Probabilistic Engineering Mechanics 42 (2015) 54–63

Contents lists available at ScienceDirect

Probabilistic Engineering Mechanics

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

Bayesian system identification of dynamical systems using large sets of training data: A MCMC solution

P.L. Green [1]

Department of Mechanical Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, United Kingdom

ARTICLE INFO

Article history:
Received 17 April 2015
Accepted 17 September 2015
Available online 25 September 2015

Keywords:
Nonlinear system identification
Markov chain Monte Carlo
Bayesian inference
Smooth data annealing
Big data

ABSTRACT

In the last 20 years the applicability of Bayesian inference to the system identification of structurally dynamical systems has been helped considerably by the emergence of Markov chain Monte Carlo (MCMC) algorithms – stochastic simulation methods which alleviate the need to evaluate the intractable integrals which often arise during Bayesian analysis. In this paper specific attention is given to the situation where, with the aim of performing Bayesian system identification, one is presented with very large sets of training data. Building on previous work by the author, an MCMC algorithm is presented which, through combing Data Annealing with the concept of 'highly informative training data', can be used to analyse large sets of data in a computationally cheap manner. The new algorithm is called Smooth Data Annealing.

1. Introduction

1.1. A Bayesian approach

Bayesian inference involves assessing the relative plausibility of a set of model structures $\boldsymbol{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots\}$ – as well as the parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^{N_\theta}$ within each model – using a combination of one's prior knowledge and a set of training data, $\mathcal{D}$. By virtue of influential papers such as [1] it is now well-established in the structural dynamics community that both levels of inference (parameter estimation and model selection) can be addressed via the sequential application of Bayes' theorem:

$$P(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}) = \frac{P(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M})P(\boldsymbol{\theta}|\mathcal{M})}{P(\mathcal{D}|\mathcal{M})} \tag{1}$$

$$P(\mathcal{M}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{M})P(\mathcal{M})}{P(\mathcal{D})}. \tag{2}$$

Evaluation of Eq. (1) requires the definition of the prior, $P(\boldsymbol{\theta}|\mathcal{M})$, and the likelihood, $P(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M})$. The prior is a subjective probability distribution which describes one's knowledge of the parameters before the data was known. The likelihood describes the probability of witnessing the data according to model, $\mathcal{M}$, with parameters, $\boldsymbol{\theta}$. As such, the likelihood is defined by a 'prediction-error model' (see [2] for a comprehensive discussion). The denominator of Eq. (1) – the 'model evidence' – is a normalising constant which ensures that $P(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M})$ integrates to unity. Successful evaluation of Eq. (1) gives one the posterior parameter distribution, which describes the probability of parameter vector, $\boldsymbol{\theta}$, given the training data, $\mathcal{D}$, and the chosen model structure, $\mathcal{M}$.

With regard to Eq. (2), $P(\mathcal{M})$ is a probability mass function which describes one's prior belief in model $\mathcal{M}$, $P(\mathcal{D})$ is a normalising constant and $P(\mathcal{D}|M)$ is equal to the evidence term on the denominator of Eq. (1). $P(\mathcal{M}|\mathcal{D})$ is a distribution describing the relative probability of different competing model structures conditional on the data, $\mathcal{D}$. One of the advantages of the Bayesian approach to model selection is that overly complex models tend to be assigned relatively low probabilities, thus preventing over-fitting (see [3–5] for more information). Furthermore, via Eqs (1) and (2), one is able to quantify and propagate the inevitable uncertainties involved in the parameter estimation and model selection processes.

1.2. Why MCMC?

It is often the case that one wishes to generate samples from $P(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M})$ as part of a Monte Carlo analysis. With the geometry of the posterior parameter distribution often being fairly complex, this is usually impossible to achieve using well-known methods such as inverse transform sampling. Additionally, Monte Carlo methods such as importance sampling and rejection sampling are difficult to apply as the density of the posterior parameter distribution tends to be concentrated in a small region of the

E-mail address: p.l.green@liverpool.ac.uk

[1] Current address: Institute for Risk and Uncertainty, Centre for Engineering Sustainability, School of Engineering, University of Liverpool, Liverpool L69 3GQ, United Kingdom.

http://dx.doi.org/10.1016/j.probengmech.2015.09.010
0266-8920/

parameter space relative to the prior. Furthermore, the model evidence – which is found by integrating the posterior parameter distribution across the entire parameter space – is often difficult to obtain in a closed-form manner and, due to the large computational cost involved, cannot usually be evaluated numerically.

Markov chain Monte Carlo (MCMC) methods involve the evolution of an ergodic Markov chain whose stationary distribution is *proportional* to the posterior parameter distribution. By allowing one's Markov chain to become stationary, MCMC can be used to generate (dependent) samples from $P(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M})$ while circumventing the need to calculate the model evidence. 'Traditional' methods include the well-known Metropolis [6] and Hybrid Monte Carlo algorithms [7]. Presently, more advanced MCMC algorithms are available which are able to generate samples from the posterior parameter distribution *and* estimate the model evidence/generate samples from the posterior model distribution simultaneously – these include Reversible Jump MCMC [8], Transitional MCMC (TMCMC) [9], Nested Sampling [10] and Asymptotically Independent Markov Sampling (AIMS) [11]. TMCMC in particular has become popular within the context of mechanical engineering, as it is able sample from distributions with complex geometries and is suitable for parallelisation [12].

Of specific relevance here is the concept of combining MCMC methods with the well-known Simulated Annealing algorithm [13]. This involves using MCMC to target a sequence of distributions defined by

$$\pi_{\beta_j}(\boldsymbol{\theta}) \propto P(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M})^{\beta_j} P(\boldsymbol{\theta}|\mathcal{M}), \quad j = 1, 2, \ldots, N_\beta \tag{3}$$

where

$$0 = \beta_1 < \beta_2 < \cdots < \beta_{N_\beta} = 1 \tag{4}$$

The result is that, by increasing β (the inverse temperature), one is inducing a gradual transition from the prior to the posterior parameter distributions. This technique has proved to be extremely useful and forms a fundamental part of the TMCMC [9] and AIMS [11] algorithms (as well as many others).

It is important to note that the strictly increasing sequence of β values – the annealing schedule – is crucial to the success of any MCMC algorithm which targets the sequence of distributions described by Eq. (3).

1.3. Motivation

The current paper is motivated by the situation where, as part of some collaborative work, one is presented with a very large set of training data from which the relative probability of various parameters/models are to be inferred (this is sometimes referred to as a 'Big Data' issue). In such situations one often finds that, despite the savings that can be achieved via parallelisation, the computational cost of MCMC dictates that only a small subset of the 'full' data can be utilised.

