SEQUENTIAL ENSEMBLE MONTE CARLO SAMPLER FOR ON-LINE BAYESIAN INFERENCE OF TIME-VARYING MODEL PARAMETERS IN ENGINEERING APPLICATIONS

Adolphus Lye[1], Luca Marino[2], Alice Cicirello[1,2,3], and Edoardo Patelli[4]

[1]Institute for Risk and Uncertainty, University of Liverpool

[2]Department of Engineering Science, University of Oxford

[3]Faculty of Civil Engineering and Geoscience, Delft University of Technology

[4]Centre for Intelligent Infrastructure, Department of Civil and Environmental Engineering, University of Strathclyde. Email: edoardo.patelli@strath.ac.uk

ABSTRACT

Several on-line identification approaches have been proposed to identify parameters and evolution models of engineering systems and structures when sequential data-sets are available via Bayesian inference. In this work, a robust and “tune-free" sampler is proposed to extend one of the Sequential Monte Carlo implementation 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 data-sets. 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 non-linear dynamical system using experimental data.

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 data-sets which are obtained sequentially, contrary to the batch learning approach which requires the availability of the entire data-set to produce estimates (Saad 2009). 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 (Lye et al. 2022a; Rocchetta et al. 2018); mistuning parameters of rotating blisks (Hu et al. 2021); modal parameters of a vehicle motion modes to investigate its dominant motion-mode (Zhang et al. 2020); structural parameters of a nonlinear structural system to update its dynamical response model (Xie and Feng 2012); terrain parameters of the Wheel-Terrain model for wheeled motion control of mobile robots (Li et al. 2018); and structural parameters of smart building structures for real-time damage detection (Ghaderi and Amini 2020).

The parameters identified through on-line learning are often time-invariant, see e.g. (Zhou et al. 2020; Hu et al. 2021; Zhang et al. 2020). However, in many engineering applications, these parameters vary with time (Sarrka 2013) to which examples include: structural modal parameters to study the dynamic response of structures (Ou et al. 2017; Weng and Loh 2011); fatigue cracking parameters (Chen et al. 2020); stress data for performance prediction of steel bridges (Fan and Liu 2019a); localized impact damage in composite panels based on sensor data (Morse et al. 2018); and earthquake ground motion parameters using generalized Kalman filter and structural absolute accelerations data (Huang et al. 2021).

Currently, in time-domain applications, system parameters and system states such as displace-

ments and velocities, are estimated via Sequential Bayesian inference using Kalman filters (Lund et al. 2020; Astroza et al. 2019), Gaussian Sum filters (Wang et al. 2015; Terejanu et al. 2011), and Particle filters (Chen et al. 2019; Rozas et al. 2020). 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" (Wan and Merwe 2002). Extended Kalman filter (Kim et al. 2021; Hur 2021) and Unscented Kalman filter (Xie and Feng 2012; Lund et al. 2020) extended the approach to non-linear state-space and non-Gaussian "noise". For instance, Gaussian Sum filters utilises weighted Gaussian Models to approximate the predictive and posterior PDFs (Terejanu et al. 2011). 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 set-up becomes highly non-linear (Terejanu et al. 2008). Particle filters are applicable in both linear and non-linear state-space set-up and do not assume the form of the "noise" (Ristic et al. 2004). However, they are computationally expensive and perform poorly even with moderately high number of parameters (i.e. above 18 dimensions) (Au and Beck 2003; Lye et al. 2021). Despite this, Particle filters are more general and robust in its implementation (Ristic et al. 2004), since the error of the estimates reducible by obtaining more samples. For more information pertaining to the above approaches, the reader is referred to the recent review paper by (Tatis et al. 2022).

The Sequential Monte Carlo (SMC) sampler, whose concept is based on the Particle Filter (Moral et al. 2006; Chopin 2002), is a popular technique when addressing Bayesian inference problems (Zhu et al. 2018). Currently, it has been implemented under off-line (i.e. batch) and on-line settings towards the parameter identification for numerous problems involving: 1) non-linear time-series model; 2) non-linear state-space models; and 3) high-dimensional target distributions (Cappe et al. 2007; Olsson et al. 2008). 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 exists the following short-comings in the SMC algorithm: 1) the choice of proposal distribution can significantly affect the sampler results (Moral et al. 2006); 2) it is computationally inefficient because of the number of parameters required to compute and tune (Frank et al. 2003); 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 (Shields et al. 2021) and for model updating via Transitional Ensemble Monte Carlo sampler (Lye et al. 2022a) thanks to its capability of sampling from highly-skewed and anisotropic distributions (see Section 3). In addition, an adaptive-tuning algorithm, inspired from the work by (Betz et al. 2016), 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 Section 3.1. Numerical investigation has proven the existence of inherent bounds in the acceptance rates values and shown in Section 4. To the best of the authors' knowledge, such investigation and analysis is 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 2 time-varying parameters to which details are provided in Section 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 Section 6. Such application example is specifically chosen due to its importance in assessing the dynamic performance of structures and avoiding friction-related failures (Marino and Cicirello 2020; Marino et al. 2019). 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 $\boldsymbol{D}$ (Beck and Katafygiotis 1998; Katafygiotis and Beck 1998). 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 (Doucet et al. 2001; Drovandi et al. 2014).

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 s^{th} time step sequence (Moral et al. 2006; Drovandi et al. 2014). 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 in addressing problems regarding the model class selection of M (see e.g. (Toni et al. 2008; Nguyen et al. 2013; Drovandi et al. 2014; Urteaga et al. 2016; Zhou et al. 2016)). $P(\boldsymbol{D}^{1:s}|M)$ can be estimated by the product of the mean of the nominal weights w_i^s at any given time step sequence $s \geq 1$ (Moral et al. 2006):

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

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. (31)) (Moral et al. 2007; Moffa and Kuipers 2014). 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 its true distribution. To illustrate the resampling concept,

a numerical example involving a a mixture of two bi-variate Gaussian distributions based on work in Ref. (Ching and Chen 2007) is presented. Here, we shall consider a 2-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 bi-variate 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 $N = 10000$ samples and the prior samples with their associated normalised weights $\hat{w}_i^s$ are presented in Figure 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 in dark blue. 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 (Moral et al. 2012).

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 $\boldsymbol{\theta}_i^{s*}$ (Moral et al. 2006). This creates 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 (Hastings 1970; Martino and Elvira 2017). However, the ideal choice of $q(\boldsymbol{\theta}_i^{s*}|\boldsymbol{\theta}_i^s)$ is one that is "optimally" scaled to the current posterior $P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$ (Moral et al. 2006; Martino and Elvira 2017; Green and Maskell 2018). 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 (Chopin 2002) 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 s^{th} time step sequence, which can be the case for both the independent MH (Chopin 2002) 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) (Zhu et al. 2018; Storn and Price 1997), 2) Gibbs sampler, i.e. SMC for Vector Auto-regressions with Stochastic Volatility (VAR-SV) (Bognanni and Zito 2020), and 3) target-invariant MCMC mutation kernel, i.e. SMC for high-dimensional inverse problems (Kantas et al. 2014).

The PEM-SMC sampler has demonstrated its strength in generating samplers more effectively from complex-shaped distributions, especially those with multiple peaks (Zhu et al. 2018). 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 towards the posterior distribution (Zhu et al. 2018). 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 is 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 (Bognanni and Zito 2020). In addition, it is highly parallelizable which allows for the rapid update of samples from one posterior to the next across iterations (Bognanni and Zito 2020). 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 (Chib 2001). Should the posterior be functionally complex, choosing such conditional distribution becomes non-trivial (Lye et al. 2021).

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 (Kantas et al. 2014). 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 (Kantas et al. 2014). 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 (Kantas et al. 2014).

For the work presented here, the AIES is implemented as the alternative MCMC kernel to the MH. The reasons behind such proposal are: 1) ability to sample from anisotropic and highly-skewed distributions; 2) can be parallelised; 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 a MCMC sampling technique endowed with the affine-invariance property which was recently developed by Goodman and Weare (2010) (Goodman and Weare 2010). 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 (Gallier 2012):

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

where $\boldsymbol{\Theta}$ denotes $\boldsymbol{\theta}$ in the affine-transformed space, $\hat{A}$ is the N_d-by-N_d non-singular 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 (Wang and Solomon 2019):

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

A sampler is affine-invariant if it possesses a 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 (Goodman and Weare 2010).

