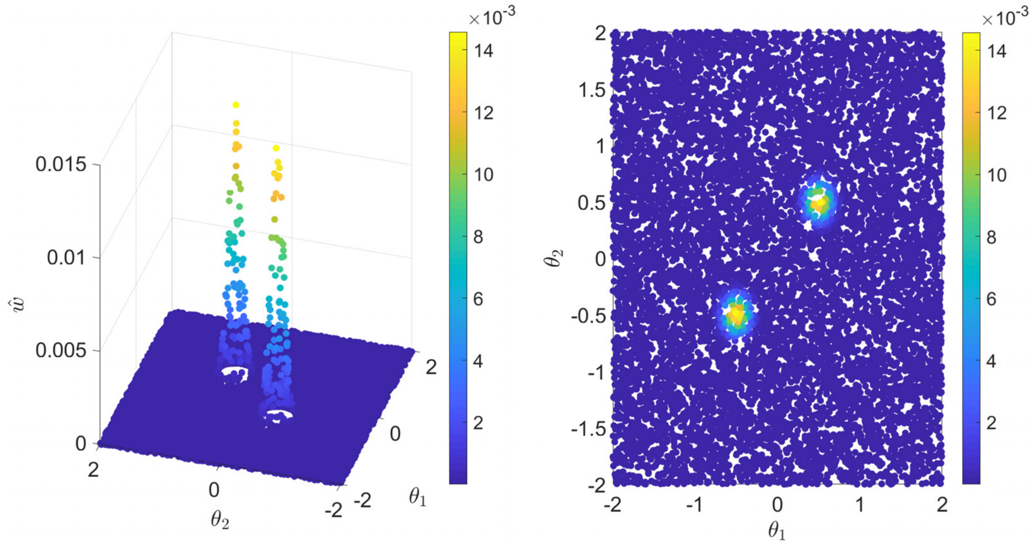

Fig. 1 Scatterplot of the prior samples, along with their associated normalized weights $\hat{w}_i^s$, obtained from the posterior consisting of a mixture of two Gaussian distributions

The PEM-SMC sampler has demonstrated its strength in generating samplers more effectively from complex-shaped distributions, especially those with multiple peaks [33]. The sampler is also able to sample efficiently from moderate-dimensional posteriors (i.e., up to 30 dimensions) thanks to its effective way to explore the dimensional sample space and generate more candidate samples with a high probability content. This ensures a quick convergence of the samples toward the posterior distribution [33]. However, the effectiveness of the algorithm is subjected to the choice of the proposal distribution (i.e., the transfer probability distribution) which leads to the potential problem of model uncertainty.

The SMC sampler for VAR-SV has demonstrated its strength in tackling the problem of degeneracy effectively through the use of the Gibbs sampler which ensures that there are less repeated samples generated in the MCMC step. This increases the number of unique samples, thereby, allowing the sample space defined by the posterior to be well-explored [57]. In addition, it is highly parallelizable which allows for the rapid update of samples from one posterior to the next across iterations [57]. However, the sampler is limited by the short-coming of the Gibbs sampler which is the latter's dependency on the choice of an appropriate conditional distribution to represent the posterior [59]. Should the posterior be functionally complex, choosing such conditional distribution becomes nontrivial [29].

The target-invariant MCMC mutation kernel is also quite robust thanks to its ability to address the issue of sample degeneracy effectively like the SMC for VAR-SV [58]. In addition, the algorithm ensures a relatively quick convergence rate of the samples to the posterior distribution through the adaptive tempering step which ensures a smooth transition from one posterior to the next between successive iterations [58]. This allows for the sampler to be applicable to cases with highly dimensional and complex-shaped posteriors. However, the use of such MCMC move kernel introduces a relatively large number of auxiliary parameters to compute such as the tempering parameter and the Fourier coefficients which increases the computational cost of the algorithm [58].

For the work presented here, the AIES is implemented as the alternative MCMC kernel to the MH. The reasons behind such proposal are: (1) the ability to sample from anisotropic and highly-skewed distributions; (2) can be parallelized; and 3) uses a reduced number of tuning parameters. This gives rise to the SEMC sampler to which an additional feature proposed is the adaptive tuning algorithm which serves to control the acceptance rate of the sampler by tuning its step-size parameter in an automatic manner.

3 Sequential Ensemble Monte Carlo Sampler

The AIES sampler is an MCMC sampling technique endowed with the affine-invariance property which was recently developed by Goodman and Weare [60]. Such property involves an affine-transformation operation ψ which is defined by an invertible linear mapping from a $\mathbb{R}^{N_d}$ to $\mathbb{R}^{N_d}$ space [61]

$$\psi(\boldsymbol{\theta}) : \boldsymbol{\Theta} = \hat{A}\boldsymbol{\theta} + \boldsymbol{b} \tag{2}$$

where $\boldsymbol{\Theta}$ denotes $\boldsymbol{\theta}$ in the affine-transformed space, $\hat{A}$ is the N_d-by-N_d nonsingular transformation matrix and $\boldsymbol{b}$ is the N_d-by-1 translation vector. Defining $P'(\boldsymbol{\Theta}|\boldsymbol{D})$ as the posterior distribution of $\boldsymbol{\Theta}$ where P' denotes the distribution function in the $\boldsymbol{\Theta}$-space, affine-invariance exists between $\boldsymbol{\Theta}$-space and $\boldsymbol{\theta}$-space if the following condition holds [62]:

$$P'(\boldsymbol{\Theta}|\boldsymbol{D}) = P'(\psi(\boldsymbol{\theta})|\boldsymbol{D}) \propto P(\boldsymbol{\theta}|\boldsymbol{D}) \tag{3}$$

A sampler is affine-invariant if it possesses an MCMC move kernel whose proposal distribution is also affine-invariant such that

$$q'(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}_i) = q'(\psi(\boldsymbol{\theta}^*)|\psi(\boldsymbol{\theta}_i)) \propto q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i) \tag{4}$$

whereby q' is the proposal distribution function in the $\boldsymbol{\Theta}$-space. When this condition is satisfied, the probability of generating a sample $\boldsymbol{\Theta}^*$ given $\boldsymbol{\Theta}_i$ in the transformed $\boldsymbol{\Theta}$-space now becomes equal to that of generating a sample $\boldsymbol{\theta}^*$ given $\boldsymbol{\theta}_i$ in the original $\boldsymbol{\theta}$-space [60].

An ensemble $\boldsymbol{\Theta}_i$ is defined as a collection of N_c Markov chains such that:

$\boldsymbol{\Theta}_i = \{\boldsymbol{\theta}_{1,i}, \boldsymbol{\theta}_{2,i}, \ldots, \boldsymbol{\theta}_{N_c-1,i}, \boldsymbol{\theta}_{N_c,i}\}$. In practice, N_c should be at least twice the dimensionality of $\boldsymbol{\theta}$ (i.e., $N_c \geq 2 \times N_d$) [60].

The AIES algorithm is initialized by N_c distinct Markov chains where each chain generates only one sample from the prior. This produces the first ensemble $\boldsymbol{\Theta}_i$ for $i = 1$. Next, the samples are updated one at a time in a sequential manner. To update the kth chain (for $k = 1, \ldots, N_c$), a sample from a complementary chain is chosen randomly from the set $\boldsymbol{\theta}_{[k],i} \sim \{\boldsymbol{\theta}_{1,i+1}, \ldots, \boldsymbol{\theta}_{k-1,i+1}, \boldsymbol{\theta}_{k+1,i}, \ldots, \boldsymbol{\theta}_{N_c,i}\}$. The affine-invariant stretch-move kernel is used to generate the candidate sample $\boldsymbol{\theta}_{k,i}^*$ [60,63]

$$\boldsymbol{\theta}_{k,i}^* = \boldsymbol{\theta}_{[k],i} + \lambda \cdot (\boldsymbol{\theta}_{k,i} - \boldsymbol{\theta}_{[k],i}) \tag{5}$$

Algorithm 1 AIES sampler algorithm

```
procedure (Generate $N$ samples from $P(\boldsymbol{\theta}|\boldsymbol{D},M)$)
  Define $N_c$ chains: $\boldsymbol{\theta}_1 = \{\boldsymbol{\theta}_{1,1}, \boldsymbol{\theta}_{2,1}, \ldots, \boldsymbol{\theta}_{N_c-1,1}, \boldsymbol{\theta}_{N_c,1}\}$   ▷ Initiate chains
  for $i = 1 : N-1$ do
    for $k = 1 : N_c$ do   ▷ Update $k$th chain
      Sample $\boldsymbol{\theta}_{[k],i}$ from the complementary set
      Sample: $\lambda \sim g(\lambda)$
      Generate $\boldsymbol{\theta}^*_{k,i}$ using Eq. (5)
      Calculate acceptance probability $\alpha_{\mathrm{AIES}}$ using Eq. (8)
      Sample: $r \sim U[0,1]$
      if $\alpha_k > r$ then   ▷ Accept/reject step
        Set $\boldsymbol{\theta}_{k,i+1} = \boldsymbol{\theta}^*_{k,i}$
      else
        Set $\boldsymbol{\theta}_{k,i+1} = \boldsymbol{\theta}_{k,i}$
      end if
    end for
  end for
end procedure
```

whereby λ is a real-valued scalar proposal stretch factor which is also a random variable following a proposal distribution $g(\lambda)$ defined as

$$g(\lambda) = \begin{cases} \dfrac{1}{2 \cdot \left(\sqrt{u} - \dfrac{1}{\sqrt{u}}\right)} \cdot \dfrac{1}{\sqrt{\lambda}} & \text{if } \lambda \in \left[\dfrac{1}{u}, u\right] \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

and $u \geq 1$ is the user-defined step-size of the AIES sampler. The form of $g(\lambda)$ in Eq. (6) is chosen such that it has a symmetric property [60]

$$g\left(\frac{1}{\lambda}\right) = \lambda \cdot g(\lambda) \tag{7}$$

The candidate sample $\boldsymbol{\theta}^*_{k,i}$ is accepted with probability α_k

$$\alpha_k = \min\left[1, \lambda^{N_d - 1} \cdot \frac{P\left(\boldsymbol{\theta}^*_{k,i}|\boldsymbol{D},M\right)}{P\left(\boldsymbol{\theta}_{k,i}|\boldsymbol{D},M\right)}\right] \tag{8}$$

This updating procedure is then repeated for the N_c chains and then the next sample is processed (i.e., set $i = i+1$ until $i = N$). To summarize the procedure, a pseudo-algorithm is presented in Algorithm 1.

The AIES sampler presents two key advantages over the MH sampler: it is able to sample from poorly scaled and highly-skewed distributions just as effectively and efficiently as it would from a well-scaled affine-transformed distribution [60,64] due to the use of the affine-invariant stretch-move kernel; and it does not require a user-defined proposal distribution $q(\boldsymbol{\theta}^*_i|\boldsymbol{\theta}_i)$ to generate candidate samples. It needs to be noted that what is described in point 2 can be achieved by the preconditioned Crank–Nicolson (pCN) MCMC algorithm recently implemented in Ref. [65]. This, in turn, removes the need for the proposed SEMC sampler to compute the covariance matrix for $q(\boldsymbol{\theta}^*_i|\boldsymbol{\theta}_i)$ and the algorithm only needs to iteratively update the scalar step-size u, whose computational cost is independent of the dimensionality of $\boldsymbol{\theta}$. The procedure to update step-size u will be presented in Sec. 3.1.

3.1 Adaptive-Tuning Algorithm.

The proposed adaptive-tuning algorithm is based on the work by Ref. [38] which serves two key purposes: (1) to adaptively tune and update the step-size u; and (2) to provide a mechanism to control the acceptance rates of the SEMC sampler such that they converge toward the user-defined target acceptance rates and fall within the optimal bounds of $[0.15, 0.50]$ suggested by Ref. [66].

The procedure undertaken by the algorithm is as follows: At $s = 1$, an initial step-size $u^{s=1}$ is defined by the user to which the recommended value is 2 [63,67]. From this initial value, the nominal step-size u_{nom} is computed after the MCMC step

$$u_{\mathrm{nom}} = u^s \cdot \exp\left[\alpha^s - \alpha_{\mathrm{tr}}\right] \tag{9}$$

where α^s is the acceptance rate for the current time-step sequence s and α_{tr} is the target acceptance rate defined as [66]

$$\alpha_{\mathrm{tr}} = \frac{0.21}{N_d} + 0.23 \tag{10}$$

If $u_{\mathrm{nom}} > 1$, then $u^{s+1} = u_{\mathrm{nom}}$. Otherwise, the algorithm sets $u^{s+1} = 1.01$ to ensure that the step-size would never be less than 1. This procedure is then repeated at the end of each iteration until the terminal iteration $s = s_{\mathrm{end}}$.