A possible solution to this problem is to use the Data Annealing algorithm [14]. Noting that, when employing a variant of Simulated Annealing, one is essentially using β to modulate (and increase) the influence of the data on the target distribution, Data Annealing achieves a similar result simply via the gradual introduction of data points into the likelihood. This involves targeting the distribution

$$\pi(\boldsymbol{\theta}) \propto P(\mathcal{D}_1^N|\boldsymbol{\theta}, \mathcal{M}) P(\boldsymbol{\theta}|\mathcal{M}) \tag{5}$$

where $\mathcal{D}_1^N = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N\}$, using MCMC (a standard Metropolis update was employed in [14]). Once a sufficient number of samples have been generated, N can then be increased such that additional data points are included in the likelihood. This process is repeated until the statistical properties of one's parameter estimates are judged to have converged. To ensure efficient MCMC performance a proposal density can be chosen whose covariance matrix (assuming a Gaussian proposal is being utilised) is a fraction of the distribution which was most recently targeted. Alternatively, as demonstrated in [14], it is possible to achieve satisfactory results simply by using a heavy-tailed proposal distribution. While Data Annealing tends to be fast (as the model does not have to reproduce the entire set of data every time a sample is generated), one has little control over the rate at which the *information* in the data is introduced into the likelihood.

A second option would be to utilise the approach described in [15], where the approximate information content of data sets was measured. This can then allow one to select a small, highly informative subset of data from which to infer parameter estimates.

This is achieved by first writing the posterior parameter parameter distribution as

$$P(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}) \propto \exp(-J(\boldsymbol{\theta})) \tag{6}$$

and employing a second order Taylor series expansion about the most-probable parameter estimate, $\hat{\boldsymbol{\theta}}$, to gain

$$J(\boldsymbol{\theta}) \approx J(\hat{\boldsymbol{\theta}}) + \frac{1}{2}\boldsymbol{\Delta\theta}^T \boldsymbol{A} \boldsymbol{\Delta\theta} \tag{7}$$

where $\boldsymbol{A} = \nabla\nabla J(\hat{\boldsymbol{\theta}})$ and $\boldsymbol{\Delta\theta} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}$. From this a Gaussian approximation of the posterior can be obtained

$$P^*(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}) = \frac{1}{Z}\exp\left(-\frac{1}{2}\boldsymbol{\Delta\theta}^T \boldsymbol{A} \boldsymbol{\Delta\theta}\right), \quad Z = \sqrt{\frac{(2\pi)^{N_\theta}}{|\boldsymbol{A}|}}. \tag{8}$$

The information content of this distribution can then be measured using the Shannon entropy:

$$S = -\int P^*(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}) \log P^*(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M})\, d\boldsymbol{\theta} = \frac{1}{2}\log\left(\frac{(2\pi e)^{N_\theta}}{|\boldsymbol{A}|}\right) \tag{9}$$

whose properties as an information measure are well known [16]. Eq. (9) can then be used to estimate the influence of the available training data on the information content of the posterior.[2] A drawback of this method is that it relies on one knowing the location of $\hat{\boldsymbol{\theta}}$ before the analysis can begin.

The algorithm proposed in the current paper encompasses elements from Data Annealing and the concept of highly informative training data. It is designed to overcome the drawbacks of both the afore-mentioned methodologies and is suitable for dynamical models (see [17] for a solution which can be applied to static models).

2. Smooth data annealing

2.1. Basic methodology

As in the previous section, $\mathcal{D}_1^N$ denotes the data $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N\}$. With Smooth Data Annealing (SDA), one begins by targeting the distribution:

$$\pi_{\beta_j}(\boldsymbol{\theta}|\mathcal{D}_1^N) \propto P(\mathcal{D}_1^N|\boldsymbol{\theta})^{\beta_j} P(\boldsymbol{\theta}) \tag{10}$$

where N is a predefined integer, $\mathcal{D}_1^N$ is a small subset of the available training data (choice of N is discussed in Section 4) and, from now on, dependence on model structure is omitted. By increasing β one can then 'anneal in' the data $\mathcal{D}_1^N$ in the usual manner. Once $\beta = 1$ then one can choose to add an additional k

[2] It can also be used to analyse the influence of the data on the relative probability of competing model structures, although this is not considered directly in the current work.

data points and redefine the target distribution as

$$\pi_{\beta_j}(\boldsymbol{\theta}|\mathcal{D}_1^{N+k}) \propto P(\mathcal{D}_1^N|\boldsymbol{\theta})P(\mathcal{D}_{N+1}^{N+k}|\boldsymbol{\theta})^{\beta_j}P(\boldsymbol{\theta}). \tag{11}$$

(assuming that $P(\mathcal{D}_1^N|\boldsymbol{\theta})$ and $P(\mathcal{D}_{N+1}^{N+k}|\boldsymbol{\theta})$ are mutually independent). This process can then be repeated until certain criteria are met. SDA therefore has all the advantages of Data Annealing, while also giving the user complete control of the rate at which the influence of the data is introduced. The choice of annealing schedule is discussed in the next section.

2.2. Constant entropy variation

Here it is hypothesised that the optimum annealing schedule is one in which the information content, measured using the Shannon entropy, varies at a constant rate. This allows the concept of only using highly informative training data [15] to become an inherent feature of SDA – data which has little influence with regard to one's parameter uncertainty is annealed in quickly, thus allowing the algorithm to focus on the data which is 'information rich'.

For the remaining part of this section it is advantageous for the target PDF (Eq. (11)) to be written in the following form:

$$\pi = \frac{\exp(-\beta_j\hat{J}_L - J_L - J_P)}{Z} \tag{12}$$

where

$$\hat{J}_L = -\ln\left(P(\mathcal{D}_{N+1}^{N+k}|\boldsymbol{\theta})\right), \quad J_L = -\ln\left(P(\mathcal{D}_1^N|\boldsymbol{\theta})\right) \tag{13}$$

and J_P is the negative log-prior.[3] Furthermore, the target distribution is written as $\pi = \pi^*/Z$ such that Z is the normalising constant of the unnormalised distribution π^*.

Before further discussion it is convenient to first derive the following properties:

$$\frac{d\pi^*}{d\beta_j} = -\hat{J}_L\pi^* \quad \frac{dZ}{d\beta_j} = -Z\mathrm{E}[\hat{J}_L] \quad \frac{d\pi}{d\beta_j} = (\mathrm{E}[\hat{J}_L] - \hat{J}_L)\pi, \tag{14}$$

(see Appendix A). The Shannon entropy of the target distribution is

$$S = \beta_j\mathrm{E}[\hat{J}_L] + \mathrm{E}[J_L] + \mathrm{E}[J_P] + \ln(Z) \tag{15}$$

such that the task is to evaluate

$$\frac{dS}{d\beta_j} = \frac{d}{d\beta_j}\left(\beta_j\mathrm{E}[\hat{J}_L]\right) + \frac{d}{d\beta_j}(\mathrm{E}[J_L]) + \frac{d}{d\beta_j}(\ln(Z)) \tag{16}$$

(noting the J_P is not a function of β_j). Using the properties in Eq. (14), the first term in Eq. (16) can be evaluated as follows:

$$\frac{d}{d\beta_j}\left(\beta_j\mathrm{E}[\hat{J}_L]\right) = \mathrm{E}[\hat{J}_L] + \beta_j\frac{d}{d\beta_j}\int\hat{J}_L\,\pi\,d\boldsymbol{\theta} \tag{17}$$

$$\frac{d}{d\beta_j}\left(\beta_j\mathrm{E}[\hat{J}_L]\right) = \mathrm{E}[\hat{J}_L] + \beta_j\left(\mathrm{E}^2[\hat{J}_L] - \mathrm{E}[\hat{J}_L^2]\right) \tag{18}$$

$$\frac{d}{d\beta_j}\left(\beta_j\mathrm{E}[\hat{J}_L]\right) = \mathrm{E}[\hat{J}_L] - \beta_j\,\mathrm{Var}(\hat{J}_L). \tag{19}$$

The second term in Eq. (16) is

Table 1
System identification of a simulated Duffing oscillator: true parameter values and moments of prior distributions.