An ensemble $\vec{\boldsymbol{\theta}}_i$ is defined as a collection of N_c Markov chains such that: $\vec{\boldsymbol{\theta}}_i = \{\boldsymbol{\theta}_{1,i}, \boldsymbol{\theta}_{2,i}, \ldots, \boldsymbol{\theta}_{N_c-1,i}, \boldsymbol{\theta}_{N_c,i}\}$. In practice, N_c should be at least twice the dimensionality of $\boldsymbol{\theta}$ (i.e. $N_c \geq 2 \times N_d$) (Goodman and Weare 2010).

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 $\vec{\boldsymbol{\theta}}_i$ for $i = 1$. Next, the samples are updated one at a time in a sequential manner. To update the k^{th} 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 $\boldsymbol{\theta}^*_{k,i}$ (Goodman and Weare 2010; Foreman-Mackey et al. 2013):

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

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

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

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 (Goodman and Weare 2010):

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

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

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

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.

Algorithm 1 AIES sampler algorithm

procedure (Generate N samples from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$)
  Define N_c chains: $\vec{\boldsymbol{\theta}}_1 = \{\boldsymbol{\theta}_{1,1}, \boldsymbol{\theta}_{2,1}, ..., \boldsymbol{\theta}_{N_c-1,1}, \boldsymbol{\theta}_{N_c,1}\}$ ▷ 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 α_{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

The AIES sampler presents 2 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 (Goodman and Weare 2010; Lampart 2012) 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 (Iglesias et al. 2018). 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 Section 3.1.

3.1 Adaptive-tuning Algorithm

The proposed adaptive-tuning algorithm is based on the work by (Betz et al. 2016) 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 towards the user-defined target acceptance rates and fall within the optimal bounds of $[0.15, 0.50]$ suggested by (Roberts et al. 1997).

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 (Foreman-Mackey et al. 2013; Hou et al. 2012). From this initial value, the nominal step-size u_{nom} is computed after the MCMC step:

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

where α^s is the acceptance rate for the current time step sequence s and α_{tr} is the target acceptance rate defined as (Roberts et al. 1997):

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

If $u_{nom} > 1$, then $u^{s+1} = u_{nom}$. 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 $s = s_{end}$.

To provide additional robustness to the SEMC sampler and assuring the acceptance rates converge towards the target acceptance value at every iteration, “virtual" iterations are introduced

which involves the updating of the posterior samples with a series of repeated data $\boldsymbol{D}^s$. The termination criteria is 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 towards 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 Section 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 Figure 2. It needs to be highlighted that the acceptance rate can be controlled by controlling the step-size u^s (Goodman and Weare 2010). 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 re-populated. This ensures that a proportion of the samples will always be accepted, resulting in a non-zero acceptance rate.

On the other hand, the upper bound of the acceptance rate is in practice always less than 1 given that for a non-Uniform 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 Figure 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 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 Figure 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 (Ching and Chen 2007); and 2) the final distribution of samples would be representative of the analytical distribution of $P(\boldsymbol{\theta}_i^s|D^{1:s}, M)$.

Algorithm 2 Proposed SEMC sampler algorithm

procedure (Generate N samples from $P(\boldsymbol{\theta}^j|\boldsymbol{D}^{1:j}, M)$)
  Set $s = 0$ ▷ Initialise time step counter
  Draw initial N sample set: $\boldsymbol{\theta}_i^{s+1} \sim P(\boldsymbol{\theta}|M)$ ▷ Generate samples from prior
  Set $u^{s+1} = 2$ ▷ Set initial value of step-size
  while $s < s_{end}$ **do** ▷ Loop over time steps
    Set $s = s + 1$
    Set $j = 1$ & $\alpha_{old} = \alpha_{tr}$ ▷ Initialize parameters
    Compute $\hat{w}_i^s$ using Eq. (34)
    while do ▷ Initiate "virtual" loop
      Resample N samples: $\boldsymbol{\theta}_i^s \sim \hat{w}_i^s$
      Set $\boldsymbol{\theta}_i^s = \boldsymbol{\theta}_{i,1}$ in ensemble $\vec{\boldsymbol{\theta}}_1$ ▷ Initiate ensemble
      Update $\vec{\boldsymbol{\theta}}_1$ with 1 iteration of AIES (see Algorithm 1) ▷ MCMC step
      Compute α using Eq. (8)
      Compute u_{nom} using Eq. (9) ▷ Tuning the step-size
      Set $u^{s+1} = \max(u_{nom}, 1.01)$
      if $|\alpha - \alpha_{old}| < 0.1 \cdot \alpha_{tr}$ **then** ▷ Check termination criteria
        Break ▷ Exit "virtual" loop
      end if
      Set $\alpha_{old} = \alpha$
      Set $j = j + 1$
    end while ▷ End "virtual" loop
    Set updated ensemble $\vec{\boldsymbol{\theta}}_1$ as $\boldsymbol{\theta}_i^s \sim P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$
    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. (38) ▷ Set as new prior PDF
    Compute $P(\boldsymbol{D}^{1:s}|M)$ using Eq. (1)
  end while
end procedure

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 $m = 0.30\ kg$ 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:

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

where x denotes the instantaneous displacement of the mass from its rest position, τ denotes

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 $c(t)$, 10 simulations 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, 6 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 its 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 10 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 hypothesised. Full details on the numerical set-up and the results can be found in (Lye et al. 2022b) and the corresponding MATLAB codes of the numerical example are freely available via: https://github.com/Adolphus8/Sequential_Ensemble_Monte_Carlo.git

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 set-up and schematic diagram of the structure are presented in Figures 3 and 4, respectively. Detailed description to the physical set-up can be found in (Marino and Cicirello 2020).

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 modelled as Spring-Mass-Damped system as shown in Figure 5 where the mass m and the stiffness k represent the participating mass and stiffness of the first vibrating mode of the structure (Marino and Cicirello 2020; Marino et al. 2019). 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.

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 (Marino et al. 2019):

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

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 (Marino et al. 2019):

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

where $\omega_n = \sqrt{\frac{k}{m}}$ is the natural frequency of the structure which was measured experimentally (Marino and Cicirello 2020) 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 (Marino et al. 2019):

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

where Y_b is the driving displacement amplitude by the rotor. Thus, kY_b is the driving force amplitude whose value was measured experimentally to be 2.50 N. Hence, the dimensionless response $\tilde{x}$ is defined as (Marino et al. 2019):

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

where x is the response displacement of m. According to Den-Hartog's theory (Den-Hartog 1930), under the assumptions of continuous and symmetric response, the steady-state solution of Eq. (13) can be obtained analytically as (Marino et al. 2019):

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

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

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

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 (Marino et al. 2019).

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 Figure 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 9 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 (Marino and Cicirello 2020). The reference values for the 9 chosen values of r, which can be obtained by adjusting ω_b (i.e. see Eq. (14)), are: $r_{nom} = \{0.65, 0.80, 0.95, 1.10, 1.25, 1.40, 1.55, 1.70, 1.85\}$. The experimental procedure to obtain the phase angles ϕ from given values of $F_\mu(t)$ and r can be found in (Marino and Cicirello 2020).

The experimental measurements of ϕ, r, and $F_\mu(t)$ are shown in Table 1 and in Figure 6. In Figure 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 $F_\mu(t)$:

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

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

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 Figure 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 (Beck and Katafygiotis 1998).