To provide additional robustness to the SEMC sampler and assuring the acceptance rates converge toward the target acceptance value at every iteration, "virtual" iterations are introduced which involve the updating of the posterior samples with a series of repeated data $\boldsymbol{D}^s$. The termination criteria are defined whereby the acceptance rate values have converged. This is indicated when the difference in the acceptance rate values Δ_α between successive "virtual" iterations j falls within, e.g., 10% of the value of α_{tr} upon which the procedure ends. In doing so, it not only allows for the automatic tuning of the step-size u^s but also ensures that the acceptance rate values converge toward α_{tr} across all s, independent of the data provided.

4 Acceptance Rates Analysis

The proposed SEMC sampler bounds the acceptance rate of the algorithms as defined by Eq. (10). While this is used as a general reference for the AIES, there exists a lower and upper bound on the acceptance rate based on the analysis done in the context of the experimental investigation to which details are presented in Sec. 6. A numerical investigation into the acceptance rate evolution by the SEMC sampler, given the Markov kernel T_1 is done for target acceptance rates of $\alpha_{tr} = \{0.100, 0.283, 0.440, 0.800, 0.900, 1.000\}$ with the respective starting step-size values $u^{s=1} = \{40, 40, 8, 2, 2, 2\}$ and presented in Fig. 2. It needs to be highlighted that the acceptance rate can be controlled by controlling the step-size u^s [60]. However, the numerical experiment shows that the acceptance rates never fall below 0.300, although the chosen target acceptance rate values were 0.100 and 0.283. The reason behind the existence of such lower bound is attributed to the resampling procedure by the SEMC algorithm which ensures that samples far from the posterior are eliminated while samples closer to or within the posterior are repopulated. This ensures that a proportion of the samples will always be accepted, resulting in a nonzero acceptance rate.

On the other hand, the upper bound of the acceptance rate is in practice always less than 1 given that for a nonuniform distribution a proportion of samples are rejected by the sampling algorithm. In fact, in the cases where the chosen target acceptance rate values were 0.900 and 1.000, the acceptance rates across the time-step sequences s never exceeded 0.850 as seen in Fig. 2. Hence, the existence of such bounds makes the algorithm particularly robust and generally applicable without the need of guessing a "good" acceptance rate. Instead, the acceptance rate can be used as a monitoring parameter of the performance of the algorithm. In fact, if the acceptance rate is high (e.g., above 0.850), it could indicate that the samples are stuck in a specific region of the posterior and the resulting sample distribution and its estimates may not be representative of the true posterior distribution. The acceptance rate bounds may depend on the dimension of the posterior

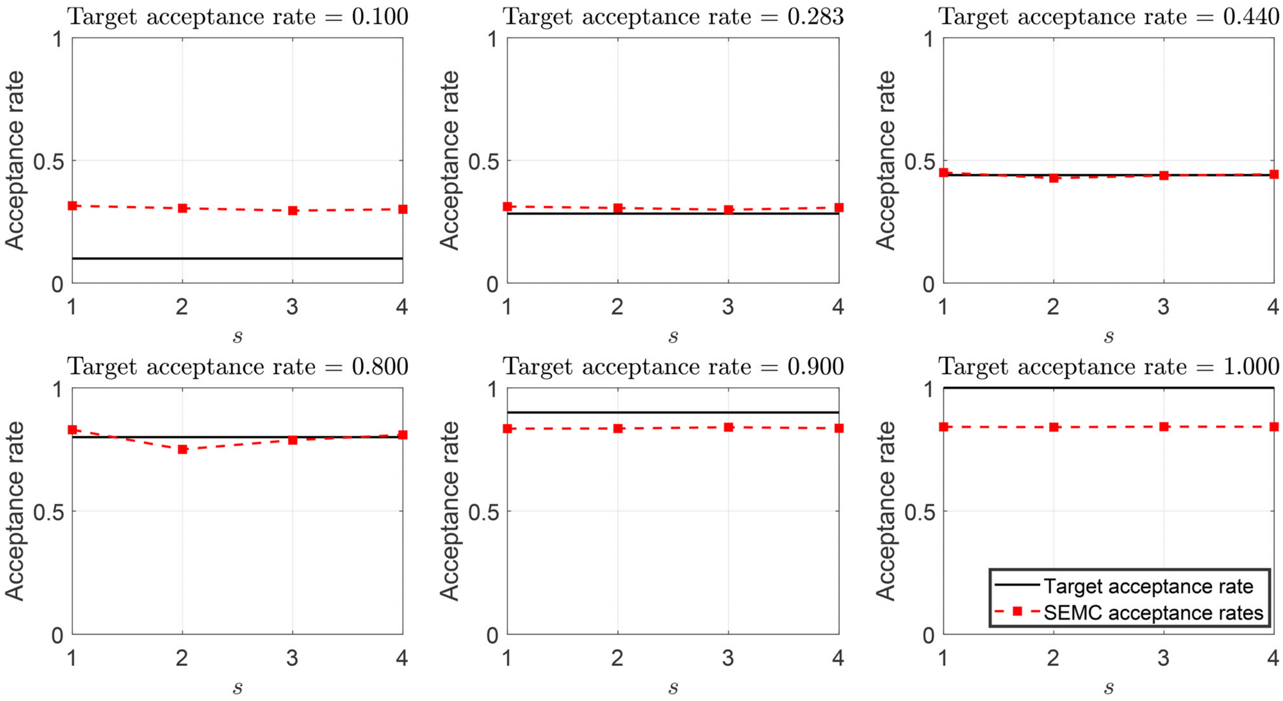

Fig. 2 Acceptance rates for the SEMC sampler: target acceptance rate values $\alpha_{tr} = \{0.100, 0.283, 0.440, 0.800, 0.900, 1.000\}$ and starting step-size values $u^{s=1} = \{40, 40, 8, 2, 2, 2\}$

distribution and the number of chains N_c used in the sampler which requires further investigation.

Without the need to decide on a "good" acceptance rate, complemented with the proposed adaptive-tuning algorithm involving the "virtual" iterations, these features highlight the robustness of the proposed SEMC sampler in that: (1) it is "tune-free" for the users in that they do not have to define an initial step-size value $u^{s=1}$; and (2) it is able to effectively control the acceptance rates across the different time-step sequence s as illustrated from the results in Fig. 2.

Algorithm 2 presents a summary of the SEMC sampling procedure. The resampling step in line 10 ensures that (1) the Markov chains initiate with a higher probability from samples $\boldsymbol{\theta}_i^s$ with a higher w_i^s [51]; and (2) the final distribution of samples would be representative of the analytical distribution of $P(\boldsymbol{\theta}_i^s|D^{1:s}, M)$.

5 Numerical Example: A Spring-Mass-Damper System

In this example, a single degree-of-freedom (SDoF) spring-mass-damper system consisting of a block with mass $m = 0.30\,\text{kg}$ attached to a spring with stiffness k, and a damper with viscous damping coefficient c is studied. The values of k and c decrease with time as described by a random degradation process. The system's equation of motion can be described following:

Algorithm 2 Proposed SEMC sampler algorithm

```
1: procedure (generate $N$ samples from $P(\boldsymbol{\theta}^j|\boldsymbol{D}^{1:j}, M)$)
2:    Set $s = 0$                                          ▷ initialize time step counter
3:    Draw initial $N$ sample set: $\boldsymbol{\theta}_i^{s+1} \sim P(\boldsymbol{\theta}|M)$    ▷ generate samples from prior
4:    Set $u^{s+1} = 2$                                    ▷ set initial value of step-size
5:    while $s < s_{\text{end}}$ do                        ▷ loop over time steps
6:       Set $s = s + 1$
7:       Set $j = 1$ & $\alpha_{\text{old}} = \alpha_{tr}$                ▷ initialize parameters
8:       Compute $\hat{w}_i^s$ using Eq. (A5)
9:       while do                                          ▷ initiate "virtual" loop
10:          Resample $N$ samples: $\boldsymbol{\theta}_i^s \sim \hat{w}_i^s$
11:          Set $\boldsymbol{\theta}_i^s = \boldsymbol{\theta}_{i,1}$ in ensemble $\boldsymbol{\theta}_1$           ▷ Initiate ensemble
12:          Update $\boldsymbol{\theta}_1$ with 1 iteration of AIES (see Algorithm 1)  ▷ MCMC step
13:          Compute $\alpha$ using Eq. (8)
14:          Compute $u_{\text{nom}}$ using Eq. (9)                 ▷ tuning the step-size
15:          Set $u^{s+1} = \max(u_{\text{nom}}, 1.01)$
16:          if $|\alpha - \alpha_{\text{old}}| < 0.1 \cdot \alpha_{\text{tr}}$ then         ▷ check termination criteria
17:             Break                                      ▷ exit "virtual" loop
18:          end if
19:          Set $\alpha_{\text{old}} = \alpha$
20:          Set $j = j + 1$
21:       end while                                        ▷ end "virtual" loop
22:       Set updated ensemble $\boldsymbol{\theta}_1$ as $\boldsymbol{\theta}_i^s \sim P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$
23:       Compute $\boldsymbol{\theta}_i^{s+1}$ using $T(\boldsymbol{\theta}^{s+1}|\boldsymbol{\theta}^s)$          ▷ set as new prior samples
24:       Compute PDF of $P(\boldsymbol{\theta}_i^{s+1}|D^{1:s})$ using Eq. (A9)    ▷ set as new prior PDF
25:       Compute $P(\boldsymbol{D}^{1:s}|M)$ using Eq. (1)
26:    end while
27: end procedure
```

$$m \cdot \frac{d^2x}{d\tau^2} + c(t) \cdot \frac{dx}{d\tau} + k(t) \cdot x = 0 \tag{11}$$

where x denotes the instantaneous displacement of the mass from its rest position, τ denotes the time variable (in seconds) of the oscillation, while t denotes the time variable (in months) of the parameters k and c. Solving the second-order differential equation in Eq. (11), we obtain its steady-state, underdamped response solution

$$\hat{x}(\tau) = x_0 \cdot \exp\left[-\frac{c(t)}{2m} \cdot \tau\right] \cdot \cos\left[\sqrt{\frac{k(t)}{m} - \left(\frac{c(t)}{2m}\right)^2} \cdot \tau\right] \tag{12}$$

where the oscillation amplitude is $x_0 = 0.05$ m.

To account for the stochastic characteristics of the random degradation process of $k(t)$ and $c(t)$, ten simulation runs have been made for each quantity using discrete staircase functions. To simulate the "black-box" nature of the true degradation process, for each run, six distinct Markov kernels are derived to describe the evolution of $k(t)$ and $c(t)$ under uncertainty. The most probable Markov kernel would be identified based on the log-evidence. Given the most probable Markov kernel, the Bayesian identification of $k(t)$ and $c(t)$ is performed for $t = 1, \ldots, 6$ months and used for prediction estimates at $t = 7, 8, 9$ months. These results obtained using both the SMC and SEMC samplers are then compared on the basis of the estimation results, prediction results, computational time, and effectiveness in controlling the acceptance rates.

The results show that the SEMC sampler is able to yield estimates of $k(t)$ and $c(t)$ for all ten runs which are well-verified against those obtained by the SMC sampler. The computational time is significantly lower for the SEMC sampler compared to the SMC sampler. Furthermore, the SEMC sampler is able to effectively control the acceptance rates of the samples within optimal bounds compared to the SMC sampler. These observations illustrate and highlight the strengths of the SEMC sampler as hypothesized. Full details on the numerical setup and the results can be found in Ref. [68] and the corresponding MATLAB codes of the numerical example are freely available via link.[2]

6 Experimental Investigation: Single-Storey Shear Frame Structure Subjected to a Coulomb Friction

This experimental example is used to investigate the capability of the proposed approach to identify the most probable Markov kernel able to model the evolution of Coulomb friction force on the response dynamics of a physical structure under uncertainty. This investigation is conducted using a single-storey shear frame with a Coulomb friction contact, subjected to a harmonic base-excitation. The physical setup and schematic diagram of the structure are presented in Figs. 3 and 4, respectively. A detailed description of the physical setup can be found in Ref. [39].