Parameter	True value	Prior mean	Prior standard deviation	Units
k	100	150	30	N/m
c	0.05	0.02	0.02	N s/m
k_3	100,000	40,000	20,000	N/m^3
σ	0.005	0.0045	0.002	–

$$\frac{d}{d\beta_j}(\mathrm{E}[J_L]) = \frac{d}{d\beta_j}\int J_L\,\pi\,d\boldsymbol{\theta} \tag{20}$$

$$\frac{d}{d\beta_j}(\mathrm{E}[J_L]) = \int J_L(\mathrm{E}[\hat{J}_L] - \hat{J}_L)\pi\,d\boldsymbol{\theta} \tag{21}$$

$$\frac{d}{d\beta_j}(\mathrm{E}[J_L]) = \mathrm{E}[J_L]\mathrm{E}[\hat{J}_L] - \mathrm{E}[J_L\hat{J}_L] \tag{22}$$

$$\frac{d}{d\beta_j}(\mathrm{E}[J_L]) = -\mathrm{Cov}(J_L, \hat{J}_L) \tag{23}$$

where $\mathrm{Cov}(J_L, \hat{J}_L)$ is the covariance between J_L and $\hat{J}_L$. Finally, the third term in Eq. (16) is

$$\frac{d}{d\beta_j}(\ln Z) = \frac{1}{Z}\left(-Z\mathrm{E}[\hat{J}_L]\right) \tag{24}$$

$$\frac{d}{d\beta_j}(\ln Z) = -\mathrm{E}[\hat{J}_L]. \tag{25}$$

Combining Eqs. (19), (23) and (25) one finds that

$$\frac{dS}{d\beta_j} = -\beta_j\,\mathrm{Var}(\hat{J}_L) - \mathrm{Cov}(J_L, \hat{J}_L). \tag{26}$$

Consequently, if the algorithm is currently using the value β_j and one wishes to 'anneal in' new data with a constant change in the Shannon entropy, ΔS, then β_{j+1} should be selected according to

$$\beta_{j+1} = \beta_j - \frac{\Delta S}{\beta_j\,\mathrm{Var}(\hat{J}_L) + \mathrm{Cov}(J_L, \hat{J}_L)}. \tag{27}$$

If one considers the initial stages of the algorithm (where only the first set of data is being annealed) then Eq. (27) simplifies to

$$\beta_{j+1} = \beta_j - \frac{\Delta S}{\beta_j\,\mathrm{Var}(\hat{J}_L)}. \tag{28}$$

It is important to note that, to avoid numerical issues, it is often beneficial to initiate the annealing schedule by selecting a value of β which is close to, but not equal to zero. By choosing a small initial β one is ensuring that the geometry of the first target distribution is similar to that of the prior (this will allow efficient sampling from the target using MCMC). A more sophisticated approach could involve using the methods described in [15] to estimate the Shannon entropy of this first target distribution, thus ensuring that this initial choice of β has not led to a large change in the Shannon entropy. Throughout this work it was found that initially setting $\beta = 1 \times 10^{-4}$ yielded acceptable results.

[3] Working with the log-likelihood and log-prior is not only mathematically convenient but also allows one to avoid the numerical overflow/underflow issues which frequently arise when one is analysing large data sets.

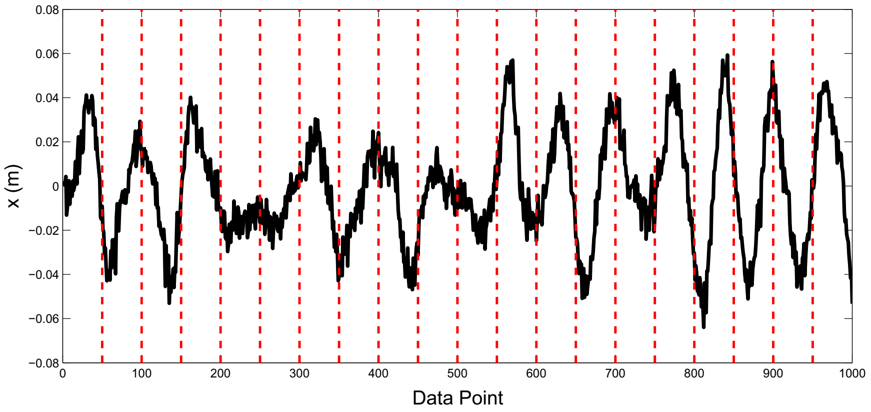

Fig. 1. Training data for the system identification of a simulated Duffing oscillator. Dashed lines indicate the segments of data which were used in the SDA algorithm. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

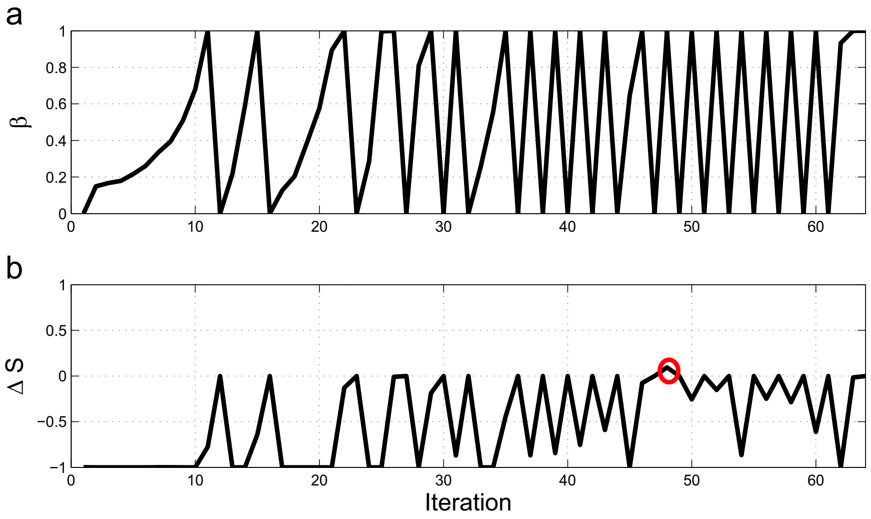

Fig. 2. Parameter estimation of a simulated Duffing oscillator: variation of (a) inverse temperature and (b) the change in Shannon entropy as training data is added to the SDA algorithm. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

2.3. *Does data reduce entropy ?*

Typically one would choose ΔS to be negative because, as the influence of the data is increased, one wishes to see a reduction in parameter uncertainty. Referring to Eq. (28) it is clear that, when the algorithm is initialised, β_{j+1} must always be larger than β_j as ΔS is negative. However, in the general case (Eq. (27)), if the covariance between J_L and $\hat{J}_L$ is negative then by imposing that $\Delta S < 0$ one can actually select a value β_{j+1} which is *lower* than β_j. This prompts one ask whether the addition of new data will necessarily reduce parameter uncertainty.