6.3 Bayesian Inference set-up

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 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 4-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 non-informative Uniform priors whose bounds are listed in Table 3. For $s > 1$ the prior is the predictive distribution that is derived using Eq. (38) 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 data-sets obtained at different time sequence t_s, the likelihood function is defined as:

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

where r_{nom}^q, ω_b^q and ϕ^q denote respectively the q^{th} 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 data set 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 3 main steps (Den-Hartog 1930):

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 Figure 6, is also computed as (Den-Hartog 1930):

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

If $\beta(t_s) > \beta_{lim}$, 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 $\tilde{x}_{num}(\tau_b)$ is computed for $\tau_b \in [0, 2\pi)$ from Eq. (17) and the numerical excitation function $\tilde{y}_{num}(\tau_b)$, expressed as (Den-Hartog 1930):

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

where:

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

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 $\tilde{x}_{FFT}(\tilde{f})$ and $\tilde{y}_{FFT}(\tilde{f})$ using the FFT algorithm (Heideman et al. 1984; Loan 1992). 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 (Marino and Cicirello 2020):

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

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

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

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

6.4 Results

To account for the statistical variability of the sampling algorithms, the inference is repeated 10 times and the means and standard deviation calculated. For each run, 1000 samples are generated from the posteriors $P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$ and the log-evidence $\log\left[P(\boldsymbol{D}^{1:s}|M)\right]$ is computed at each time step sequence s for each Markov kernel for both samplers. The numerical results are summarised in Tables 4 and 5 while a graphical plot illustration is provided in Figure 8. As seen in Tables 4 and 5, $\log\left[P(\boldsymbol{D}^{1:s}|M)\right]$ 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 Figure 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 $s = 3$ onwards. 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 1-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 summarised in Figures 10-11 and in Tables 6-7. Figure 10 shows the identified value of the parameter $F_\mu(t_s)$ with the corresponding error bars in correspond to the 1-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 behaviour 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 show in Figure 11. This is obtained by propagating the 10000 posterior samples from the 10 repeated runs each with 1000 samples through the Markov kernels (see Eq. (31). 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 Figures 12 and 13 while the corresponding numerical results summarised in Tables 8 to 13. Like in Figure 10, the error bars correspond to the 1-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 Section 6.1) . This indicates the effectiveness of the samplers in inferring ω_n.

The average over the time step s of the sampled standard deviations for the measured parameters are used as reference values denoted as σ_ϕ^{ref} and σ_r^{ref} respectively:

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

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

The SEMC sampler provides an estimate of σ_ϕ close to the reference values given either choice of the Markov kernel. However, the 1-sigma bounds of the estimates given T_2 is significantly larger than that for T_1 (see Figures 12 and 13). A similar observation is made for the SMC estimates of σ_ϕ although its estimates are approximately twice of that by the SEMC sampler with nearly thrice the 1-sigma bounds. In estimating σ_r, neither the results obtained by the SEMC and SMC samplers come close to the reference value of $\sigma_r^{ref} = 0.010$. 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 an automatic control of acceptance rates. Thanks to the introduction of the “virtual" iteration mechanism the sampler has not only automatised the tuning of the step-size parameter, but also ensured that the acceptance rate converges towards 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: https://github.com/Adolphus8/Sequential_Ensemble_Monte_Carlo.git. This allows for the easy replication and independent check of the approach as well the possibility of replicating the results presented.

8 APPENDIX

8.1 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 (Bayes and Price 1763):

$$P(\boldsymbol{\theta}|\boldsymbol{D}, M) = \frac{P(\boldsymbol{D}|\boldsymbol{\theta}, M) \cdot P(\boldsymbol{\theta}|M)}{P(\boldsymbol{D}|M)} \tag{30}$$

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. (30) are defined as (Lye et al. 2021):

- $P(\boldsymbol{\theta}|M)$ 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 which 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 normalised and have a defined Cumulative Distribution Functions (CDF) (Lye et al. 2021; Robert and Casella 2013). 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 (Lye et al. 2021).

The interest here is in the use of the Bayesian inference framework towards 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 s^{th} 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 data-

set $\boldsymbol{D}$ obtained sequentially up to the s^{th} 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 (Fan and Liu 2019b):

$$T(\boldsymbol{\theta}(t_{s+1})|\boldsymbol{\theta}(t_s)) : \boldsymbol{\theta}(t_{s+1}) = \Gamma(\boldsymbol{\theta}(t_s)) + \nu_{\boldsymbol{\theta}} \tag{31}$$

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. (Fan and Liu 2019b; Schon et al. 2015; Eftekhar-Azam et al. 2017)).

The Sequential Bayesian filtering procedure can be summarised as follows (Sarrka 2013): At time step sequence $s = 1$, the posterior $P(\boldsymbol{\theta}(t_s)|\boldsymbol{D}^{1:s}, M)$ is defined (see Eq. (30)). Following which, the predictive distribution $P(\boldsymbol{\theta}(t_{s+1})|\boldsymbol{D}^{1:s}, M)$ is computed (Ristic et al. 2004):

$$P(\boldsymbol{\theta}(t_{s+1})|\boldsymbol{D}^{1:s}, M) = \int T(\boldsymbol{\theta}(t_{s+1})|\boldsymbol{\theta}(t_s)) \cdot P(\boldsymbol{\theta}(t_s)|\boldsymbol{D}^{1:s}, M) \cdot d\boldsymbol{\theta}(t_s) \tag{32}$$

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 prior to 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 (Doucet et al. 2001; Lye et al. 2022b).

8.2 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 (Doucet et al. 2001). The sampling procedure is as follows (Chopin 2002): 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 (Hammersley and Handscomb 1964):

$$w_i^s = P(\boldsymbol{D}^s|\boldsymbol{\theta}_i^s, M). \tag{33}$$

and normalised (Moral et al. 2006):

$$\hat{w}_i^s = \frac{w_i^s}{\sum_{i=1}^N w_i^s} \tag{34}$$

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 1 sample from each Markov chains (Hastings 1970). 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 (Chopin 2002):

$$\bar{\boldsymbol{\theta}}^s = \sum_{i=1}^N \boldsymbol{\theta}_i^s \cdot \hat{w}_i^s \tag{35}$$

and

$$\boldsymbol{\Sigma}^s = \gamma^2 \cdot \sum_{i=1}^N \hat{w}_i^s \cdot \left[\{\boldsymbol{\theta}_i^s - \bar{\boldsymbol{\theta}}^s\} \times \{\boldsymbol{\theta}_i^s - \bar{\boldsymbol{\theta}}^s\}^T\right] \tag{36}$$

where γ is the scaling parameter which will be set here as 1 (Chopin 2002). The candidate sample is accepted with probability α_i:

$$\alpha_i = \min\left[1, \frac{P(\boldsymbol{\theta}_i^{s*}|\boldsymbol{D}^{1:s}, M)}{P(\boldsymbol{\theta}_i^s|\boldsymbol{D}^{1:s}, M)}\right] \tag{37}$$

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:

$$P(\boldsymbol{\theta}^{s+1}|\boldsymbol{D}^s, M) \approx \frac{1}{N}\sum_{i=1}^N K\left(\frac{\boldsymbol{\theta} - \boldsymbol{\theta}_i^{s+1}}{\boldsymbol{h}}\right) \tag{38}$$

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 inferred parameter(s). Each diagonal element of the bandwidth matrix h_d, for $d = 1, \ldots, N_d$ is computed using the Silverman's Rule of Thumb (Silverman et al. 1986):

$$h_d = \tilde{\sigma}_d \cdot \left[\frac{4}{N \cdot (d+2)}\right]^{\frac{1}{(d+4)}} \tag{39}$$

where $\hat{\sigma}_d$ is the standard deviation of the d^{th} 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 $s = s_{end}$. 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 $s = 0$ ▷ Initialise counter
  Draw initial N sample set: $\boldsymbol{\theta}_i^{s+1} \sim P(\boldsymbol{\theta}|M)$ ▷ Generate samples prior
  while $s < s_{end}$ **do** ▷ Main sampling loop
    Set $s = s + 1$
    Compute $\hat{w}_i^s$ using Eq. (34)
    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 α_i using Eq. (37)
    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. (38) ▷ Set as new prior
    Compute $P(\boldsymbol{D}^{1:s}|M)$ using Eq. (1)
  end while
end procedure

REFERENCES

Astroza, R., Alessandri, A., and Conte, J. P. (2019). "A dual adaptive filtering approach for nonlinear finite element model updating accounting for modeling uncertainty." *Mechanical Systems and Signal Processing*, 115, 782–800.

Au, S. K. and Beck, J. L. (2003). "Important Sampling in High Dimensions." *Structural Safety*, 25, 139–163.

Bayes, T. and Price (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.." *Philosophical Transactions of the Royal Society of London*, 53, 370–418.

Beck, J. L. and Katafygiotis, L. S. (1998). "Updating Models and Their Uncertainties. I: Bayesian Statistical Framework." *Journal of Engineering Mechanics*, 124, 455–461.