The objective of this investigation is to evaluate the robustness of the proposed SEMC sampler in its ability to infer the values of Coulomb friction force at each observation, but also the time-invariant natural frequency of the structure and measurement error using sets of actual experimental data obtained sequentially.

6.1 Physics-Based Model of the Structure. The building can be modeled as spring-mass-damped system as shown in Fig. 5 where the mass m and the stiffness k represent the participating mass and stiffness of the first vibrating mode of the structure [39,40]. The Coulomb friction force F_μ is generated as a result of contact between the mass m and a fixed wall and is obtained as the product of a friction coefficient μ and the normal contact force F_N.

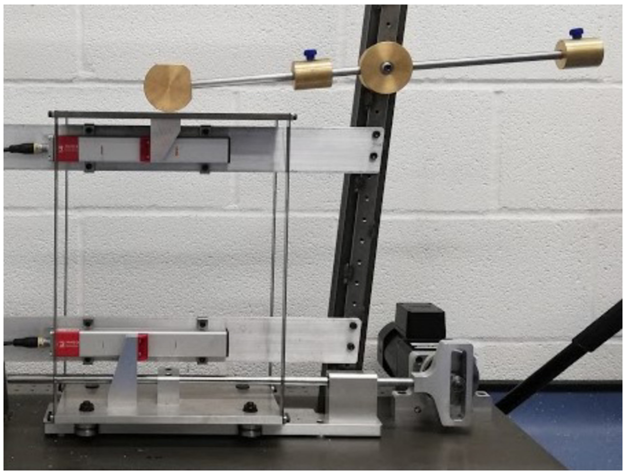

Fig. 3 SDoF single-storey shear frame structure subjected to Coulomb friction

To study the response dynamics of the structure as well as the parameters to be inferred in this problem, it is possible to write the governing equation of this SDoF model in a dimensionless form as [40]

$$r^2 \cdot \frac{d^2\tilde{x}}{d{\tau_b}^2} + \tilde{x} + \beta(t) \cdot \mathrm{sgn}\left(\frac{d\tilde{x}}{d\tau_b}\right) = \cos(\tau_b) \tag{13}$$

where $\tilde{x}$ is the dimensionless response of m, r is the frequency ratio, $\beta(t)$ is the time-varying friction ratio, and τ_b is the dimensionless time parameter. The dimensionless frequency ratio r is defined as [40]

$$r = \frac{\omega_b}{\omega_n} \tag{14}$$

where $\omega_n = \sqrt{\frac{k}{m}}$ is the natural frequency of the structure which was measured experimentally [39] to be 19.572 rad/s and ω_b is the driving frequency of the harmonic base-excitation. Hence, the dimensionless time τ_b in Eq. (13) is defined as: $\tau_b = \omega_b \cdot \tau$, where τ is the physical time parameter. The dimensionless force ratio $\beta(t)$ is defined as [40]

$$\beta(t) = \frac{F_\mu(t)}{kY_b} \tag{15}$$

where Y_b is the driving displacement amplitude by the rotor. Thus, kY_b is the driving force amplitude whose value was measured

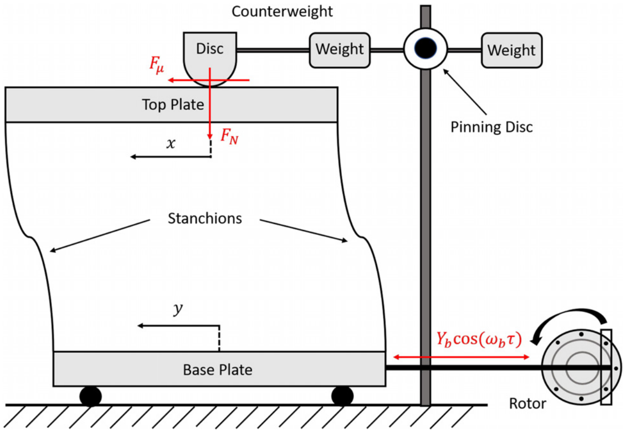

Fig. 4 Schematic diagram of the SDoF single-storey shear frame structure subjected to Coulomb friction. Image adapted from Ref. [43].

[2] https://github.com/Adolphus8/Sequential_Ensemble_Monte_Carlo.git

experimentally to be 2.50 N. Hence, the dimensionless response $\tilde{x}$ is defined as [40]

$$\tilde{x} = \frac{x}{Y_b} \tag{16}$$

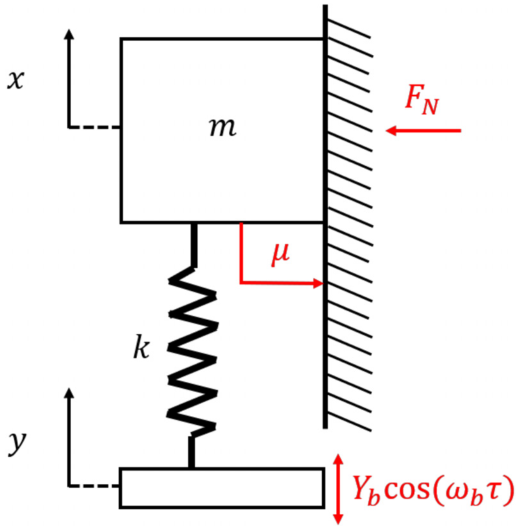

Fig. 5 Spring-mass representation of the SDoF single-storey shear frame structure subjected to Coulomb friction

where x is the response displacement of m. According to Den–Hartog's theory [69], under the assumptions of continuous and symmetric response, the steady-state solution of Eq. (13) can be obtained analytically as [40]

$$\tilde{x}_{num}(\tau_b) = \begin{cases} \tilde{x}(\tau_b) & \text{for } \tau_b \in [0, \pi) \\ -\tilde{x}(\tau_b - \pi) & \text{for } \tau_b \in [\pi, 2\pi) \end{cases} \tag{17}$$

where $\tilde{x}(\tau_b)$ can be evaluated as

$$\begin{aligned} \tilde{x}(\tau_b) = {} & \tilde{x}_0 \cdot \cos(\tau_b) + \beta(t) U \cdot \sin(\tau_b) + \beta(t) \\ & \cdot \left[1 - \cos\left(\frac{\tau_b}{r}\right) - Ur \cdot \sin\left(\frac{\tau_b}{r}\right) \right] \end{aligned} \tag{18}$$

In Eq. (18), the damping function U is defined as

$$U = \frac{\sin(\pi/r)}{r \cdot [1 + \cos(\pi/r)]} \tag{19}$$

while the dimensionless response amplitude $\tilde{x}_0$ can be evaluated as

$$\tilde{x}_0 = \sqrt{\left(\frac{1}{1 - r^2}\right)^2 - (\beta(t) \cdot U)^2} \tag{20}$$

Details on the derivation of the above terms can be found in Ref. [40].

Table 1 A spring-mass-damper: numerical results of r and ϕ (deg) obtained for the respective $F_\mu(t_s)$

		$F_\mu(t_1) = 1.435\ N$		$F_\mu(t_2) = 0.980\ N$		$F_\mu(t_3) = 0.662\ N$		$F_\mu(t_4) = 0.217\ N$	
Experiment	r_{nom}	r	ϕ	r	ϕ	r	ϕ	r	ϕ
1	0.65	0.649	46.464	0.637	27.217	0.624	16.622	0.652	6.496
2	0.80	0.791	41.492	0.789	26.269	0.807	16.900	0.796	4.292
3	0.95	0.952	41.864	0.944	29.936	0.941	14.934	0.936	5.839
4	1.10	1.098	132.661	1.099	147.498	1.123	162.318	1.110	174.223
5	1.25	1.278	137.022	1.253	150.104	1.255	160.757	1.262	174.832
6	1.40	1.407	129.795	1.406	152.246	1.409	156.074	1.392	171.955
7	1.55	1.557	136.944	1.549	152.011	1.540	161.960	1.548	173.666
8	1.70	1.706	131.314	1.694	152.008	1.711	157.884	1.715	171.698
9	1.85	1.848	134.294	1.849	153.251	1.833	161.017	1.860	169.833

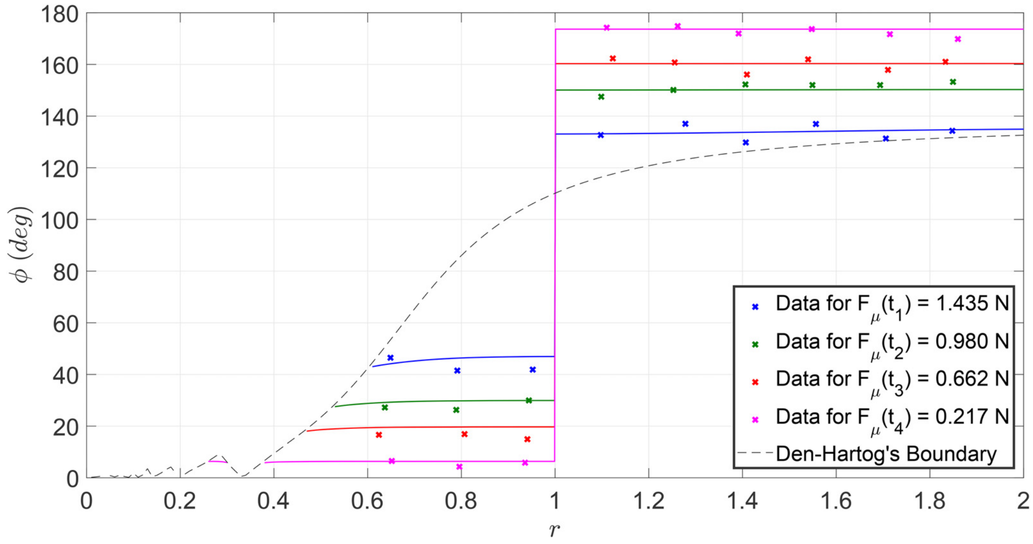

Fig. 6 Shear frame structure: Plots of r and ϕ for the corresponding values of $F_\mu(t_s)$ for $s = \{1, \ldots, 4\}$

Table 2 Shear frame structure: numerical values of ω_b (rad/s) used for the respective $F_\mu(t_s)$(N)

Experiment	1.435	0.980	0.662	0.217
$F_\mu(t_s)$	ω_b	ω_b	ω_b	ω_b
1	12.696	12.462	12.213	12.751
2	15.487	15.444	15.791	15.576
3	18.639	18.478	18.415	18.318
4	21.480	21.500	21.983	21.729
5	25.015	24.514	24.569	24.694
6	27.540	27.522	27.585	27.239
7	30.468	30.321	30.141	30.296
8	33.390	33.149	33.478	33.558
9	36.165	36.195	35.880	36.398

6.2 Data Collection. Four different values of Coulomb friction force $F_\mu(t)$ are considered to simulate its time-varying aspect for $t = \{1, 2, 3, 4\}$ months. This variation in $F_\mu(t)$ can be simulated by varying the configuration of the weights in the counterweight system seen in Fig. 4. The values of each $F_\mu(t)$ are $\{1.435, 0.980, 0.662, 0.217\}$ N, respectively. For each $F_\mu(t)$, 9 sets of phase angle data ϕ are collected across nine chosen values of frequency ratio r. The phase angle ϕ is chosen as the response data due to its high-degree of sensitivity to the variation in $F_\mu(t)$ as shown in Ref. [39]. The reference values for the nine chosen values of r, which can be obtained by adjusting ω_b (i.e., see Eq. (14)), are: $r_{\text{nom}} = \{0.65, 0.80, 0.95, 1.10, 1.25, 1.40, 1.55, 1.70, 1.85\}$. The experimental procedure to obtain the phase angles ϕ from given values of $F_\mu(t)$ and r can be found in Ref. [39].

The experimental measurements of ϕ, r, and $F_\mu(t)$ are shown in Table 1 and in Fig. 6. In Fig. 6, the Den–Hartog's boundary denotes the boundary between the continuous motion and the stick-slip regime for the dynamic response of the top plate under Coulomb friction, while the continuous colored lines represent the true model output of ϕ, given $F_\mu(t_s)$, for the different values of r within the continuous motion regime. In addition, the values of the driving frequency ω_b measured from each test are presented in Table 2.