Intuitively the answer appears to be no – parameter uncertainty could increase in the situation where the new data contradicts the information in the old data (which is also when $\mathrm{Cov}(J_L, \hat{J}_L)$ will be negative). To address this issue one can use the well-known result

$$\mathrm{E}[\mathrm{Var}(\boldsymbol{\theta}|\mathcal{D})] = \mathrm{Var}(\boldsymbol{\theta}) - \mathrm{Var}(\mathrm{E}[\boldsymbol{\theta}|\mathcal{D}]) \tag{29}$$

which states that, *on average*, the variance of the posterior must be less than that of the prior. Consequently, if new data does contradict the information in the old data, one can simply allow the Shannon entropy to increase (safe in the knowledge that, on average, the increasing influence of more data must ultimately lead to a decrease in parameter uncertainty). The key is to ensure that the Shannon entropy always remains between some predefined limits, such that the transition from prior to posterior is still conducted in a gradual manner.

To be specific, β_{j+1} should be selected according to

$$\beta_{j+1} = \beta_j - \frac{\Delta S}{\beta_j \,\mathrm{Var}(\hat{J}_L) + \mathrm{Cov}(J_L, \hat{J}_L)} \tag{30}$$

but *subject to the conditions that*

$$\beta_j < \beta_{j+1} \le 1, \quad -\Delta S_{\mathrm{lim}} < \Delta S < \Delta S_{\mathrm{lim}} \tag{31}$$

where ΔS_{lim} is defined by the user.

3. Algorithm

The method by which SDA anneals in the data $\mathcal{D}_{N+1}^{N+k}$ is summarised here using pseudo-code:

```
• Set $j=1$, $\beta_j = \beta_{\mathrm{initial}}$
• While $\beta_j < 1$
    ○ Generate samples $\{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N_s)}\}$ from $\pi_{\beta_j}(\boldsymbol{\theta}|\mathcal{D}_1^{N+k}) \propto \exp(-\beta_j \hat{J}_L - J_L - J_P)$ using MCMC
    ○ Estimate $\mathrm{Var}(\hat{J}_L)$ and $\mathrm{Cov}(\hat{J}_L, J_L)$
    ○ Set $\beta_{j+1} = \beta_j - \frac{\Delta S}{\beta_j \mathrm{Var}(\hat{J}_L) + \mathrm{Cov}(J_L, \hat{J}_L)}$ subject to the conditions that
      $\beta_j < \beta_{j+1} \le 1$ and $-\Delta S_{\mathrm{lim}} < \Delta S < \Delta S_{\mathrm{lim}}$
```

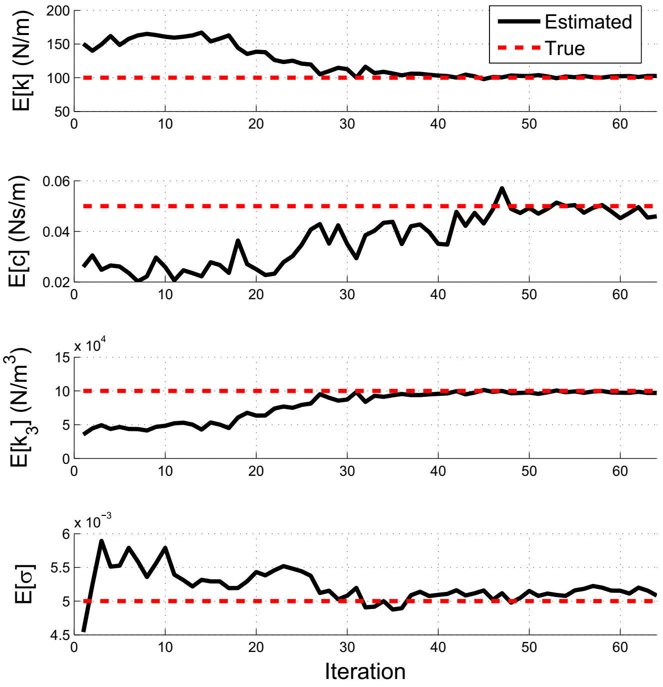

Fig. 3. Parameter estimation of a simulated Duffing oscillator: convergence of parameter estimates to true values as SDA algorithm is run.

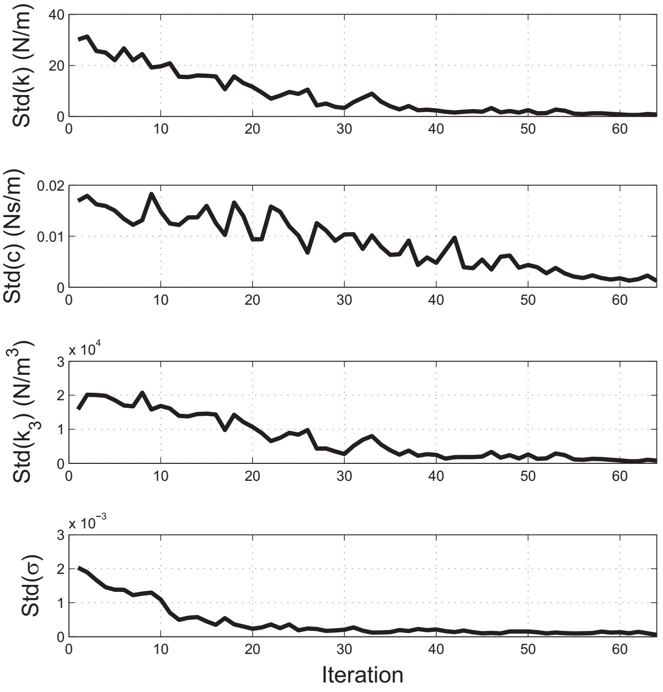

Fig. 4. Parameter estimation of a simulated Duffing oscillator: reduction of posterior parameter standard deviation as SDA algorithm is run.

```
    ○ $j = j + 1$
● End
● If Stopping criteria met
    ○ Terminate algorithm
● else
    ○ Add more data by setting $N = N + k$
● End
```

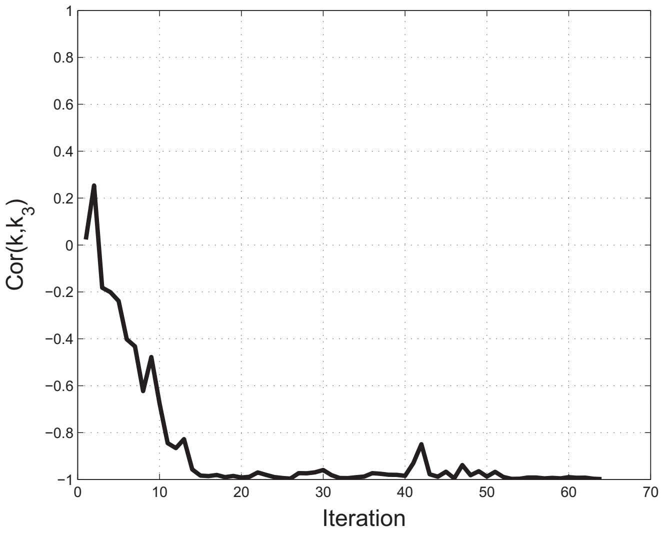

Fig. 5. Parameter estimation of a simulated Duffing oscillator: variation of the (normalised) correlation coefficient between k and k_3 as training data is added.