Betz, W., Papaioannou, I., and Straub, D. (2016). "Transitional Markov Chain Monte Carlo: Observations and Improvements." *Journal of Engineering Mechanics*, 142, .

Bognanni, M. and Zito, J. (2020). "Sequential Bayesian inference for vector autoregressions with stochastic volatility." *Journal of Economic Dynamics and Control*, 113, 103851.

Cappe, O., Godsill, S. J., and Moulines, E. (2007). "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo." *Proceedings of the IEEE*, 95, 899–924.

Chen, J., Yuan, S., and Jin, X. (2019). "On-line prognosis of fatigue cracking via a regularized particle filter and guided wave monitoring." *Mechanical Systems and Signal Processing*, 131, 1–17.

Chen, J., Yuan, S., and Wang, H. (2020). "On-line updating gaussian process measurement model for crack prognosis using the particle filter." *Mechanical Systems and Signal Processing*, 140, 106646.

Chib, S. (2001). "Markov Chain Monte Carlo Methods: Computation and Inference." *Handbook of Econometrics*, 5, 3569–3649.

Ching, J. Y. and Chen, Y. C. (2007). "Transitional Markov Chain Monte Carlo Method for Bayesian Model Updating, Model Class Selection, and Model Averaging." *Journal of Engineering Me-*

chanics, 133, .

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

Den-Hartog, J. P. (1930). "Forced vibrations with combined viscous and Coulomb damping." *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 9, 801–817.

Doucet, A., de Freitas, N., Gordon, N., and Smith, A. (2001). *Sequential Monte Carlo methods in practice*. New York: Springer ISBN: 978-0387951461.

Drovandi, C. C., McGree, J. M., and Pettitt, A. N. (2014). "A Sequential Monte Carlo Algorithm to Incorporate Model Uncertainty in Bayesian Sequential Design." *Journal of Computational and Graphical Statistics*, 23, 3–24.

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 Dynamics*, 89, 1489–1511.

Fan, X. and Liu, Y. (2019a). "Use of monitored daily extreme stress data for performance prediction of steel bridges: Dynamic linear models and Gaussian mixed particle filter." *Mechanical Systems and Signal Processing*, 121, 841–855.

Fan, X. and Liu, Y. (2019b). "Use of monitored daily extreme stress data for performance prediction of steel bridges: Dynamic linear models and Gaussian mixed particle filter." *Mechanical Systems and Signal Processing*, 121, 841–855.

Foreman-Mackey, D., Hogg, D. W., Lang, D., and Goodman, J. (2013). "Emcee: The mcmc hammer." *Publications of the Astronomical Society of the Pacific*, 125, 306–312.

Frank, O., Nieto, J., Guivant, J., and Scheding, S. (2003). "Multiple target tracking using Sequential Monte Carlo Methods and statistical data association." *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, 3, 2718–2723.

Gallier, J. H. (2012). *Geometric methods and applications: For computer science and engineering*. Springer Science and Business Media ISBN: 978-1461301370.

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." *Applied Mathematical Modelling*, 87, 203–221.

Goodman, J. and Weare, J. (2010). "Ensemble samplers with affine invariance." *Communications in Applied Mathematics and Computational Science*, 5, 65–80.

Green, P. and Maskell, S. (2018). "Estimating the Parameters of Dynamical Systems from Big Data using Sequential Monte Carlo Samplers." *Mechanical Systems and Signal Processing*, 93, 379–396.

Hammersley, J. M. and Handscomb, D. C. (1964). *Monte Carlo methods*. Springer Netherlands ISBN: 978-9400958197.

Hastings, W. K. (1970). "Monte Carlo Sampling Methods using Markov Chains and their Applications." *Biometrika*, 57, 97—109.

Heideman, M., Johnson, D., and Burrus, C. (1984). "Gauss and the history of the fast fourier transform." *IEEE ASSP Magazine*, 1, 14–21.

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." *The Astrophysical Journal*, 745, 198.

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)." *Mechanical Systems and Signal Processing*, 147, 107074.

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." *Mechanical Systems and Signal Processing*, 151, 107362.

Hur, S. (2021). "Short-term wind speed prediction using Extended Kalman filter and machine learning." *Energy Reports*, 7, 1046–1054.

Iglesias, M., Park, M., and Tretyakov, M. V. (2018). "Bayesian inversion in resin transfer molding." *Inverse Problems*, 34, 105002.

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 Journal on Uncertainty Quantification*, 2, 464–489.

Katafygiotis, L. S. and Beck, J. L. (1998). "Updating Models and Their Uncertainties. II: Model Identifiability." *Journal of Engineering Mechanics*, 124, 463—467.

Kim, H., Jin, C., and Kim, M. (2021). "Real-time estimation of riser's deformed shape using inclinometers and Extended Kalman Filter." *Marine Structures*, 77, 102933.

Lampart, T. (2012). "Implementation and performance comparison of an ensemble sampler with affine invariance." *Technical Report*, MOSAIC Group, Institute of Theoretical Computer Science, Department of Computer Science, ETH Zurich.

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." *Mechanical Systems and Signal Processing*, 104, 758–775.

Loan, C. V. (1992). *Computational Frameworks for the Fast Fourier Transform*. SIAM ISBN: 978-0898712858.

Lund, A., Dyke, S. J., Song, W., and Bilionis, I. (2020). "Identification of an experimental nonlinear energy sink device using the unscented Kalman filter." *Mechanical Systems and Signal Processing*, 136, 106512.

Lye, A., Cicirello, A., and Patelli, E. (2021). "Sampling Methods for solving Bayesian Model Updating Problems: A Tutorial." *Mechanical Systems and Signal Processing*, 159, 107760.

Lye, A., Cicirello, A., and Patelli, E. (2022a). "An efficient and robust sampler for Bayesian inference: Transitional Ensemble Markov Chain Monte Carlo." *Mechanical Systems and Signal Processing*, 167, 108471.

Lye, A., Cicirello, A., and Patelli, E. (2022b). "On-line Bayesian Inference for Structural Health Monitoring under Model Uncertainty using Sequential Ensemble Monte Carlo." *In the Proceedings of the 13th International Conference on Structural Safety and Reliability*, 1.

Marino, L. and Cicirello, A. (2020). "Experimental investigation of a single-degree-of-freedom system with Coulomb friction." *Nonlinear Dynamics*, 99, 1781–1799.

Marino, L., Cicirello, A., and Hills, D. A. (2019). "Displacement transmissibility of a Coulomb friction oscillator subject to joined base-wall motion." *Nonlinear Dynamics*, 98, 2595–2612.

Martino, L. and Elvira, V. (2017). "Metropolis sampling." *Wiley StatsRef: Statistics Reference Online*, , 1–18.

Moffa, G. and Kuipers, J. (2014). "Sequential Monte Carlo EM for multivariate probit models." *Computational Statistics and Data Analysis*, 72, 252–272.

Moral, P. D., Doucet, A., and Jasra, A. (2006). "Sequential Monte Carlo Samplers." *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 68, 411–436.

Moral, P. D., Doucet, A., and Jasra, A. (2007). "Sequential Monte Carlo for Bayesian Computations." *Bayesian Statistics*, 8, 1–34.

Moral, P. D., Doucet, A., and Jasra, A. (2012). "On adaptive resampling strategies for sequential Monte Carlo methods." *Bernoulli*, 18, 252–278.

Morse, L., Khodaei, Z. S., and Aliabadi, M. H. (2018). "Reliability based impact localization in composite panels using Bayesian updating and the Kalman filter." *Mechanical Systems and Signal Processing*, 99, 107–128.

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." *In Proceedings of 21st European Signal Processing Conference (EUSIPCO 2013)*, 1, 1–5.

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, 155–179.

Ou, G., Dyke, S. J., and Prakash, A. (2017). "Real time hybrid simulation with online model updating: An analysis of accuracy." *Mechanical Systems and Signal Processing*, 84B, 223–240.

Ristic, B., Arulampalam, S., and Gordon, N. (2004). *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston: Artech House ISBN: 978-1580538510.

Robert, C. P. and Casella, G. (2013). *Monte Carlo Statistical Methods*. Springer Science & Business Media, 2 edition ISBN: 978-1475741452.