Based on the experimentally obtained values of $F_\mu(t)$, two choices of Markov kernels are identified to model the time-based degradation of $F_\mu(t)$

$$T_1\text{: } F_\mu(t_{s+1}) = F_\mu(t_s) - 0.375 + \nu_1 \tag{21}$$

$$T_2\text{: } F_\mu(t_{s+1}) = \exp\left[-0.470\right] \cdot F_\mu(t_s) + \nu_2 \tag{22}$$

whereby $s = 1, \ldots, 4$ denotes the time sequence index, and ν_1 and ν_2 are the zero-mean normally distributed process noise terms with the respective standard deviations: $\{\sigma_1, \sigma_2\} = \{0.040, 0.090\}$ N. The parameters of the Markov kernels are obtained using a curve-fitting procedure via the least-squares method on the experimentally obtained values of $F_\mu(t)$. The corresponding nominal models Γ_1 and Γ_2 (i.e., Kernel models without the noise term, are shown in Fig. 7). It needs to be added that while in this example the parameters of the Markov kernel are assumed to be known, this is not always the case in general. In such cases, the parameters of the Markov kernel can also be included in the set of inferred parameters through Bayesian inference [41].

6.3 Bayesian Inference Setup. The sequential Bayesian inference procedure is done for $\boldsymbol{\theta}(t_s) = (F_\mu(t_s), \omega_n, \sigma_\phi, \sigma_r)$, where σ_ϕ and σ_r are the standard deviations of the respective noise associated with the experimentally obtained values of ϕ and r. The parameters ω_n, σ_ϕ, and σ_r are assumed to be time-invariant and it needs to be noted that σ_ϕ and σ_r are internal parameters of the likelihood function and are not used in the models to predict $F_\mu(t_s)$ and ω_n. This gives rise to a four-dimensional Bayesian inference problem for the estimation of the aforementioned parameters at each time sequence t_s.

The initial priors at $s = 1$ for each of the inferred parameters are set to be noninformative uniform priors whose bounds are listed in Table 3. For $s > 1$ the prior is the predictive distribution that is derived using Eq. (A9) by propagating the samples obtained at previous time-step s–1 through the Markov kernel. The likelihood function for each time sequence t_s is set to follow a Normal distribution. Assuming independence between individual observations of ϕ and r as well as between datasets obtained at different time sequence t_s, the likelihood function is defined as

$$P\left(\boldsymbol{D}^s | \boldsymbol{\theta}(t_s), \hat{\phi}, \hat{r}\right) = \prod_{q=1}^{9} \frac{1}{2\pi \cdot \sigma_r \cdot \sigma_\phi} \exp\left[-\frac{\left(r^q_{nom} - \hat{r}\left(\omega_b^{q,s}, \omega_n\right)\right)^2}{2 \cdot {\sigma_r}^2} - \frac{\left(\phi^{q,s} - \hat{\phi}\left(r^{q,s}, F_\mu(t_s)\right)\right)^2}{2 \cdot {\sigma_\phi}^2}\right] \tag{23}$$

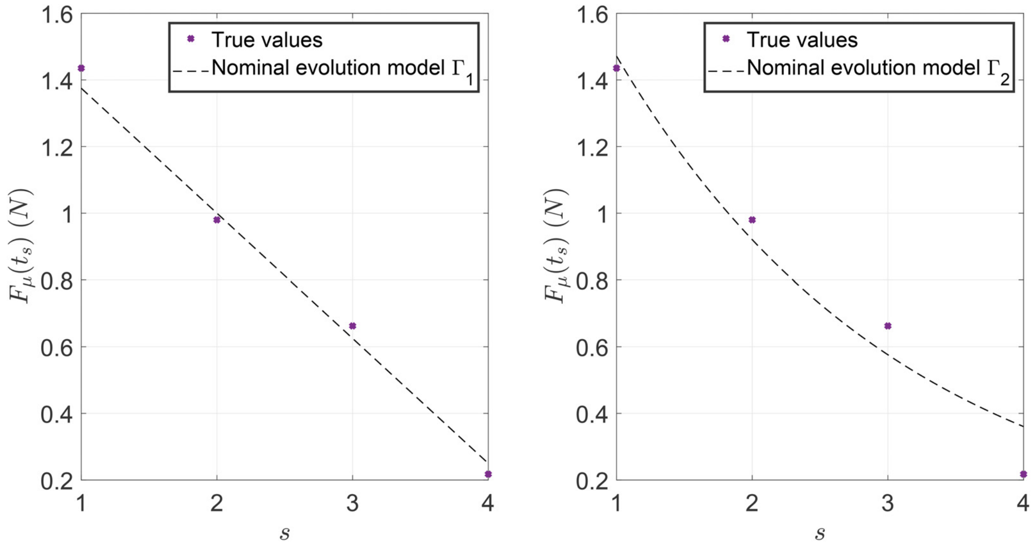

Fig. 7 Shear frame structure: Nominal evolution models Γ_1 and Γ_2

Algorithm 3 Pseudo-algorithm of model $\hat{\phi}$

```
procedure (compute $\phi$ from $F_\mu(t_s)$ and $r$)
    Compute $\beta(t_s)$ with $F_\mu(t_s)$ using Eq. (15)
    Compute $\beta_{\text{lim}}$ with $r$ using Eq. (24)
    if $\beta > \beta_{\text{lim}}$ then ▷ steady, continuous motion condition not satisfied
        Set $\phi = \text{NaN}$
    else
        Compute $\tilde{x}_{\text{num}}(\tau_b)$ using Eq. (17)
        Compute $\tilde{y}_{\text{num}}(\tau_b)$ using Eq. (25)
        Execute FFT on $\tilde{x}_{\text{num}}(\tau_b)$ to generate $\tilde{x}_{\text{FFT}}(\tilde{f})$
        Execute FFT on $\tilde{y}_{\text{num}}(\tau_b)$ to generate $\tilde{y}_{\text{FFT}}(\tilde{f})$
        Set $\tilde{f} = 1$
        Compute $\phi$ using Eq. (27)
    end if
end procedure
```

where r^q_{nom}, ω^q_b, and ϕ^q denote respectively the qth value/observation of r_{nom}, ω_b, and ϕ obtained at time sequence t_s for $q = 1, \ldots, 9$, $\hat{r}$ denotes the model used to compute r, $\hat{\phi}$ is the model used to compute ϕ, and $\boldsymbol{D}^s = (\phi, r)^s$ denotes the dataset obtained at t_s.

The model $\hat{r}$ evaluates the analytical solution for r from a given value of ω_b and ω_n according to Eq. (14). The computation procedure by the model $\hat{\phi}$ to evaluate the analytical solution of ϕ from a given value of $F_\mu(t_s)$ and r consists of three main steps [69].

In the first step, the algorithm computes $\beta(t_s)$ with the input value of $F_\mu(t_s)$ using Eq. (15). In order to verify the assumption of continuous response, the value of β corresponding to the boundary between continuous and stick-slip regimes, shown in Fig. 6, is also computed as [69]

$$\beta_{\text{lim}} = \sqrt{\frac{1}{\left(U^2 + \frac{1}{r^4}\right) \cdot \left(1 - r^2\right)^2}} \tag{24}$$

If $\beta(t_s) > \beta_{\text{lim}}$, the condition for a continuous motion is not satisfied. Therefore, the algorithm proceeds to assign a NaN (i.e., not a number) value for ϕ and the procedure terminates here. Otherwise, the algorithm proceeds to the next step. In the second step, the analytical steady-state response solution $\tilde{x}_{\text{num}}(\tau_b)$ is computed for $\tau_b \in [0, 2\pi)$ from Eq. (17) and the numerical excitation function $\tilde{y}_{\text{num}}(\tau_b)$, expressed as [69]:

$$\tilde{y}(\tau_b) = \cos(\tau_b + \varphi) \tag{25}$$

where

$$\varphi = \text{atan2}[-\beta(t_s) \cdot U \cdot (1 - r^2), \tilde{x}_0 \cdot (1 - r^2)] \tag{26}$$

is also computed for $\tau_b \in [0, 2\pi)$. In the last step, the algorithm proceeds to compute the phase angle ϕ between the excitation and the response functions. This is done by obtaining their respective dimensionless frequency spectra $\tilde{x}_{\text{FFT}}(\tilde{f})$ and $\tilde{y}_{\text{FFT}}(\tilde{f})$ using the FFT algorithm [70,71]. The dimensionless frequency is here defined as $\tilde{f} = \frac{2\pi \cdot f}{\omega_b}$, where f is the frequency variable in the FFT-space. From there, the phase angle is computed at $\tilde{f} = 1$ (i.e., resonance) following [39]:

$$\phi = \arg\{\tilde{x}_{\text{FFT}}(\tilde{f} = 1)\} - \arg\{\tilde{y}_{\text{FFT}}(\tilde{f} = 1)\} \tag{27}$$

The above procedure is provided as a pseudo-algorithm as shown in Algorithm 3. In the event $\hat{\phi} = \text{NaN}$, the likelihood function $P(\boldsymbol{D}^s | \boldsymbol{\theta}(t_s), \hat{\phi}, \hat{r})$ returns a 0.

6.4 Results.

To account for the statistical variability of the sampling algorithms, the inference is repeated ten times, and the means and standard deviation are calculated. For each run, 1000 samples are generated from the posteriors $P(\boldsymbol{\theta}^s | \boldsymbol{D}^{1:s}, M)$ and the log-evidence $\log[P(\boldsymbol{D}^{1:s} | M)]$ is computed at each time-step sequence s for each Markov kernel for both samplers. The numerical results are summarized in Tables 4 and 5 while a graphical plot illustration is provided in Fig. 8. As seen in Tables 4 and 5, $\log[P(\boldsymbol{D}^{1:s} | M)]$ is consistently higher for the case of T_1 which indicates that T_1 is the most probable Markov kernel to represent the variation of $F_\mu(t_s)$ across t_s.

The results of acceptance rates are illustrated in Fig. 9 showing a superior convergence of the acceptance rates of the SEMC sampler for both Markov kernels T_1 and T_2 right from time-step sequence $s = 1$ while the SMC sampler reached convergence from $s = 3$ onward. This demonstrates the effectiveness of the SEMC tuning algorithm in ensuring the convergence of the acceptance rates. Furthermore, the SEMC variability of the acceptance rate (shown as one-sigma error bars) is significantly smaller than the corresponding SMC variability.

The resulting statistics of the estimates for $F_\mu(t_s)$ across each time-step sequence s are obtained for each choice of Markov kernel and summarized in Figs. 10 and 11 and in Tables 6 and 7. Figure 10 shows the identified value of the parameter $F_\mu(t_s)$ with the corresponding error bars in correspond to the one-sigma bounds. In both cases follow the trend defined by the evolution model defined within the respective Markov kernels, for both the SEMC and SMC samplers.

The SEMC estimates for $F_\mu(t_s)$ given T_1 are generally closer to the true values compared to the estimates using T_2 while the standard deviations of the estimates at each time-step sequence s are generally the same between the different Markov kernels. Similar behavior is also observed for the SMC estimates. This is due to the fact that T_1 describes better the change of $F_\mu(t_s)$ across the time-step sequences s compared to T_2. Hence, this set of results illustrates the direct influence of the choice of Markov kernel on not just the trajectory of the time-varying estimates of $F_\mu(t_s)$, but also the accuracy of the estimates of $F_\mu(t_s)$ across the simulation runs at any given s.

The predictive capabilities of the identified models are shown in Fig. 11. This is obtained by propagating the 10,000 posterior samples from the ten repeated runs each with 1000 samples through the Markov kernels (see Eq. (A2)). In the figure, the red histogram plots represent the distribution profile of the predicted values of $F_\mu(t_s)$ from the posterior distribution at $s = 1$; the green histogram plots represent the prediction from the posterior distribution at $s = 2$, the blue histogram plots represent the prediction from the posterior distribution at $s = 3$, and the yellow histogram plots represent the posterior distribution obtained at $s = 4$. Although the samples from both models generally include the true value of $F_\mu(t_s)$ at any given time-step, the T_1 model shows a better predictive capability compared to T_2. Moreover, T_2 produces a significant number of samples with a value of Coulomb friction equal to zero, in particular for the model calibrated using only the data from $s = 1$ and $s = 2$.