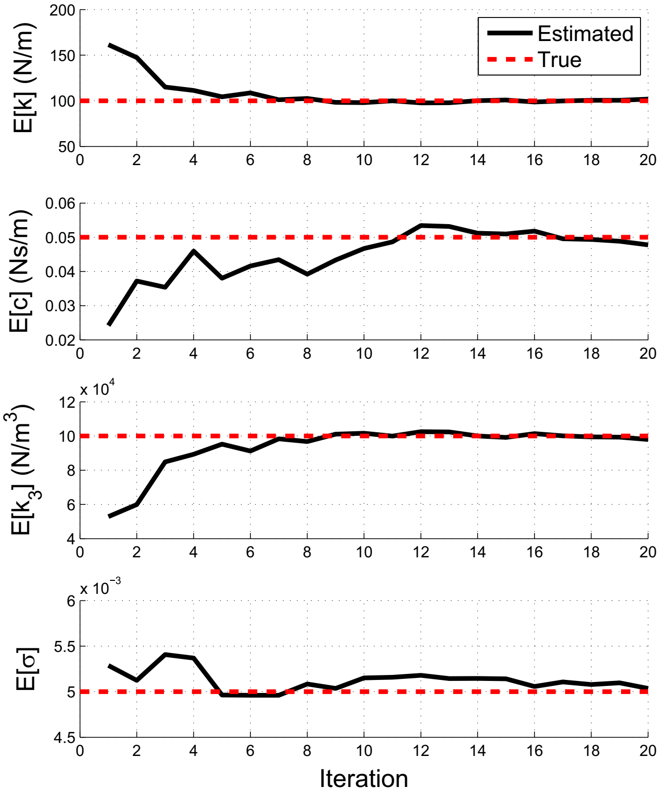

Fig. 6. Parameter estimation of a simulated Duffing oscillator: convergence of parameter estimates to true values as the Data Annealing algorithm is run.

It should be noted that when samples are being generated from $\pi_{\beta_j}(\boldsymbol{\theta}|\mathcal{D}_1^{N+k})$, any MCMC algorithm can be employed. While the Metropolis algorithm was utilised in the current paper, it should be relatively easy to incorporate more advanced MCMC methods as part of SDA. One could, for example, combine SDA with TMCMC [9].

4. Example 1 – simulated data

As an initial example, a time history of displacement data x was

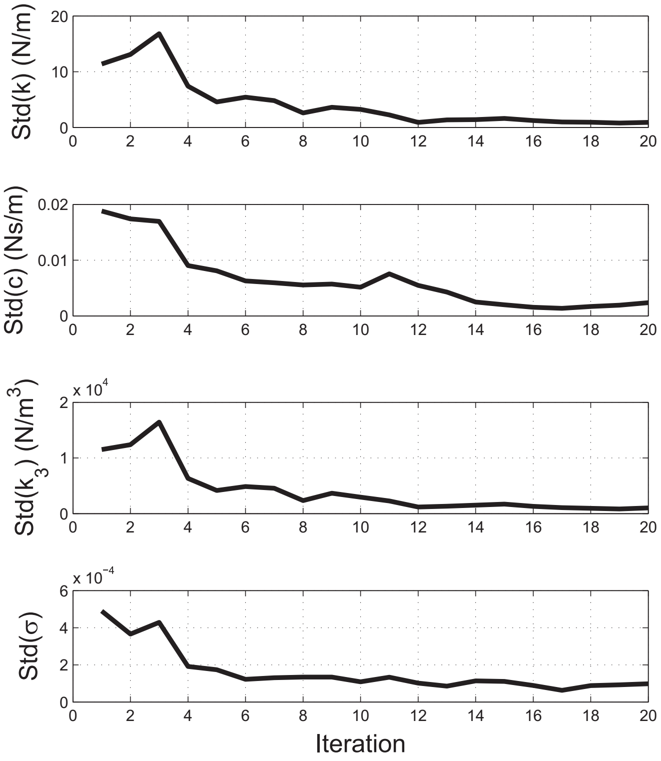

Fig. 7. Parameter estimation of a simulated Duffing oscillator: reduction of posterior parameter standard deviation as the Data Annealing algorithm is run.

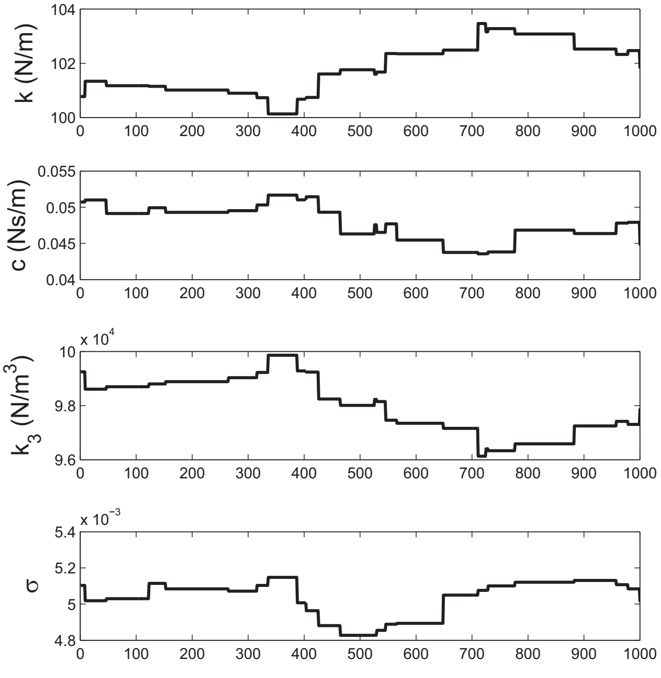

Fig. 8. Evolution of Markov chains used in the Data Annealing algorithm.

created by simulating the response of a Duffing oscillator:

$$m\ddot{x} + c\dot{x} + kx + k_3x^3 = F, \tag{32}$$

where F was a Gaussian white noise force. The time history was then artificially corrupted with Gaussian measurement noise of standard deviation σ. The mass m was set equal to 0.1 and was assumed to be known. The parameters c, k, k_3 and σ were left as parameters to be estimated. Throughout this example Gaussian distributions, truncated at zero, were used as priors. To study the convergence properties of SDA, the mean values of the priors were deliberately set to be different from the true parameter values (see Table 1).

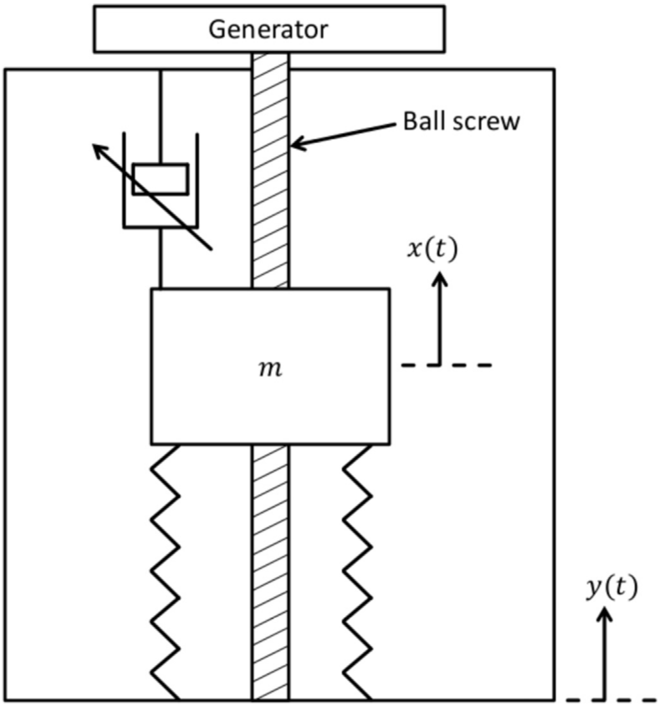

Fig. 9. Schematic of rotational energy harvester.