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). "Weak Convergence and Optimal Scaling of

Random Walk Metropolis Algorithms." *The Annals of Applied Probability*, 7, 110–120.

Rocchetta, R., Broggi, M., Huchet, Q., and Patelli, E. (2018). "On-line Bayesian Model Updating for Structural Health Monitoring." *Mechanical Systems and Signal Processing*, 103, 174–195.

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." *Mechanical Systems and Signal Processing*, 135, 106421.

Saad, D. (2009). *On-Line Learning in Neural Networks*. Cambridge: Cambridge University Press ISBN: 978-0521117913.

Sarrka, S. (2013). *Bayesian filtering and smoothing*. Cambridge: Cambridge University Press, 4 edition ISBN: 978-1107030657.

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, 775–786.

Shields, M. D., Giovanis, D. G., and Sundar, V. S. (2021). "Subset simulation for problems with strongly non-Gaussian, highly anisotropic, and degenerate distributions." *Computers and Structures*, 245, 106431.

Silverman, B. W., Cox, D. R., and Hinkley, D. V. (1986). *Density Estimation for Statistics and Data Analysis*. Taylor and Francis, 1 edition.

Storn, R. and Price, K. (1997). "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces." *Journal of Global Optimization*, 11, 341–359.

Tatis, K. E., Dertimanis, V. K., and Chatzi, E. N. (2022). "Sequential Bayesian inference for Uncertain Nonlinear Dynamic Systems: A tutorial." *Journal of Structural Dynamics*, 1, 236–262.

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." *In Proceedings of the 11th International Conference on Information Fusion*, 1, 1–8.

Terejanu, G., Singla, P., Singh, T., and Scott, P. D. (2011). "Adaptive Gaussian Sum Filter for Nonlinear Bayesian Estimation." *IEEE Transactions on Automatic Control*, 56, 2151–2156.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. (2008). "Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems." *Journal of the Royal Society Interface*, 6, 187–202.

Urteaga, I., Bugallo, M. F., and Djuric, P. M. (2016). "Sequential Monte Carlo methods under model uncertainty." *In Proceedings of 2016 IEEE Statistical Signal Processing Workshop (SSP)*, , 1–5.

Wan, E. A. and Merwe, R. V. D. (2002). *Kalman Filtering and Neural Network*. Wiley, New York.

Wang, X., Pan, Q., Liang, Y., and Li, H. (2015). "Gaussian sum approximation filter for nonlinear dynamic time-delay system." *Nonlinear Dynamics*, 82, 501–517.

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*, , 41–115.

Weng, J. H. and Loh, C. H. (2011). "Recursive subspace identification for on-line tracking of structural modal parameter." *Mechanical Systems and Signal Processing*, 25, 2923–2937.

Xie, Z. and Feng, J. (2012). "Real-time nonlinear structural system identification via iterated unscented Kalman filter." *Mechanical Systems and Signal Processing*, 28, 309–322.

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." *Mechanical Systems and Signal Processing*, 143, 106843.

Zhou, Y., Johansen, A. M., and Aston, J. A. D. (2016). "Toward Automatic Model Comparison: An Adaptive Sequential Monte Carlo Approach." *Journal of Computational and Graphical Statistics*, 25, 701–726.

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." *Physics of the Dark Universe*, 28, .

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.” *Advances in Water Resources*, 114, 164–179.

List of Tables

Exp.	r_{nom}	$F_\mu(t_1) = 1.435\ N$		$F_\mu(t_2) = 0.980\ N$		$F_\mu(t_3) = 0.662\ N$		$F_\mu(t_4) = 0.217\ N$	
		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

TABLE 1. A Spring-Mass-Damper: Numerical results of r and ϕ $[deg]$ obtained for the respective $F_\mu(t_s)$.

Exp.	$F_\mu(t_1) = 1.435$	$F_\mu(t_2) = 0.980$	$F_\mu(t_3) = 0.662$	$F_\mu(t_4) = 0.217$
	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

TABLE 2. Shear frame structure: Numerical values of ω_b $[rad/s]$ used for the respective $F_\mu(t_s)[N]$.

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	$[0.001, 1]$	$-$

TABLE 3. Shear frame structure: Bounds of the non-informative Uniform prior for the respective inferred parameters.

	T_1		T_2	
s	E(Log-evidence)	Stdev.	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 4. Shear frame structure: Model identification using SEMC - mean of the Log-evidence ($\log\left[P(\boldsymbol{D}^{1:s}|M)\right]$), and its standard deviation (Stdev).

	T_1		T_2	
s	E(Log-evidence)	Stdev.	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

TABLE 5. Shear frame structure: Model identification using SMC - mean of the Log-evidence ($\log\left[P(\boldsymbol{D}^{1:s}|M)\right]$), and its standard deviation (Stdev).

		T_1		T_2	
s	True value	$\mathrm{E}[F_\mu(t_s)]$	Stdev.	$\mathrm{E}[F_\mu(t_s)]$	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 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	$\mathrm{E}[F_\mu(t_s)]$	Stdev.	$\mathrm{E}[F_\mu(t_s)]$	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 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$[\omega_n]$	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 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	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 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	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 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	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 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	$\mathrm{E}[\sigma_r]$	Stdev.	$\mathrm{E}[\sigma_r]$	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 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	$\mathrm{E}[\sigma_r]$	Stdev.	$\mathrm{E}[\sigma_r]$	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

TABLE 13. Shear frame structure: Parameter identification of σ_r using SMC - mean of σ_r, and its standard deviation (Stdev).

List of Figures

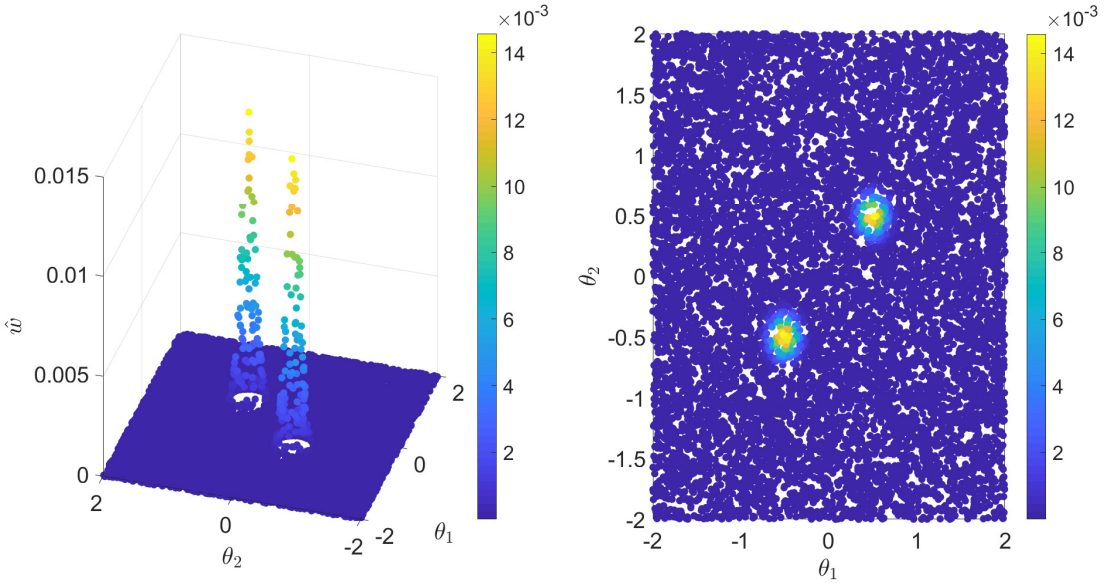

Fig. 1. Scatterplot of the prior samples, along with their associated normalised weights $\hat{w}_i^s$, obtained from the posterior consisting of a mixture of two Gaussian distributions.