The resulting statistics of the estimates for the parameters, ω_n, σ_ϕ, and σ_r, across the time-step sequence s, obtained for each choice of Markov kernel, are shown in Figs. 12 and 13 while the corresponding numerical results are summarized in Tables 8–13. Like in Fig. 10, the error bars correspond to the one-sigma bounds. For the case of ω_n, the SEMC and SMC estimates given either choice of the Markov kernel both converge to the experimentally measured values (see Sec. 6.1). This indicates the effectiveness of the samplers in inferring ω_n.

Table 3 Shear frame structure: bounds of the noninformative uniform prior for the respective inferred parameters

Parameter	Bounds	Units
$F_\mu(t_s)$	$[0.01, 100]$	N
ω_n	$[0.01, 100]$	rad/s
σ_ϕ	$[0.001, 10]$	deg
σ_r	$[0.001, 1]$	—

The average over the time-step s of the sampled standard deviations for the measured parameters are used as reference values denoted as $\sigma_\phi^{\mathrm{ref}}$ and σ_r^{ref}, respectively,

$$\sigma_\phi^{\mathrm{ref}} = \frac{1}{4}\sum_{s=1}^{4}\sqrt{\frac{1}{8}\sum_{q=1}^{9}\left[\phi^{q,s} - \hat{\phi}(F_\mu(t_s), r^{q,s})\right]^2} \tag{28}$$

$$\sigma_r^{\mathrm{ref}} = \frac{1}{4}\sum_{s=1}^{4}\sqrt{\frac{1}{8}\sum_{q=1}^{9}(r^{q,s} - r_{\mathrm{nom}})^2} \tag{29}$$

The SEMC sampler provides an estimate of σ_ϕ close to the reference values given either choice of the Markov kernel. However, the one-sigma bounds of the estimates given T_2 is significantly larger than that for T_1 (see Figs. 12 and 13). A similar observation

Table 4 Shear frame structure: model identification using SEMC—mean of the log-evidence ($\log[P(D^{1:s}|M)]$), and its standard deviation (Stdev)

s	T_1 E (log-evidence)	Stdev.	T_2 E (log-evidence)	Stdev.
1	–23.614	5.210	–23.574	5.212
2	–37.486	9.020	–38.913	13.052
3	–51.988	13.738	–53.311	19.356
4	–65.276	18.829	–67.125	25.783

Table 5 Shear frame structure: model identification using SMC—mean of the log-evidence ($\log[P(D^{1:s}|M)]$), and its standard deviation (Stdev)

s	T_1 E (log-evidence)	Stdev.	T_2 E (log-evidence)	Stdev.
1	–23.493	5.212	–23.543	5.209
2	–44.599	8.399	–45.435	10.501
3	–61.712	15.436	–64.471	18.207
4	–75.096	30.674	–82.697	26.252

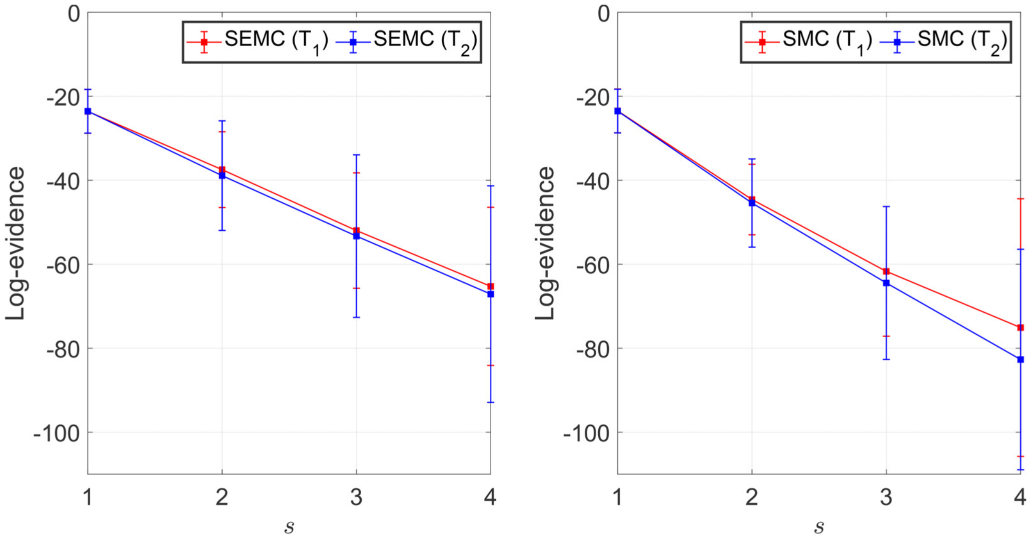

Fig. 8 Shear frame structure: model identification results using SEMC and SMC—mean of the log-evidence ($\log[P(D^{1:s}|M)]$), and its one-sigma bounds

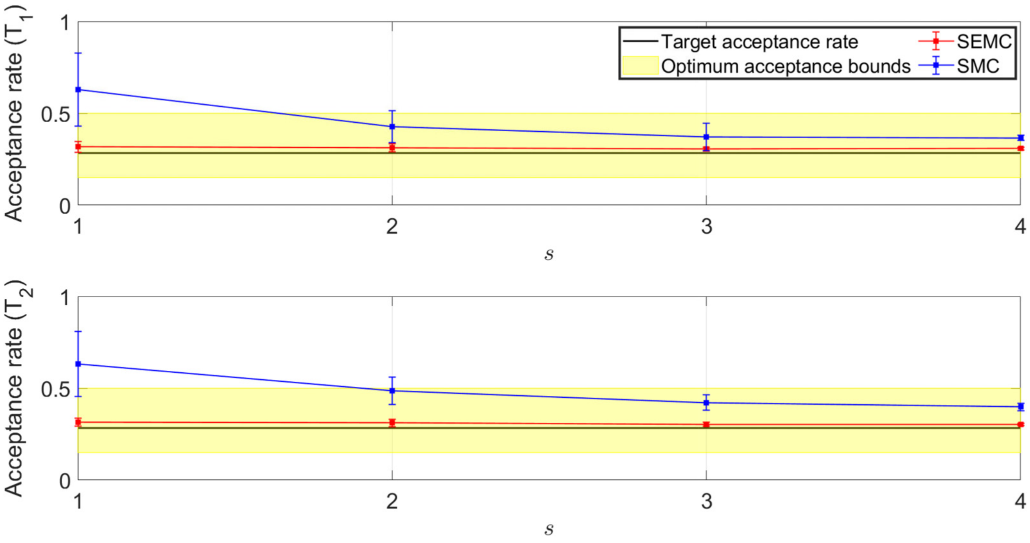

Fig. 9 Shear frame structure: acceptance rates result using SEMC and SMC—mean of the acceptance rates and its one-sigma bounds. Target acceptance rate: 0.283 (i.e., see Eq. (10)).

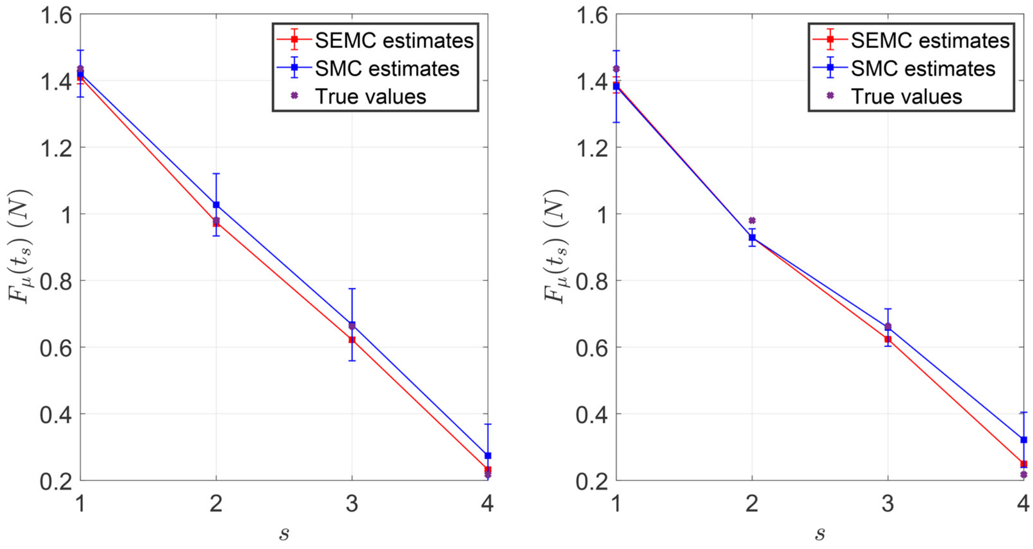

Fig. 10 Shear frame structure: parameter identification of $F_\mu(t_s)$ $[N]$ using SEMC and SMC—mean of $F_\mu(t_s)$, and its one-sigma bounds

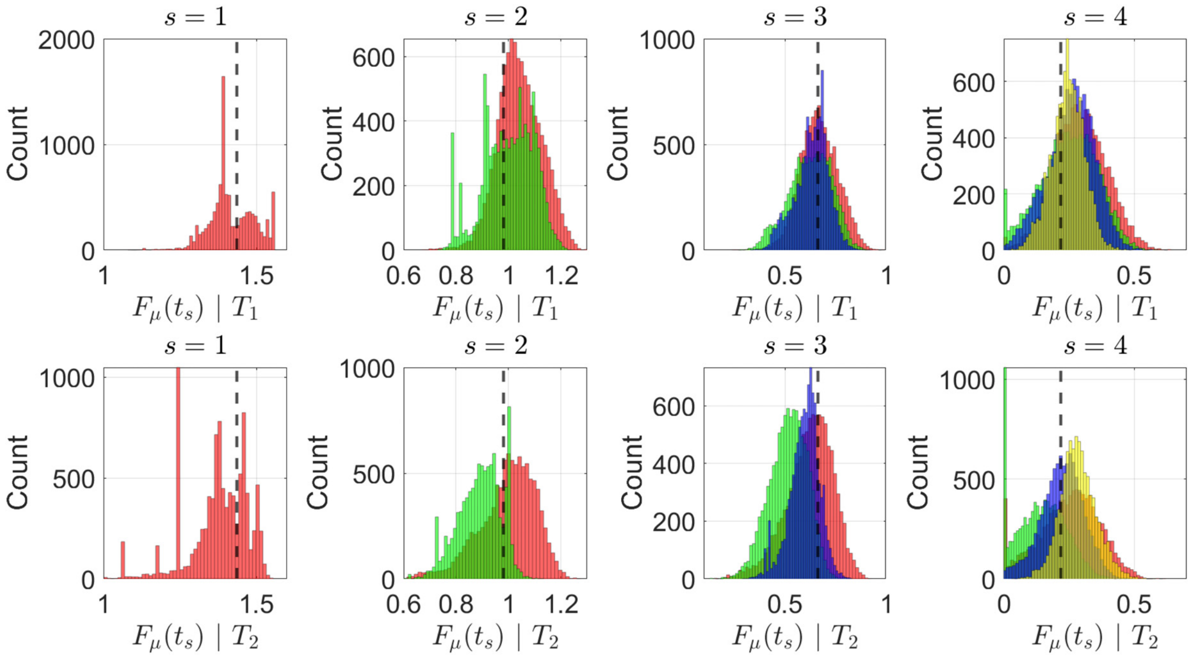

Fig. 11 Shear frame structure: posterior sample histogram profiles of the predicted values of $F_\mu(t_s)$ $[N]$ using SEMC. The black dotted vertical line denotes the true value of $F_\mu(t_s)$ at a given s.