The 'full' set of training data consisted of 1000 displacement measurements – this is shown in Fig. 1. Of this set, the data was introduced to SDA in segments of 50 points at a time (these segments of data are separated by dashed red lines in Fig. 1).

Setting $\Delta S_{\text{lim}} = 1$ the SDA algorithm was run, generating 1000 samples at each iteration. Fig. 2 shows the resulting variation in β and ΔS. It should be noted that a new segment of data is introduced every time β has reached a value of one. Each point on the horizontal axis of Fig. 2 therefore represents an iteration of the algorithm (not the introduction of a new segment of data). It is clear that, after 35 iterations (where 350 points have been analysed), the remaining data appears to be relatively uninformative and the desired change in Shannon entropy can be realised by instantly setting $\beta = 1$. It is interesting to observe that, in the algorithm's 48th iteration (marked with a red circle in Fig. 2), a slight increase in Shannon entropy occurred. For the most part however, for this simple example, it appears that the introduction of more data has consistently reduced parameter uncertainty.

As mentioned previously, one can simply allow the SDA algorithm to run until certain criteria are met (so long as training data is still available). Fig. 3 shows how the posterior mean estimates of each parameter converged to their true values while, in Fig. 4, one can see how the posterior standard deviation of each parameter estimate reduced while additional data was being analysed. Furthermore, as MCMC can be used to approximate the posterior parameter covariance matrix [18], one can also track parameter correlations as data is added. Fig. 5 shows how, as training data is annealed in, the well-known negative correlation between the linear and nonlinear stiffnesses becomes apparent.

This same data set was then analysed using the original Data Annealing algorithm, such that the relative performance of the two methods could be assessed. The same prior and segments of data were used. Figs. 6 and 7 show how the posterior mean and standard deviation estimates evolved as more data was analysed.

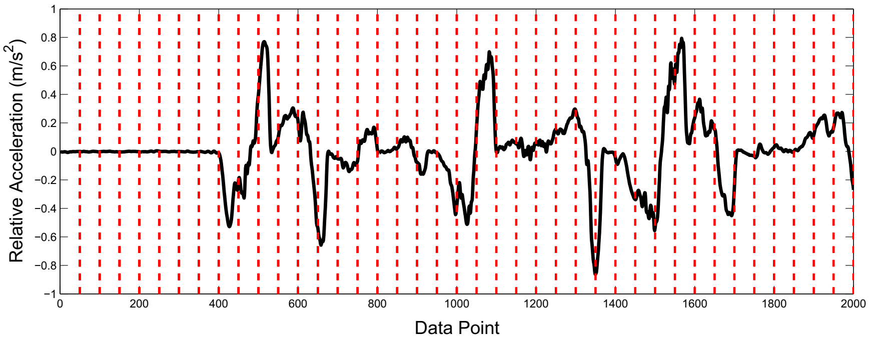

Fig. 10. Training data for the system identification of a rotational energy harvester. Dashed red lines indicate the segments of data which were used in the SDA algorithm. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Table 2
System identification of rotational energy harvester: moments of prior distributions.

Parameter	Prior mean	Prior standard deviation	Units
c	170	50	N s/m
F_c	10	5	N
α	100	100	s/m
σ	0.07	0.03	–

While these results look promising, it was found that they were based on relatively few independent samples of $\boldsymbol{\theta}$. This is because the algorithm was unable to appropriately adapt the size of its proposal density as the geometry of the posterior altered – this resulted in the acceptance ratio dropping to below 10% once all 1000 data points were being analysed (see Fig. 8 for example). The crucial point here is that Data Annealing can be a very efficient algorithm, just so long as it is tuned appropriately. Choosing smaller segments of data could, for example, have allowed the algorithm to realise a higher acceptance ratio. The advantage of SDA is that it is relatively insensitive to this sort of tuning. If very small segments of data are used then, as they will contain less

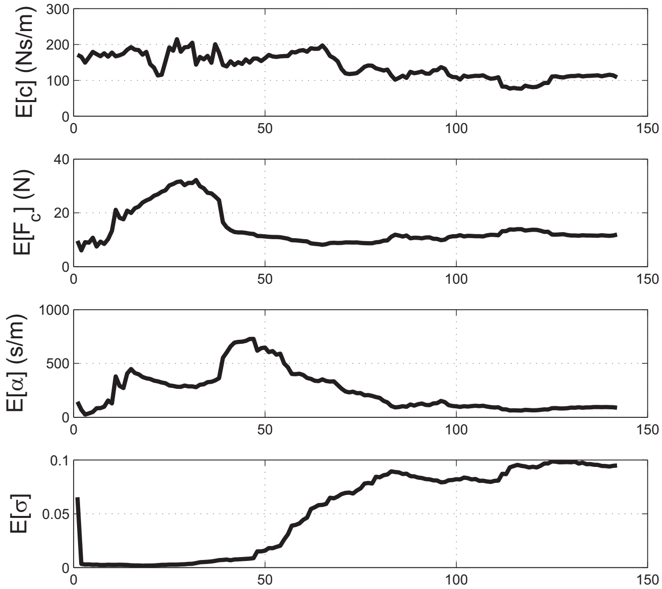

Fig. 12. Parameter estimation of a rotational energy harvester: convergence of posterior mean parameter estimates as SDA algorithm is run.

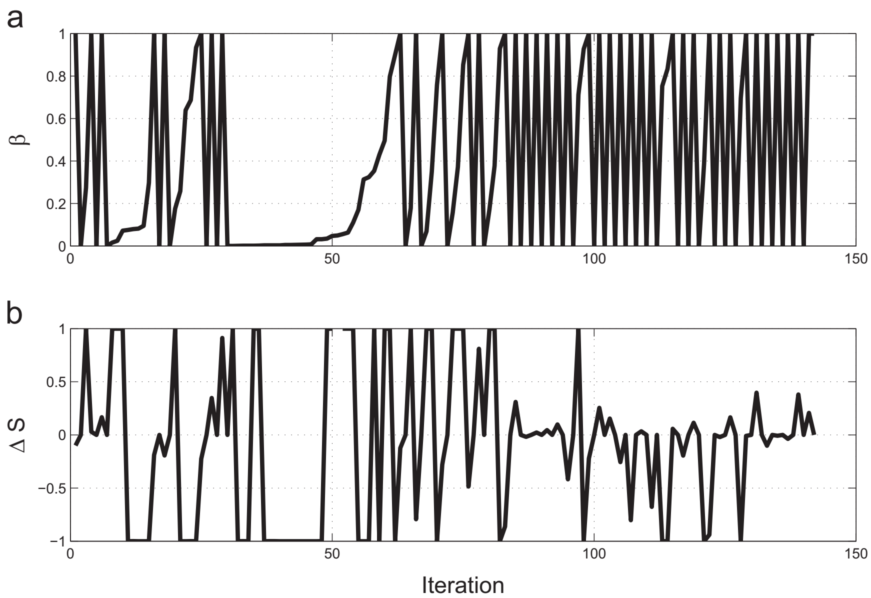

Fig. 11. Parameter estimation of a rotational energy harvester: variation of (a) inverse temperature and (b) the change in Shannon entropy as training data is added to the SDA algorithm.