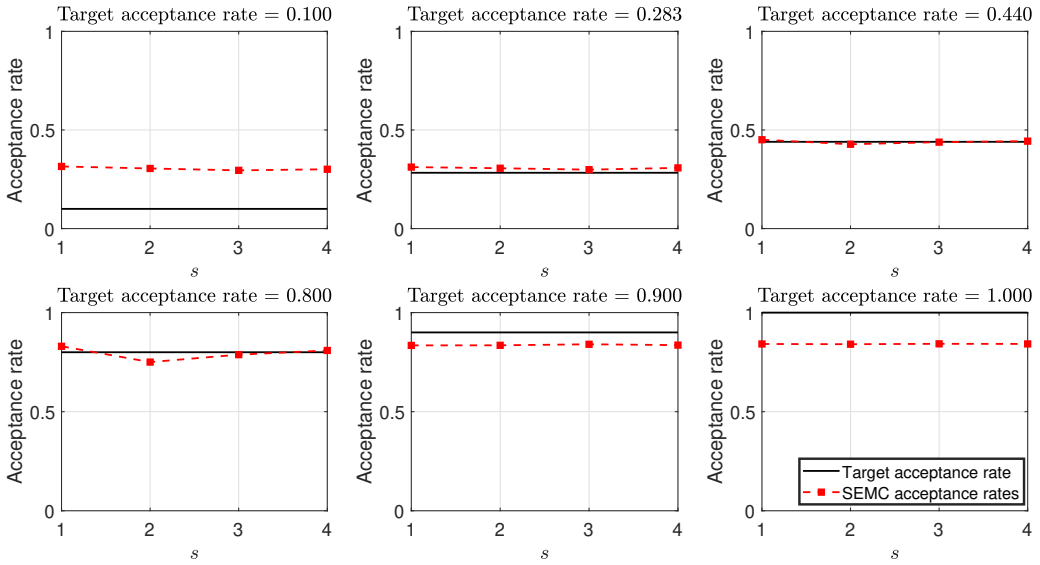

Fig. 2. Acceptance rates for the SEMC sampler: target acceptance rate values $\alpha_{tr} = \{0.100, 0.283, 0.440, 0.800, 0.900, 1.000\}$ and starting step-size values $u^{s=1} = \{40, 40, 8, 2, 2, 2\}$.

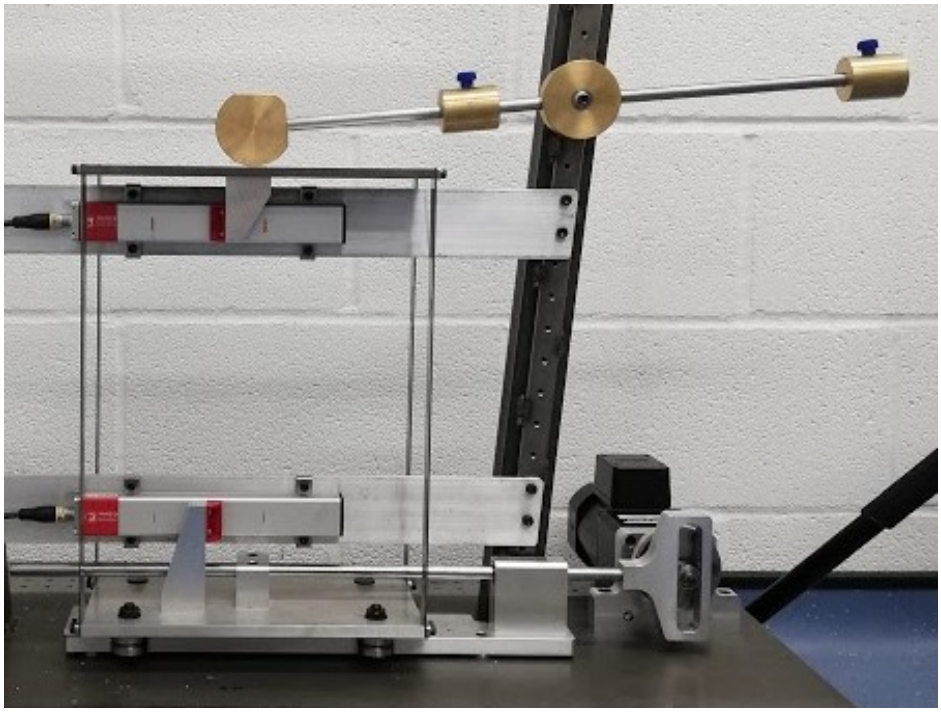

Fig. 3. SDoF single-storey shear frame structure subjected to Coulomb friction.

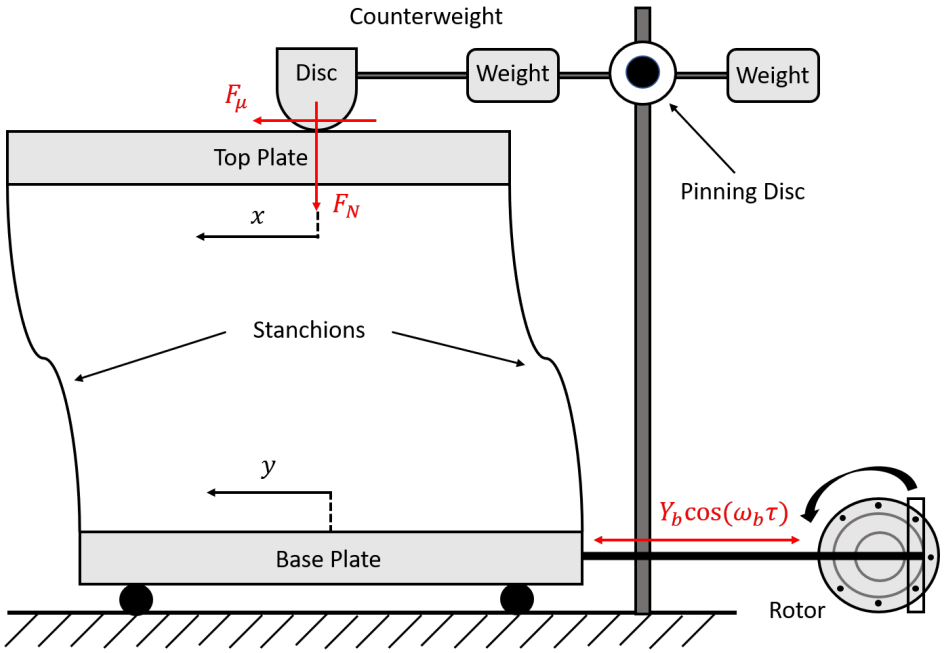

Fig. 4. Schematic diagram of the SDoF single-storey shear frame structure subjected to Coulomb friction. Image adapted from (Marino and Cicirello 2020).

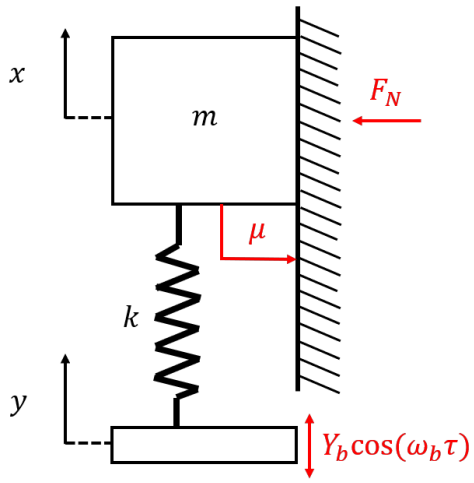

Fig. 5. Spring-mass representation of the SDoF single-storey shear frame structure subjected to Coulomb friction.