Table 6 Shear frame structure: parameter identification of $F_\mu(t_s)$ (N) using SEMC—mean of $F_\mu(t_s)$, and its standard deviation (Stdev)

		T_1		T_2	
s	True value	$E[F_\mu(t_s)]$	Stdev.	$E[F_\mu(t_s)]$	Stdev.
1	1.435	1.409	0.019	1.387	0.024
2	0.980	0.975	0.012	0.929	0.007
3	0.662	0.622	0.007	0.624	0.003
4	0.217	0.233	0.004	0.250	0.003

Table 7 Shear frame structure: parameter identification of $F_\mu(t_s)$ $[N]$ using SMC—mean of $F_\mu(t_s)$, and its standard deviation (Stdev)

		T_1		T_2	
s	True value	$E[F_\mu(t_s)]$	Stdev.	$E[F_\mu(t_s)]$	Stdev.
1	1.435	1.421	0.070	1.382	0.108
2	0.980	1.027	0.094	0.929	0.026
3	0.662	0.667	0.108	0.659	0.056
4	0.217	0.274	0.095	0.322	0.083

Table 8 Shear frame structure: parameter identification of ω_n (rad/s) using SEMC—mean of ω_n, and its standard deviation (Stdev)

		T_1		T_2	
s	True value	$E[\omega_n]$	Stdev.	$E[\omega_n]$	Stdev.
1	19.572	20.943	1.198	21.056	1.590
2	19.572	20.737	1.220	21.252	1.583
3	19.572	20.791	1.202	21.145	1.504
4	19.572	20.771	1.198	21.155	1.531

Table 9 Shear frame structure: parameter identification of ω_n (rad/s) using SMC—mean of ω_n, and its standard deviation (Stdev)

		T_1		T_2	
s	True value	$E[\omega_n]$	Stdev.	$E[\omega_n]$	Stdev.
1	19.572	65.762	1.844	46.800	1.908
2	19.572	23.725	1.750	24.748	1.741
3	19.572	21.450	1.540	21.735	1.237
4	19.572	21.045	1.025	21.668	1.978

Table 10 Shear frame structure: parameter identification of σ_ϕ (deg) using SEMC—mean of σ_ϕ, and its standard deviation (Stdev)

		T_1		T_2	
s	Reference value	$E[\sigma_\phi]$	Stdev.	$E[\sigma_\phi]$	Stdev.
1	2.512	3.492	0.199	3.559	0.275
2	2.512	3.471	0.183	3.440	0.296
3	2.512	3.495	0.190	3.466	0.295
4	2.512	3.482	0.196	3.456	0.294

Table 11 Shear frame structure: parameter identification of σ_ϕ (deg) using SMC—mean of σ_ϕ, and its standard deviation (Stdev).

		T_1		T_2	
s	Reference value	$E[\sigma_\phi]$	Stdev.	$E[\sigma_\phi]$	Stdev.
1	2.512	6.600	0.686	6.584	0.683
2	2.512	6.921	0.686	5.646	0.658
3	2.512	6.934	0.638	6.030	0.665
4	2.512	6.479	0.660	5.949	0.682

Table 12 Shear frame structure: parameter identification of σ_r using SEMC—mean of σ_r, and its standard deviation (Stdev)

		T_1		T_2	
s	Reference value	$E[\sigma_r]$	Stdev.	$E[\sigma_r]$	Stdev.
1	0.010	0.383	0.053	0.383	0.108
2	0.010	0.378	0.052	0.357	0.111
3	0.010	0.366	0.054	0.357	0.119
4	0.010	0.366	0.052	0.356	0.118

Table 13 Shear frame structure: parameter identification of σ_r using SMC–mean of σ_r, and its standard deviation (Stdev)

		T_1		T_2	
s	Reference value	$E[\sigma_r]$	Stdev.	$E[\sigma_r]$	Stdev.
1	0.010	0.393	0.144	0.408	0.252
2	0.010	0.248	0.140	0.311	0.246
3	0.010	0.170	0.149	0.203	0.267
4	0.010	0.156	0.140	0.161	0.226

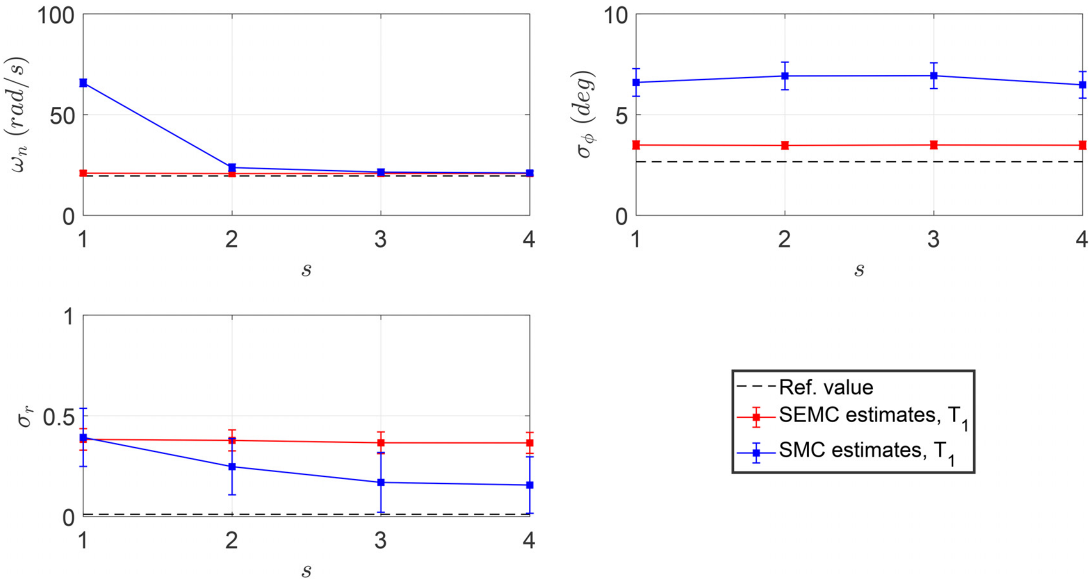

Fig. 12 Shear frame structure: parameter identification of ω_n (rad/s), σ_ϕ (deg), and σ_r using SEMC and SMC has given T_1—their corresponding means, and one-sigma bounds. The reference values for the respective parameters are: $\{\omega_n, \sigma_\phi, \sigma_r\} = \{19.572\,\text{rad/s}, 2.664\,\text{deg}, 0.011\}$.

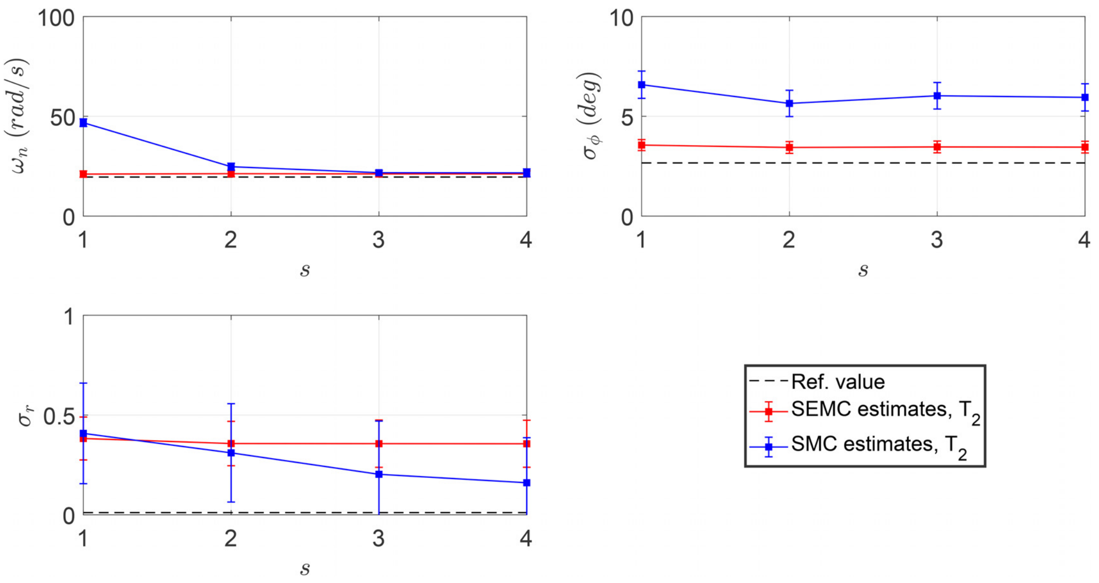

Fig. 13 Shear frame structure: parameter identification of ω_n (rad/s), σ_ϕ (deg), and σ_r using SEMC and SMC was given T_2—their corresponding means, and one-sigma bounds. The reference values for the respective parameters are: $\{\omega_n, \sigma_\phi, \sigma_r\} = \{19.572\text{rad/s}, 2.664\,\text{deg}, 0.011\}$.

is made for the SMC estimates of σ_ϕ although its estimates are approximately twice of that the SEMC sampler with nearly thrice the 1-sigma bounds. In estimating σ_r, neither the results obtained by the SEMC nor SMC samplers come close to the reference value of $\sigma_r^{\text{ref}} = 0.010$. Finally, for $F_\mu(t_s)$, ω_n, σ_ϕ, and σ_r, the SEMC sampler generally has a smaller standard deviation on its estimates compared to the SMC sampler.

7 Conclusions

A "tune-free" and robust sampler named sequential ensemble Monte Carlo has been proposed for the on-line Bayesian inference of time-varying parameters. The proposed sampler is characterized by the implementation of the affine-invariant ensemble sampler and automatic control of acceptance rates. Thanks to the introduction of the "virtual" iteration mechanism the sampler has not only automatized the tuning of the step-size parameter but also ensured that the acceptance rate converges toward the prescribed value defined within the admissible bounds. This, in turn, removes the need of tuning parameters thanks to the adaptive approach implemented in the algorithm. By doing so, the proposed sampler is robust, "tune-free" and generally applicable. All features required for the use of such tool in solving real engineering problems.

The proposed sampler has been verified and validated by identifying the time-variant parameters. The examples provided showed that the Sequential Ensemble Monte Carlo outperforms the traditional sampler yielding estimates with tighter bounds across independent simulation runs and with acceptance rates well-moderated within optimum bounds. Moreover, the experimental investigation has also shown that capability of the method to identify the most probable Markov kernel under model uncertainty.

The implemented algorithm and all the associated files and examples are freely accessible on the following repository.[2] This allows for the easy replication and independent check of the approach as well as the possibility of replicating the results presented.

Data Availability Statement

The data and information that support the findings of this article are freely available at following link.[2]

Appendix

A.1 Sequential Bayesian Filtering

The mathematical formulation of sequential Bayesian filtering to infer the time-varying parameter(s) is based on the well-known Bayes' theorem [72]

$$P(\boldsymbol{\theta}|\boldsymbol{D}, M) = \frac{P(\boldsymbol{D}|\boldsymbol{\theta}, M) \cdot P(\boldsymbol{\theta}|M)}{P(\boldsymbol{D}|M)} \tag{A1}$$

whereby $\boldsymbol{\theta}$ represents the vector of inferred model parameters (can be either time-invariant or time-varying), $\boldsymbol{D}$ represents the vector of measurements (or observations) used to update our knowledge of $\boldsymbol{\theta}$, and M denotes the model class which refers to a collection of mathematical models (i.e., as functions of $\boldsymbol{\theta}$) which are believed to best represent the observations $\boldsymbol{D}$ and the dynamics of the time-varying $\boldsymbol{\theta}$. The terms in Eq. (A1) are defined as [29]:

- $P(\boldsymbol{\theta}|M)$ is the prior distribution which describes our knowledge of $\boldsymbol{\theta}$ before observing $\boldsymbol{D}$,
- $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$ is the likelihood function which accounts for the degree of error between $\boldsymbol{D}$ and the output from M,
- $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ is the posterior distribution which describes our updated knowledge of $\boldsymbol{\theta}$ after observing $\boldsymbol{D}$,
- $P(\boldsymbol{D}|M)$ is the evidence that serves as the normalizing constant of the posterior.

A standard approach to sample from a target distribution would be Monte Carlo sampling which requires that the target distributions be normalized and have defined cumulative distribution functions (CDF) [29,73]. However, due to $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ not being known until its evaluation, Markov Chain Monte Carlo (MCMC) techniques are adopted to generate samples from it [29].

The interest here is in the use of the Bayesian inference framework toward inferring the time-varying $\boldsymbol{\theta}$, whose posterior is consequently varying over time. To reflect such aspects, we define $\boldsymbol{\theta}(t_s)$ as the parameter value at the sth time-step sequence whose instantaneous posterior can be expressed as $P(\boldsymbol{\theta}(t_s)|\boldsymbol{D}^{1:s}, M)$, while $\boldsymbol{D}^{1:s} = \{\boldsymbol{D}^1, \ldots, \boldsymbol{D}^s\}$ denotes the stream of dataset $\boldsymbol{D}$ obtained sequentially up to the sth time-step sequence in an on-line manner. An essential requirement in the inference of $\boldsymbol{\theta}(t_s)$ is the underlying Markov kernel $T(\boldsymbol{\theta}(t_{s+1})|\boldsymbol{\theta}(t_s))$ that describes the evolution from $\boldsymbol{\theta}(t_s)$ to $\boldsymbol{\theta}(t_{s+1})$ and can be expressed as

$$T(\boldsymbol{\theta}(t_{s+1})|\boldsymbol{\theta}(t_s)) : \boldsymbol{\theta}(t_{s+1}) = \Gamma(\boldsymbol{\theta}(t_s)) + \nu_{\boldsymbol{\theta}} \tag{A2}$$

where $\Gamma(\bullet)$ is the nominal evolution model and $\nu_{\boldsymbol{\theta}}$ is the process noise. In this paper, we shall assume that $\nu_{\boldsymbol{\theta}}$ follows a zero-mean Normal distribution with fixed standard deviation σ_ν (see, e.g., Refs. [74], and [75]).