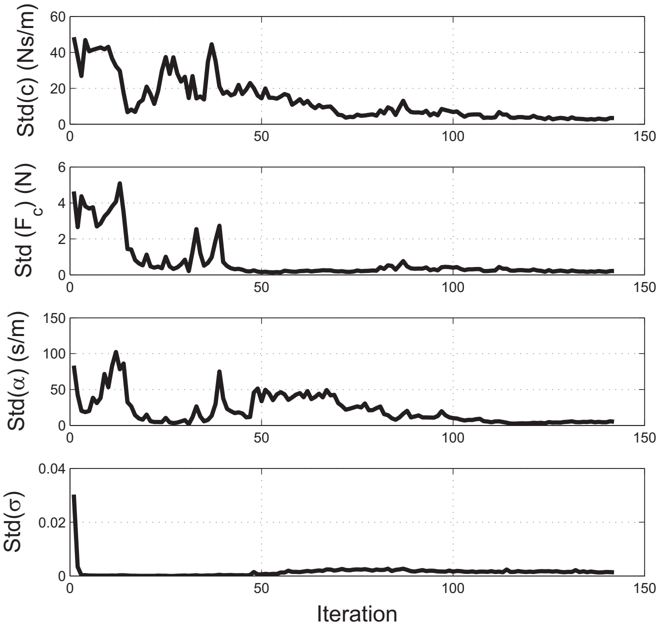

Fig. 13. Parameter estimation of a rotational energy harvester: convergence of posterior standard deviation parameter estimates as SDA algorithm is run.

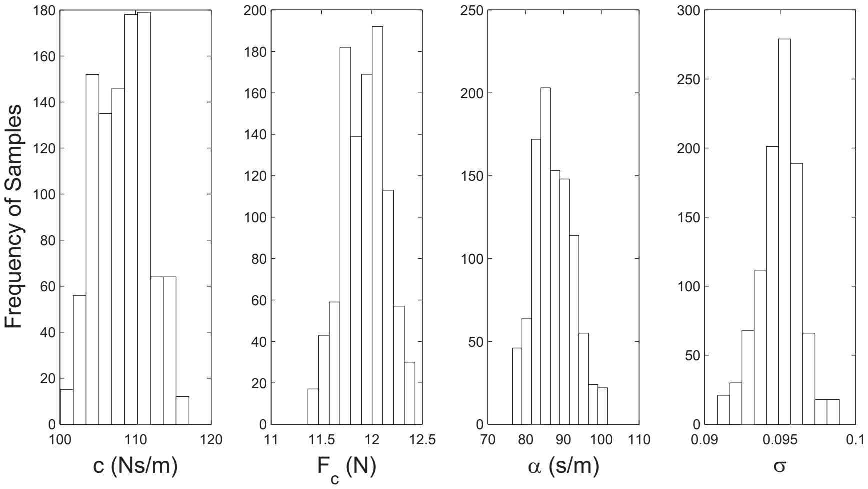

Fig. 14. Parameter estimation of a rotational energy harvester: histograms of SDA results.

information, SDA will move through them quickly. If very large segments of data are used then SDA will ensure that the information contained within will be introduced slowly, and no dramatic changes in the geometry of the posterior will occur.[4] This property therefore gives the user great flexibility when selecting the size of each data segment. Throughout this paper segments were chosen on the basis that they appeared to capture some of the system's dynamic behavior – this led to satisfactory results in both examples.

5. Example 2 – experimental data

In this section SDA is applied to experimentally obtained data. The system in question is a vibrational energy harvesting device which, via a ball-screw mechanism, is able to convert low frequency translational motion into high frequency rotational motion. Originally tested at the University of Southampton's Institute of Sound and Vibration Research, only a very brief description of the device and experimental procedure is given here – more information can be found in the references [19,20]. It should be noted that the data from this experiment can be found in the electronic supplementary material of the paper [22].

A schematic of the energy harvester is shown in Fig. 9. As the device experiences base motion, the mass, m, oscillates relative to the outer frame. This translational motion is then converted into rotational motion via a ball screw. The response of device is strongly affected by friction (as a result of the coupling between the mass and the ball screw). Building on other work on rotational energy harvesters [21], the hyperbolic tangent model was used to model friction effects in the device. Defining $z = x - y$ as the relative displacement between the mass and the base, the proposed

[4] If a single segment is used (which contains all available data), SDA essentially becomes a standard Simulated Annealing algorithm whose annealing schedule will result in constant variations in the Shannon entropy of the target distribution.

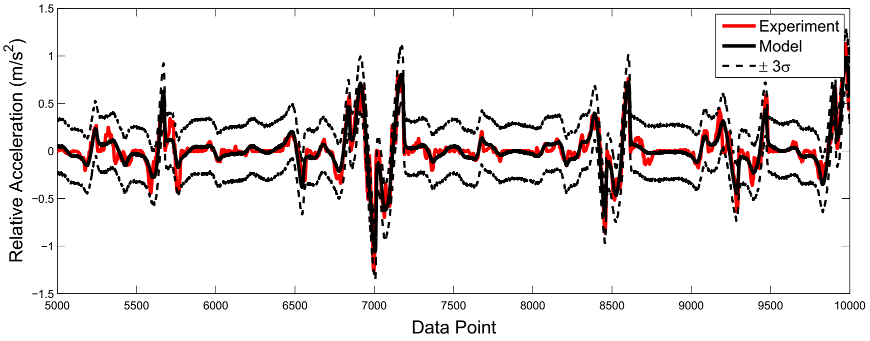

Fig. 15. Propagating parameter uncertainty in rotational energy harvester model.

equation of motion is therefore

$$M\ddot{z} + b_m\dot{z} + kz + F_c \tanh(\alpha\dot{z}) = -m\ddot{y} \tag{33}$$

where

$$M = m + J\left(\frac{2\pi}{l}\right)^2, \quad b_m = \left(\frac{2\pi}{l}\right)^2 c_m, \tag{34}$$

J is the moment of inertia of the system and l is the ball-screw lead. The parameters to be estimated were c, F_c, α and σ where, as before, σ is the likelihood standard deviation.

The 'full' training data consisted of 2000 points of relative acceleration time history ($\ddot{z}$). This was 'fed' into SDA in segments of 50 points at a time (as shown in Fig. 10). The parameters of SDA were the same as in the previous example while, again, Gaussian priors truncated at zero were utilised (see Table 2). The prior was selected based on several static tests which had already been conducted – see [20] for more details.