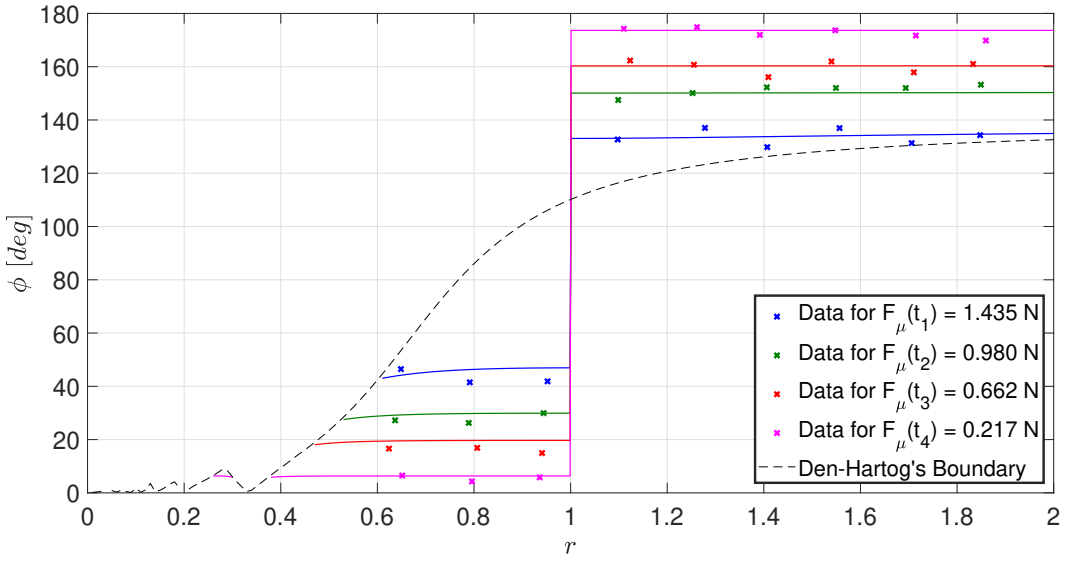

Fig. 6. Shear frame structure: Plots of r and ϕ for the corresponding values of $F_\mu(t_s)$ for $s = \{1, \ldots, 4\}$.

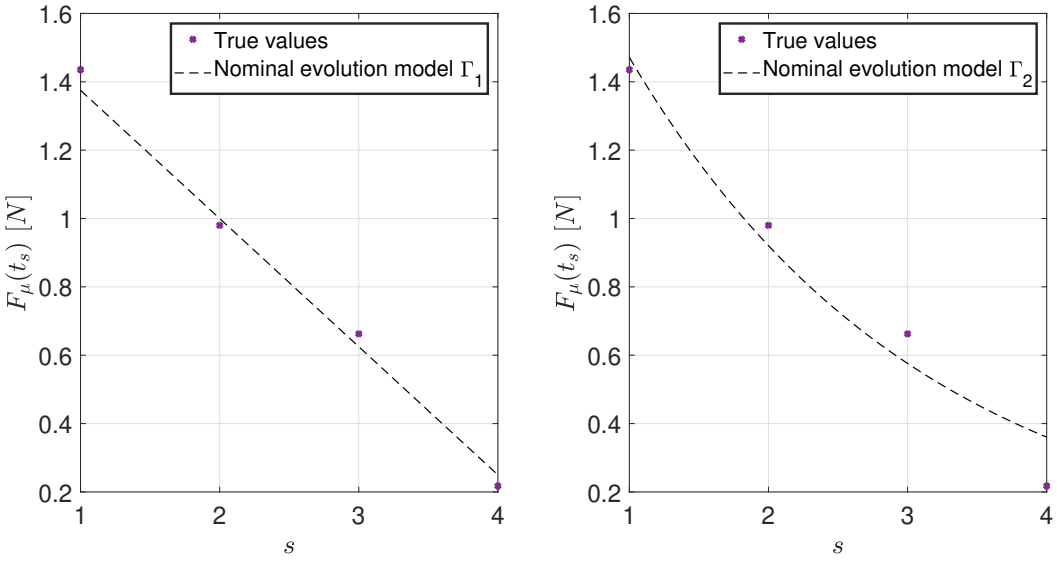

Fig. 7. Shear frame structure: Nominal evolution models Γ_1 and Γ_2.

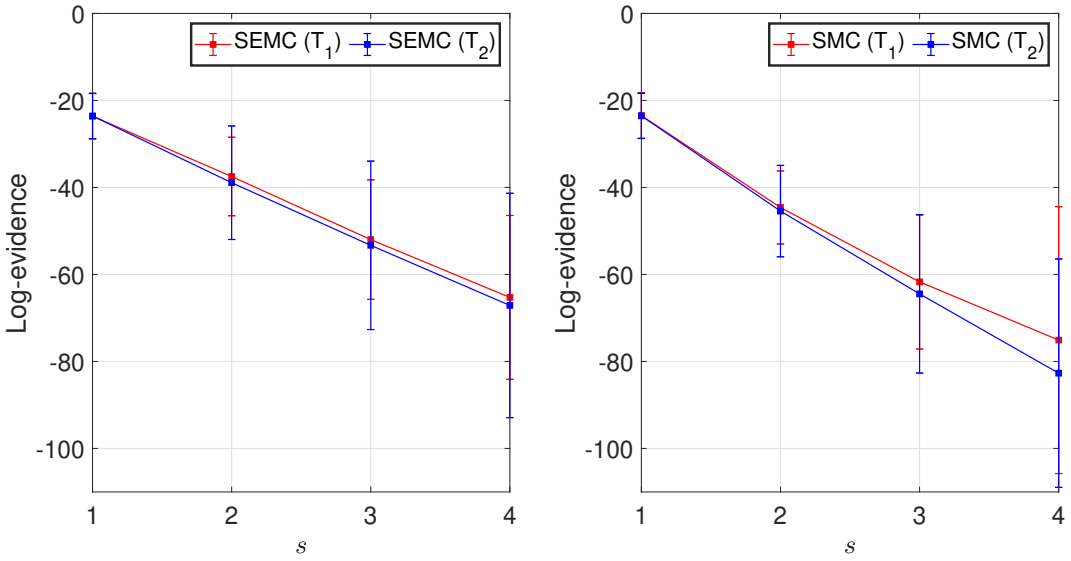

Fig. 8. Shear frame structure: Model identification results using SEMC and SMC - mean of the Log-evidence ($\log\left[P(\boldsymbol{D}^{1:s}|M)\right]$), and its 1-sigma bounds.

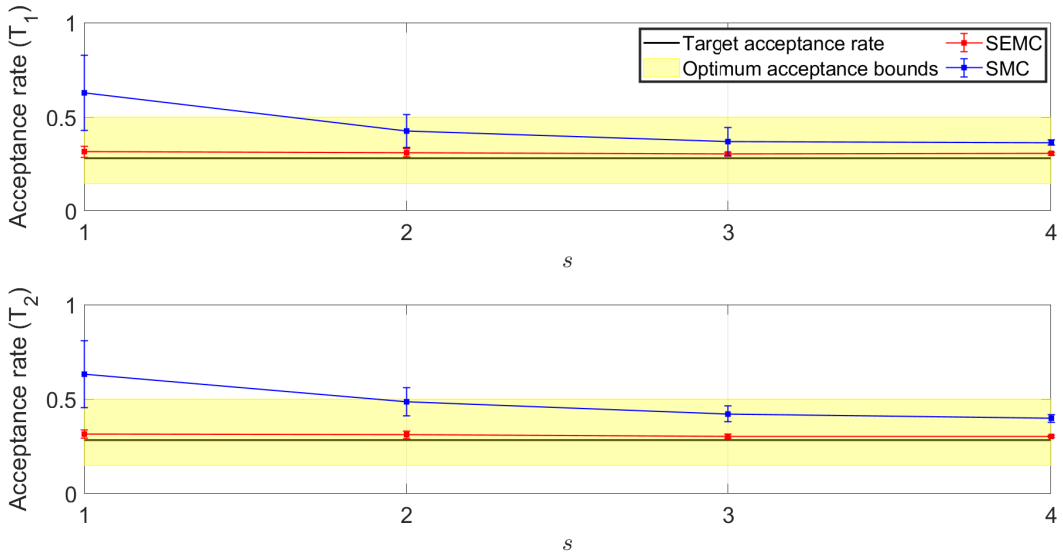

Fig. 9. Shear frame structure: Acceptance rates results using SEMC and SMC - mean of the acceptance rates, and its 1-sigma bounds. Target acceptance rate: 0.283 (i.e. see Eq. (10)).