The sequential Bayesian filtering procedure can be summarized as follows [10]: At time-step sequence $s = 1$, the posterior $P(\boldsymbol{\theta}(t_s)|\boldsymbol{D}^{1:s}, M)$ is defined (see Eq. (A1)). Following which, the predictive distribution $P(\boldsymbol{\theta}(t_{s+1})|\boldsymbol{D}^{1:s}, M)$ is computed [27]:

$$P(\boldsymbol{\theta}(t_{s+1})|\boldsymbol{D}^{1:s}, M) = \int T(\boldsymbol{\theta}(t_{s+1})|\boldsymbol{\theta}(t_s)) \cdot P(\boldsymbol{\theta}(t_s)|\boldsymbol{D}^{1:s}, M) \cdot d\boldsymbol{\theta}(t_s) \tag{A3}$$

The predictive distribution $P(\boldsymbol{\theta}(t_{s+1})|\boldsymbol{D}^{1:s}, M)$ describes our prediction of $\boldsymbol{\theta}(t_{s+1})$ before observing the data $\boldsymbol{D}^{s+1}$ to be obtained in time-step sequence number $s+1$. In this regard, $P(\boldsymbol{\theta}(t_{s+1})|\boldsymbol{D}^{1:s}, M)$ is set as the new before be updated and the process is repeated for time-step sequence number $s+1$ until the terminal sequence s_{end}.

To sample sequentially from the time-varying $P(\boldsymbol{\theta}(t_s)|\boldsymbol{D}^{1:s}, M)$, the SMC sampler can be implemented to which details can be found in the literature [43,68].

A.2 Sequential Monte Carlo Sampler

The SMC sampler is based on the sequential importance-resampling (SIR) algorithm in SMC methods (or Particle filters) to generate samples sequentially from a time-evolving posterior [43]. The sampling procedure is as follows [31]: At iteration $s = 0$, sampling algorithm is initialized by generating N samples generated from the prior $P(\boldsymbol{\theta}^{s+1})$ via standard Monte Carlo sampling. Next, the nominal weights are computed using the current likelihood function [76]

$$w_i^s = P(\boldsymbol{D}^s|\boldsymbol{\theta}_i^s, M) \tag{A4}$$

and normalized [32]

$$\hat{w}_i^s = \frac{w_i^s}{\sum_{i=1}^{N} w_i^s} \tag{A5}$$

where i is the index over the sample.

In the updating step, N single-step Markov chains are initiated, each starting from sample $\boldsymbol{\theta}_i^s$ obtained using weighted resampling (with replacement) according to w_i^s. The Metropolis–Hastings (MH) approach is then adopted to generate one sample from each Markov chains [53]. The candidate samples are generated from a normal proposal distribution $q(\boldsymbol{\theta}_i^{s*}|\boldsymbol{\theta}_i^s)$ with mean $\bar{\boldsymbol{\theta}}^s$ and covariance matrix $\boldsymbol{\Sigma}^s$ defined as [31]

$$\bar{\boldsymbol{\theta}}^s = \sum_{i=1}^{N} \boldsymbol{\theta}_i^s \cdot \hat{w}_i^s \tag{A6}$$

and

$$\boldsymbol{\Sigma}^s = \gamma^2 \cdot \sum_{i=1}^{N} \hat{w}_i^s \cdot [\{\boldsymbol{\theta}_i^s - \bar{\boldsymbol{\theta}}^s\} \times \{\boldsymbol{\theta}_i^s - \bar{\boldsymbol{\theta}}^s\}^T] \tag{A7}$$

where γ is the scaling parameter which will be set here as 1 [31]. The candidate sample is accepted with probability α_i:

$$\alpha_i = \min\left[1, \frac{P\left(\boldsymbol{\theta}_i^{s*}|\boldsymbol{D}^{1:s}, M\right)}{P\left(\boldsymbol{\theta}_i^{s}|\boldsymbol{D}^{1:s}, M\right)}\right] \tag{A8}$$

Then, the updated samples $\boldsymbol{\theta}_i^s$ are then passed through the Markov kernel $T(\boldsymbol{\theta}^{s+1}|\boldsymbol{\theta}^s)$ to generate $\boldsymbol{\theta}_i^{s+1}$ (i.e., the predictive samples). Finally, the predictive PDF is estimated using a kernel density estimate (KDE) in the form of

$$P\left(\boldsymbol{\theta}^{s+1}|\boldsymbol{D}^s, M\right) \approx \frac{1}{N}\sum_{i=1}^{N} K\left(\frac{\boldsymbol{\theta} - \boldsymbol{\theta}_i^{s+1}}{\boldsymbol{h}}\right) \tag{A9}$$

where $\boldsymbol{\theta}$ is a random variable, $K(\bullet)$ is the Kernel smoothing function which is set as the standard Normal distribution, and $\boldsymbol{h}$ is the N_d-by-N_d diagonal bandwidth matrix where N_d is the number of an inferred parameter(s). Each diagonal element of the bandwidth matrix h_d, for $d = 1, \ldots, N_d$ is computed using Silverman's rule of thumb [77]

$$h_d = \tilde{\sigma}_d \cdot \left[\frac{4}{N \cdot (d+2)}\right]^{\frac{1}{(d+4)}} \tag{A10}$$

where $\hat{\sigma}_d$ is the standard deviation of the dth component of $\boldsymbol{\theta}^{s+1}$.

The algorithm proceeds to the next time-step $s = s+1$ if data are available by setting $\boldsymbol{\theta}_i^{s+1}$ as the new prior samples and $P(\boldsymbol{\theta}^{s+1}|\boldsymbol{D}^{1:s}, M)$ as the new prior PDF. When no further data is obtained beyond that point, the algorithm terminates at time-step $s = s_{\text{end}}$. A pseudo-algorithm of the sampling procedure by the SMC sampler is presented in Algorithm 4.

Algorithm 4 SMC sampler algorithm

```
procedure (generate $N$ samples sequentially from $P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$)
    Set $s = 0$                                                        ▷ initialize counter
    Draw initial $N$ sample set: $\boldsymbol{\theta}_i^{s+1} \sim P(\boldsymbol{\theta}|M)$      ▷ generate samples prior
    while $s < s_{\text{end}}$ do                                         ▷ main sampling loop
        Set $s = s + 1$
        Compute $\hat{w}_i^s$ using Eq. (A5)
        for $i = 1 : N$ do                              ▷ For each $i$th chain (MCMC step)
            Resample: $\boldsymbol{\theta}_i^s \sim \hat{w}_s^j$
            Draw candidate sample: $\boldsymbol{\theta}_i^{s*} \sim q(\boldsymbol{\theta}_i^{s*}|\boldsymbol{\theta}_i^s)$
                Accept/Reject $\boldsymbol{\theta}_i^{s*}$ with probability $\alpha_i$ using Eq. (A8)
        end for
        Compute $\boldsymbol{\theta}_i^{s+1}$ using $T(\boldsymbol{\theta}^{s+1}|\boldsymbol{\theta}^s)$          ▷ set as new prior samples
        Compute PDF of $P(\boldsymbol{\theta}_i^{s+1}|D^{1:s})$ using Eq. (A9) ▷ set as new prior
        Compute $P(\boldsymbol{D}^{1:s}|M)$ using Eq. (1)
    end while
end procedure
```

References

[1] Saad, D., 2009, *On-Line Learning in Neural Networks*, Cambridge University Press, Cambridge, UK.

[2] Lye, A., Cicirello, A., and Patelli, E., 2022, "An Efficient and Robust Sampler for Bayesian Inference: Transitional Ensemble Markov Chain Monte Carlo," Mech. Syst. Signal Process., **167**, p. 108471.

[3] Rocchetta, R., Broggi, M., Huchet, Q., and Patelli, E., 2018, "On-Line Bayesian Model Updating for Structural Health Monitoring," Mech. Syst. Signal Process., **103**, pp. 174–195.

[4] Hu, D., Wang, W., Zhang, X., and Chen, K., 2021, "On-Line Real-Time Mistuning Identification and Model Calibration Method for Rotating Blisks Based on Blade Tip Timing (BTT)," Mech. Syst. Signal Process., **147**, p. 107074.

[5] Zhang, N., Chen, T., Zheng, M., Luo, L., and Liu, P., 2020, "Real-Time Identification of Vehicle Body Motion-Modes Based on Motion-Mode Energy Method," Mech. Syst. Signal Process., **143**, p. 106843.

[6] Xie, Z., and Feng, J., 2012, "Real-Time Nonlinear Structural System Identification Via Iterated Unscented Kalman Filter," Mech. Syst. Signal Process., **28**, pp. 309–322.

[7] Li, Y., Ding, L., Zheng, Z., Yang, Q., Zhao, X., and Liu, G., 2018, "A Multi-Mode Real-Time Terrain Parameter Estimation Method for Wheeled Motion Control of Mobile Robots," Mech. Syst. Signal Process., **104**, pp. 758–775.

[8] Ghaderi, P., and Amini, F., 2020, "Development of a New Method for Online Parameter Identification in Seismically Excited Smart Building Structures Using Virtual Synchronization and Adaptive Control Design," Appl. Math. Modell., **87**, pp. 203–221.

[9] Zhou, Y., Popolo, A. D., and Chang, Z., 2020, "On the Absence of a Universal Surface Density, and a Maximum Newtonian Acceleration in Dark Matter Haloes: Consequences for MOND," Phys. Dark Univ., **28**, p. 100468.

[10] Sarrka, S., 2013, *Bayesian Filtering and Smoothing*, 4th ed., Cambridge University Press, Cambridge, UK.

[11] Ou, G., Dyke, S. J., and Prakash, A., 2017, "Real Time Hybrid Simulation With Online Model Updating: An Analysis of Accuracy," Mech. Syst. Signal Process., **84**, pp. 223–240.

[12] Weng, J. H., and Loh, C. H., 2011, "Recursive Subspace Identification for on-Line Tracking of Structural Modal Parameter," Mech. Syst. Signal Process., **25**(8), pp. 2923–2937.

[13] Chen, J., Yuan, S., and Wang, H., 2020, "On-Line Updating Gaussian Process Measurement Model for Crack Prognosis Using the Particle Filter," Mech. Syst. Signal Process., **140**, p. 106646.

[14] Fan, X., and Liu, Y., 2019, "Use of Monitored Daily Extreme Stress Data for Performance Prediction of Steel Bridges: Dynamic Linear Models and Gaussian Mixed Particle Filter," Mech. Syst. Signal Process., **121**, pp. 841–855.

[15] Morse, L., Khodaei, Z. S., and Aliabadi, M. H., 2018, "Reliability Based Impact Localization in Composite Panels Using Bayesian Updating and the Kalman Filter," Mech. Syst. Signal Process., **99**, pp. 107–128.

[16] Huang, J., Li, X., Zhang, F., and Lei, Y., 2021, "Identification of Joint Structural State and Earthquake Input Based on a Generalized Kalman Filter With Unknown Input," Mech. Syst. Signal Process., **151**, p. 107362.

[17] Astroza, R., Alessandri, A., and Conte, J. P., 2019, "A Dual Adaptive Filtering Approach for Nonlinear Finite Element Model Updating Accounting for Modeling Uncertainty," Mech. Syst. Signal Process., **115**, pp. 782–800.

[18] Lund, A., Dyke, S. J., Song, W., and Bilionis, I., 2020, "Identification of an Experimental Nonlinear Energy Sink Device Using the Unscented Kalman Filter," Mech. Syst. Signal Process., **136**, p. 106512.

[19] Terejanu, G., Singla, P., Singh, T., and Scott, P. D., 2011, "Adaptive Gaussian Sum Filter for Nonlinear Bayesian Estimation," IEEE Trans. Autom. Control, **56**(9), pp. 2151–2156.