The variation of β and Shannon entropy is shown in Fig. 11. It is clear that, relative to the previous example, this problem was more challenging (as many more positive values of Shannon entropy were realised). It is also interesting to note that, between the 31st and 63rd iterations of the algorithm, a large amount of time has been spent annealing in a single segment of data. This was actually the 9th subset of data which, as can be seen from from Fig. 10, is where the relatively high amplitude portion of the training data begins.

For the sake of completeness the convergence of the mean and standard deviation parameter estimates is shown in Figs. 12 and 13 while Fig. 14 shows histograms of the resulting MCMC samples. Using the MCMC samples to propagate parameter uncertainties, Monte Carlo simulations were conducted to compare the model response with a new set of test data. Fig. 15 shows that the model is able to replicate the data accurately.

6. Conclusions

Presented here is a novel MCMC algorithm – Smooth Data Annealing (SDA) – which is designed to be used in situations where one is conducting Bayesian system identification of dynamical models using large sets of training data. The algorithm is designed to 'absorb' data in a smooth and continuous manner, ensuring that the resulting change in the Shannon entropy of one's target distribution remains within predefined limits. This allows the algorithm to quickly move through training data which is relatively uninformative, and concentrate on that which has a greater influence on one's parameter estimates.

Acknowledgements

The author would like to thank Dr. Mehdi Hendijanizadeh, Mr. Luigi Simeone and Professor Stephen Elliott from the University of Southampton's Institute of Sound and Vibration Research for providing the data used in the second example.

The author would also like to acknowledge funding from the EPSRC programme Grant 'Engineering Nonlinearity' EP/K003836.

Appendix A. Deriving Eq. (14)

Recalling that

$$\pi = \frac{\exp(-\beta_j \hat{J}_L - J_L - J_P)}{Z} = \frac{\pi^*}{Z} \tag{A.1}$$

then the first property can be derived by

$$\begin{aligned}\frac{d\pi^*}{d\beta_j} &= \frac{d}{d\beta_j}\exp\left(-\beta_j\hat{J}_L - J_L - J_P\right) = -\hat{J}_L \exp\left(-\beta_j\hat{J}_L - J_L - J_P\right)\\ &= -\hat{J}_L \pi^*.\end{aligned} \tag{A.2}$$

This allows the second property to be derived by

$$\begin{aligned}\frac{dZ}{d\beta_j} &= \frac{d}{d\beta_j}\int \pi^* \, d\boldsymbol{\theta} = \int \frac{d\pi^*}{d\beta_j}\, d\boldsymbol{\theta} = -\int \hat{J}_L \pi^* \, d\boldsymbol{\theta} = -Z\int \hat{J}_L \pi \, d\boldsymbol{\theta}\\ &= -Z\mathrm{E}[\hat{J}_L].\end{aligned} \tag{A.3}$$

Finally then, the third property can be derived by

$$\begin{aligned}\frac{d\pi}{d\beta_j} &= \frac{d}{d\beta_j}(\pi^* Z^{-1}) = -\hat{J}_L \pi - \frac{\pi}{Z}\frac{dZ}{d\beta_j} = -\hat{J}_L\pi + \pi\mathrm{E}[\hat{J}_L]\\ &= \left(\mathrm{E}[\hat{J}_L] - \hat{J}_L\right)\pi.\end{aligned} \tag{A.4}$$

References

[1] J.L. Beck, L.S. Katafygiotis, Updating models and their uncertainties. I. Bayesian statistical framework, J. Eng. Mech. 124 (4) (1998) 455–461.

[2] E. Simoen, C. Papadimitriou, G. Lombaert., On prediction error correlation in Bayesian model updating, J. Sound Vib. 332 (18) (2013) 4136–4152.

[3] D.J.C. MacKay., Bayesian interpolation, Neural Comput. 4 (3) (1992) 415–447.

[4] D.J.C. MacKay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, Cambridge, UK, 2003.

[5] M. Muto, J.L. Beck, Bayesian updating and model class selection for hysteretic structural models using stochastic simulation, J. Vib. Control 14 (1–2) (2008) 7–34.

[6] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller., Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (1953) 1087.

[7] S. Duane, A.D. Kennedy, B.J. Pendleton, D. Roweth., Hybrid Monte Carlo, Phys. Lett. B 195 (2) (1987) 216–222.

[8] P.J. Green, Reversible jump Markov Chain Monte Carlo computation and

Bayesian model determination, Biometrika 82 (4) (1995) 711–732.

[9] J. Ching, Y. Chen, Transitional Markov Chain Monte Carlo method for Bayesian model updating, model class selection, and model averaging, J. Eng. Mech. 133 (7) (2007) 816–832.

[10] J. Skilling, Nested sampling for general Bayesian computation, Bayesian Anal. 1 (4) (2006) 833–859.

[11] J.L. Beck, K.M. Zuev, Asymptotically independent Markov sampling: a new Markov Chain Monte Carlo scheme for Bayesian inference, Int. J. Uncertain. Quant. 3 (5) (2013).

[12] P. Angelikopoulos, C. Papadimitriou, P. Koumoutsakos, Bayesian uncertainty quantification and propagation in molecular dynamics simulations: a high performance computing framework, J. Chem. Phys. 137 (14) (2012) 144103.

[13] S. Kirkpatrick, D.G. Jr., M.P. Vecchi, Optimization by simmulated annealing, Science 220(4598) (1983) 671–680.

[14] P.L. Green, Bayesian system identification of a nonlinear dynamical system using a novel variant of simulated annealing, Mech. Syst. Signal Process. 52 (2015) 133–146.

[15] P.L. Green, E.J. Cross, K. Worden, Bayesian system identification of dynamical systems using highly informative training data, Mech. Syst. Signal Process. 56 (2015) 109–122.

[16] C.E. Shannon, A mathematical theory of communication, ACM SIGMOBILE Mob. Comput. Commun. Rev. 5 (1) (2001) 3–55.

[17] N. Chopin, A sequential particle filter method for static models, Biometrika 89 (3) (2002) 539–552.

[18] K. Worden, J.J. Hensman, Parameter estimation and model selection for a class of hysteretic systems using Bayesian inference, Mech. Syst. Signal Process. 32 (2012) 153–169.

[19] L. Simeone, M.G. Tehrani, S. Elliott, M. Hendijanizadeh, Nonlinear damping in an energy harvesting device, in: Proceedings of ISMA 2014 International Conference on Noise and Vibration Engineering, Leuven, Belgium, 2014.

[20] P.L. Green, M. Hendijanizadeh, L. Simeone, S.J. Elliott, Probabilistic modelling of a rotational energy harvester, J. Intell. Mater. Syst. Struct., http://dx.doi.org/10.1177/1045389X15573343, in press.

[21] I.L. Cassidy, J.T. Scruggs, S. Behrens, H.P. Gavin, Design and experimental characterization of an electromagnetic transducer for large-scale vibratory energy harvesting applications, J. Intell. Mater. Syst. Struct. 22 (17) (2011) 2009–2024.

[22] P.L. Green, K. Worden, Bayesian and Markov chain Monte Carlo methods for identifying nonlinear systems in the presence of uncertainty, Phil. Trans. R. Soc. A 373 (2051) (2015) 20140405.