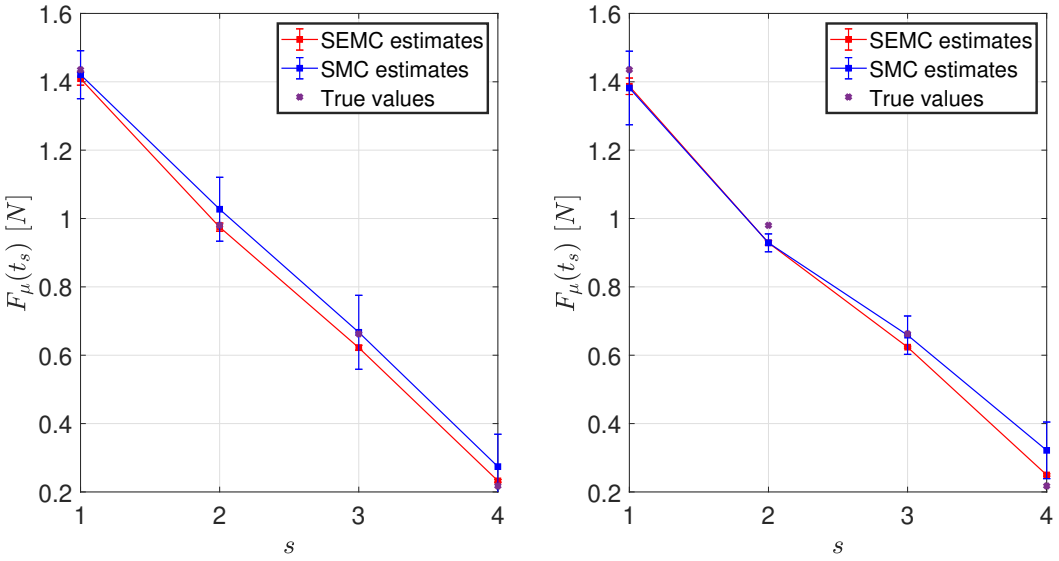

Fig. 10. Shear frame structure: Parameter identification of $F_\mu(t_s)$ $[N]$ using SEMC and SMC - mean of $F_\mu(t_s)$, and its 1-sigma bounds.

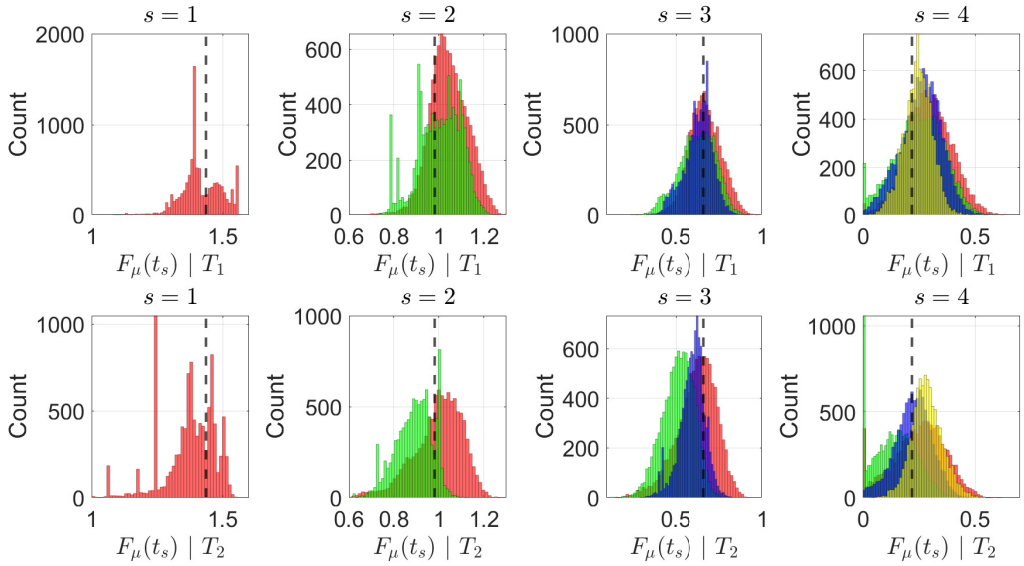

Fig. 11. Shear frame structure: Posterior sample histogram profiles of the predicted values of $F_\mu(t_s)$ $[N]$ using SEMC. The black dotted vertical line denotes the true value of $F_\mu(t_s)$ at a given s.

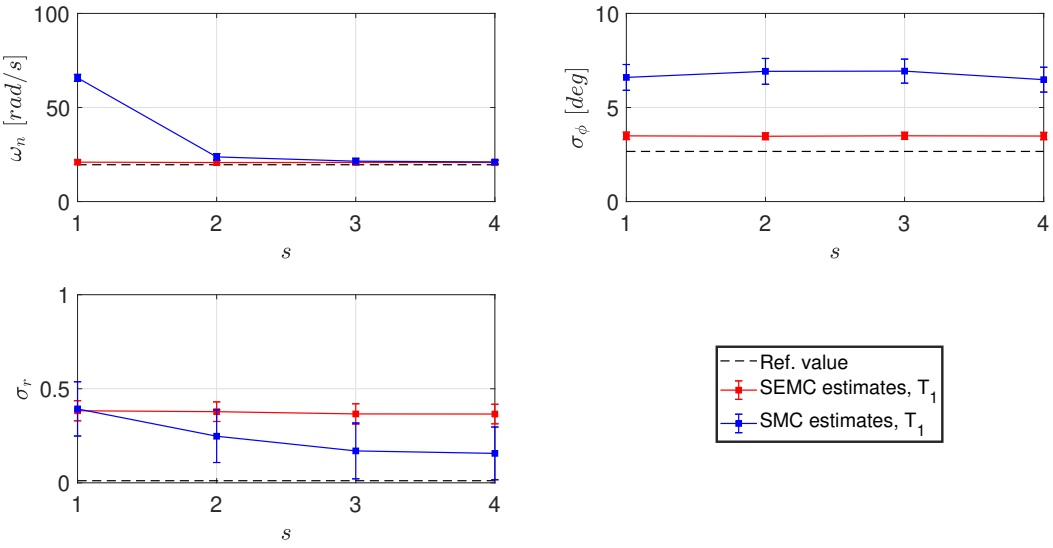

Fig. 12. Shear frame structure: Parameter identification of ω_n $[rad/s]$, σ_ϕ $[deg]$, and σ_r using SEMC and SMC given T_1 - their corresponding means, and 1-sigma bounds. The reference values for the respective parameters are: $\{\omega_n, \sigma_\phi, \sigma_r\} = \{19.572 \text{ rad/s}, 2.664^o, 0.011\}$.

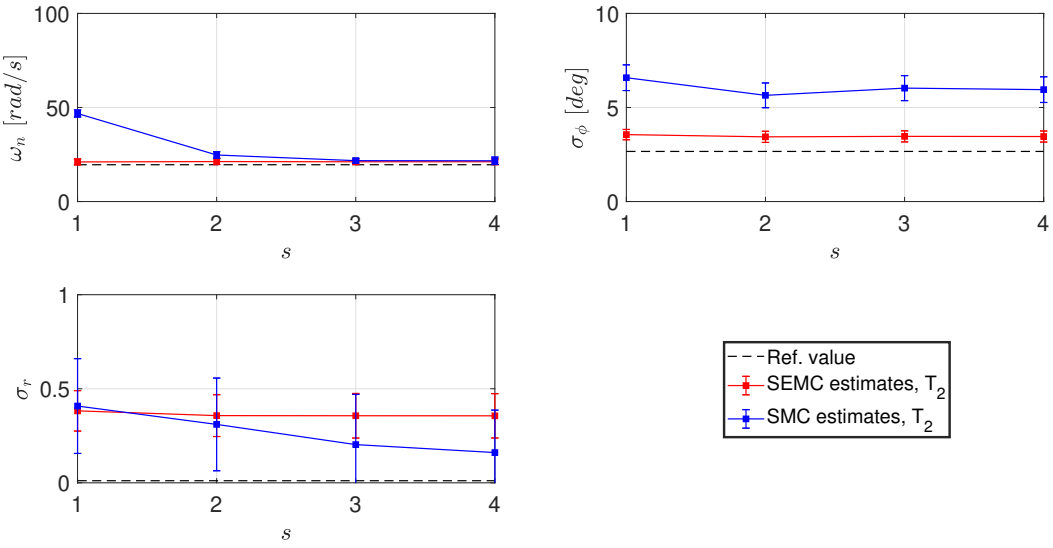

Fig. 13. Shear frame structure: Parameter identification of ω_n $[rad/s]$, σ_ϕ $[deg]$, and σ_r using SEMC and SMC given T_2 - their corresponding means, and 1-sigma bounds. The reference values for the respective parameters are: $\{\omega_n, \sigma_\phi, \sigma_r\} = \{19.572 \text{ rad/s}, 2.664^o, 0.011\}$.