[20] Wang, X., Pan, Q., Liang, Y., and Li, H., 2015, "Gaussian Sum Approximation Filter for Nonlinear Dynamic Time-Delay System," Nonlinear Dyn., **82**(1–2), pp. 501–517.

[21] Chen, J., Yuan, S., and Jin, X., 2019, "On-Line Prognosis of Fatigue Cracking Via a Regularized Particle Filter and Guided Wave Monitoring," Mech. Syst. Signal Process., **131**, pp. 1–17.

[22] Rozas, H., Jaramillo, F., Perez, A., Jimenez, D., Orchard, M. E., and Medjaher, K., 2020, "A Method for the Reduction of the Computational Cost Associated With the Implementation of Particle-Filter-Based Failure Prognostic Algorithms," Mech. Syst. Signal Process., **135**, p. 106421.

[23] Wan, E. A., and Merwe, R. V. D., 2002, *Kalman Filtering and Neural Network*, Wiley, New York.

[24] Hur, S., 2021, "Short-Term Wind Speed Prediction Using Extended Kalman Filter and Machine Learning," Energy Rep., **7**, pp. 1046–1054.

[25] Kim, H., Jin, C., and Kim, M., 2021, "Real-Time Estimation of Riser's Deformed Shape Using Inclinometers and Extended Kalman Filter," Mar. Struct., **7**, p. 102933.

[26] Terejanu, G., Singla, P., Singh, T., and Scott, P. D., 2008, "A Novel Gaussian Sum Filter Method for Accurate Solution to the Nonlinear Filtering Problem," Proceedings of the 11th International Conference on Information Fusion, Vol. 1, Cologne, Germany, pp. 1–8.

[27] Ristic, B., Arulampalam, S., and Gordon, N., 2004, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, Boston, MA.

[28] Au, S. K., and Beck, J. L., 2003, "Important Sampling in High Dimensions," Struct. Saf., **25**(2), pp. 139–163.

[29] Lye, A., Cicirello, A., and Patelli, E., 2021, "Sampling Methods for Solving Bayesian Model Updating Problems: A Tutorial," Mech. Syst. Signal Process., **159**, p. 107760.

[30] Tatsis, K. E., Dertimanis, V. K., and Chatzi, E. N., 2022, "Sequential Bayesian Inference for Uncertain Nonlinear Dynamic Systems: A Tutorial," J. Struct. Dyn., **1**, pp. 236–262.

[31] Chopin, N., 2002, "A Sequential Particle Filter Method for Static Models," Biometrika, **89**(3), pp. 539–552.

[32] Moral, P. D., Doucet, A., and Jasra, A., 2006, "Sequential Monte Carlo Samplers," J. R. Stat. Soc. Ser. B (Stat. Methodol.), **68**(3), pp. 411–436.

[33] Zhu, G., Li, X., Ma, J., Wang, Y., Liu, S., Huang, C., Zhang, K., and Hu, X., 2018, "A New Moving Strategy for the Sequential Monte Carlo Approach in Optimizing the Hydrological Model Parameters," Adv. Water Resour., **114**, pp. 164–179.

[34] Cappe, O., Godsill, S. J., and Moulines, E., 2007, "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo," Proc. IEEE, **95**(5), pp. 899–924.

[35] Olsson, J., Cappe, O., Douc, R., and Moulines, E., 2008, "Sequential Monte Carlo Smoothing With Application to Parameter Estimation in Nonlinear State Space Models," Bernoulli, **14**(1), pp. 155–179.

[36] Frank, O., Nieto, J., Guivant, J., and Scheding, S., 2003, "Multiple Target Tracking Using Sequential Monte Carlo Methods and Statistical Data Association," Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vol. 3, Las Vegas, NV, Oct. 27–31, pp. 2718–2723.

[37] Shields, M. D., Giovanis, D. G., and Sundar, V. S., 2021, "Subset Simulation for Problems With Strongly Non-Gaussian, Highly Anisotropic, and Degenerate Distributions," Comput. Struct., **245**, p. 106431.

[38] Betz, W., Papaioannou, I., and Straub, D., 2016, "Transitional Markov Chain Monte Carlo: Observations and Improvements," J. Eng. Mech., **142**(5), p. 04016016.

[39] Marino, L., and Cicirello, A., 2020, "Experimental Investigation of a Single-Degree-of-Freedom System With Coulomb Friction," Nonlinear Dyn., **99**(3), pp. 1781–1799.

[40] Marino, L., Cicirello, A., and Hills, D. A., 2019, "Displacement Transmissibility of a Coulomb Friction Oscillator Subject to Joined Base-Wall Motion," Nonlinear Dyn., **98**(4), pp. 2595–2612.

[41] Beck, J. L., and Katafygiotis, L. S., 1998, "Updating Models and Their Uncertainties. I: Bayesian Statistical Framework," J. Eng. Mech., **124**(4), pp. 455–461.

[42] Katafygiotis, L. S., and Beck, J. L., 1998, "Updating Models and Their Uncertainties. II: Model Identifiability," J. Eng. Mech., **124**(4), pp. 463–467.

[43] Doucet, A., de Freitas, N., Gordon, N., and Smith, A., 2001, *Sequential Monte Carlo Methods in Practice*, Springer, New York.

[44] Drovandi, C. C., McGree, J. M., and Pettitt, A. N., 2014, "A Sequential Monte Carlo Algorithm to Incorporate Model Uncertainty in Bayesian Sequential Design," J. Comput. Graph. Stat., **23**(1), pp. 3–24.

[45] Nguyen, T. L. T., Septier, F., Peters, G. W., and Delignon, Y., 2013, "Bayesian Model Selection and Parameter Estimation in Penalized Regression Model Using SMC Samplers," Proceedings of 21st European Signal Processing Conference (EUSIPCO 2013), Vol. 1, Marrakech, Morocco, Sept. 9–13, pp. 1–5.

[46] Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P., 2009, "Approximate Bayesian Computation Scheme for Parameter Inference and Model Selection in Dynamical Systems," J. R. Soc. Interface, **6**(31), pp. 187–202.

[47] Urteaga, I., Bugallo, M. F., and Djuric, P. M., 2016, "Sequential Monte Carlo Methods Under Model Uncertainty," Proceedings of IEEE Statistical Signal Processing Workshop (SSP), Palma de Mallorca, Spain, June 26–29, pp. 1–5.

[48] Zhou, Y., Johansen, A. M., and Aston, J. A. D., 2016, "Toward Automatic Model Comparison: An Adaptive Sequential Monte Carlo Approach," J. Comput. Graph. Stat., **25**(3), pp. 701–726.

[49] Moffa, G., and Kuipers, J., 2014, "Sequential Monte Carlo EM for Multivariate Probit Models," Comput. Stat. Data Anal., **72**, pp. 252–272.

[50] Moral, P. D., Doucet, A., and Jasra, A., 2007, "Sequential Monte Carlo for Bayesian Computations," Proceedings of the Eight Valencia International Meeting, Benidorm, Spain, June 2–6, pp. 1–34.

[51] Ching, J. Y., and Chen, Y. C., 2007, "Transitional Markov Chain Monte Carlo Method for Bayesian Model Updating, Model Class Selection, and Model Averaging," J. Eng. Mech., **133**(7), pp. 816–832.

[52] Moral, P. D., Doucet, A., and Jasra, A., 2012, "On Adaptive Resampling Strategies for Sequential Monte Carlo Methods," Bernoulli, **18**(1), pp. 252–278.

[53] Hastings, W. K., 1970, "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," Biometrika, **57**(1), pp. 97–109.

[54] Martino, L., and Elvira, V., 2017, "Metropolis Sampling," *Wiley StatsRef: Statistics Reference Online*, Wiley, Hoboken, NJ, pp. 1–18.

[55] Green, P., and Maskell, S., 2017, "Estimating the Parameters of Dynamical Systems From Big Data Using Sequential Monte Carlo Samplers," Mech. Syst. Signal Process., **93**, pp. 379–396.

[56] Storn, R., and Price, K., 1997, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces," J. Global Optim., **11**(4), pp. 341–359.

[57] Bognanni, M., and Zito, J., 2020, "Sequential Bayesian Inference for Vector Autoregressions With Stochastic Volatility," J. Econ. Dyn. Control, **113**, p. 103851.

[58] Kantas, N., Beskos, A., and Jasra, A., 2014, "Sequential Monte Carlo Methods for High-Dimensional Inverse Problems: A Case Study for the Navier-Stokes Equations," SIAM/ASA J. Uncert. Quantif., **2**(1), pp. 464–489.

[59] Chib, S., 2001, "Markov Chain Monte Carlo Methods: Computation and Inference," Handbook of Econometrics, Vol. 5, Elsevier, The Netherlands, pp. 3569–3649.

[60] Goodman, J., and Weare, J., 2010, "Ensemble Samplers With Affine Invariance," Commun. Appl. Math. Comput. Sci., **5**(1), pp. 65–80.

[61] Gallier, J. H., 2012, *Geometric Methods and Applications: For Computer Science and Engineering*, Springer Science and Business Media, Germany.

[62] Wang, Y., and Solomon, J., 2019, "Intrinsic and Extrinsic Operators for Shape Analysis," *Handbook of Numerical Analysis Processing, Analyzing and Learning of Images, Shapes, and Forms: Part 2*, North Holland, The Netherlands, pp. 41–115.

[63] Foreman-Mackey, D., Hogg, D. W., Lang, D., and Goodman, J., 2013, "Emcee: The Mcmc Hammer," Publ. Astron. Soc. Pac., **125**(925), pp. 306–312.

[64] Lampart, T., 2012, "Implementation and Performance Comparison of an Ensemble Sampler With Affine Invariance," MOSAIC Group, Institute of Theoretical Computer Science, Department of Computer Science, ETH Zurich, Report.

[65] Iglesias, M., Park, M., and Tretyakov, M. V., 2018, "Bayesian Inversion in Resin Transfer Molding," Inverse Probl., **34**(10), p. 105002.

[66] Roberts, G. O., Gelman, A., and Gilks, W. R., 1997, "Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms," Ann. Appl. Probab., **7**(1), pp. 110–120.

[67] Hou, F., Goodman, J., Hogg, D. W., Weare, J., and Schwab, C., 2012, "An Affine-Invariant Sampler for Exoplanet Fitting and Discovery in Radial Velocity Data," Astrophys. J., **745**(2), p. 198.

[68] Lye, A., Cicirello, A., and Patelli, E., 2022, "On-Line Bayesian Inference for Structural Health Monitoring Under Model Uncertainty Using Sequential Ensemble Monte Carlo," Proceedings of the 13th International Conference on Structural Safety and Reliability, Vol. 1, Shanghai, China, Sept. 13–17.

[69] Den-Hartog, J. P., 1930, “Forced Vibrations With Combined Viscous and Coulomb Damping,” London, Edinburgh, Dublin Philos. Mag. J. Sci., **9**(59), pp. 801–817.

[70] Heideman, M., Johnson, D., and Burrus, C., 1984, “Gauss and the History of the Fast Fourier Transform,” IEEE ASSP Mag., **1**(4), pp. 14–21.

[71] Loan, C. V., 1992, *Computational Frameworks for the Fast Fourier Transform*, Siam, Switzerland.

[72] Bayes, T. P., 1763, “LII. An Essay Towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S,” Philos. Trans. R. Soc. London, **53**, pp. 370–418.

[73] Robert, C. P., and Casella, G., 2013, *Monte Carlo Statistical Methods*, 2nd ed., Springer Science & Business Media, Germany.

[74] Eftekhar-Azam, S., Mariani, S., and Attari, N. K. A., 2017, “Online Damage Detection Via a Synergy of Proper Orthogonal Decomposition and Recursive Bayesian Filters,” Nonlinear Dyn., **89**(2), pp. 1489–1511.

[75] Schon, T. B., Lindsten, F., Dahlin, J., Wagberg, J., Naesseth, C. A., Svensson, A., and Dai, L., 2015, “Sequential Monte Carlo Methods for System Identification,” IFAC-PapersOnLine, **48**(28), pp. 775–786.

[76] Hammersley, J. M., and Handscomb, D. C., 1964, *Monte Carlo Methods*, Springer, Netherlands.

[77] Silverman, B. W., Cox, D. R., and Hinkley, D. V., 1986, *Density Estimation for Statistics and Data Analysis*, 1st ed., Taylor and Francis, UK.