Parameter estimation from big data using a sequential monte carlo sampler

P.L. Green, S.Maskell
Institute for Risk and Uncertainty, University of Liverpool, Liverpool, UK, L69 3GQ
e-mail: **p.l.green@liverpool.ac.uk**

Abstract

It is now well established that, through the use of sequential Monte Carlo methods, it is possible to track the time-varying state of mechanical systems 'online', using a continous stream of measurements. The best known of these algorithms is the particle filter - a numerical algorithm that can be applied to a large variety of nonlinear problems and which, in recent years, has been used to aid the condition monitoring of many mechanical systems. In this paper, a Sequential Monte Carlo method is used to estimate the parameters of a model from a continuous stream of measurements, with the aim of establishing how one's parameter estimates converge as more data is analysed. Crucially, for reasons described in this paper, this is a situation where a particle filter is unsuitable. The issue is instead resolved using a variant of a Sequential Monte Carlo sampler. It is shown how the algorithm can be used to identify the parameters of a model from large data sets and, within the context of structural dynamics, it is compared with the performance of a similar, Markov chain Monte Carlo method.

1 Introduction

Consider a scenario where one wishes to infer probabilistic estimates of a model's parameters, $\boldsymbol{\theta}$, from a set of measurement data, $\boldsymbol{z}_{1:n} = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_n\}$. In this context $\boldsymbol{z}_t$ represents a vector of measurements taken at time t and so $\boldsymbol{z}_{1:n} = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_n\}$ represents the set of all measurements taken up to time n. These could, for example, represent time histories of acceleration measurements taken from a dynamic structure. Adopting a Bayesian framework, one can realise probabilistic estimates of $\boldsymbol{\theta}$, conditional on $\boldsymbol{z}_{1:n}$, from the posterior parameter distribution:

$$p(\boldsymbol{\theta}\,|\boldsymbol{z}_{1:n}) \propto p(\boldsymbol{z}_{1:n}|\,\boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{1}$$

At this stage one will often generate samples from $p(\boldsymbol{\theta}\,|\,\boldsymbol{z}_{1:n})$ using, for example, Markov chain Monte Carlo (MCMC) methods. When n is large however, this can be very expensive, and so a subset of this training data must be used. How this subset should be selected is often unclear, and it can be difficult to establish when a sufficient amount of information has been extracted from the 'full' dataset. An information-theoretic solution to this issue was proposed in [1] which, while useful in some scenarios, does rely on one approaching the problem with a reasonable estimate of the most probable vector of model parameters. MCMC methods designed to address this scenario were proposed in [2, 3]. These allow one to track how one's estimates of $\boldsymbol{\theta}$ converge as an increasingly large set of data is analysed. Unfortunately, the computational cost of applying these methods in situations where large amounts of data are required can still be prohibitive. This is also an issue which cannot be tackled using a particle filter, as this would lead to the influence on one's initial measurements decreasing as more data is analysed (see [4] for more details).

The current paper proposes a method which, the authors believe, can tackle this problem at reduced computational cost and in a way which is very well suited to parallel processing. It therefore tackles 'Big Data' in the sense that it aims to facilitate the analysis of data sets which are too large for current methodologies. The proposed algorithm is essentially a variant of a Sequential Monte Carlo (SMC) sampler (which, somewhat confusingly, is part of the family of SMC *methods*).

2 Prerequisites

2.1 Importance sampling

Importance sampling forms the foundation of the method proposed here and, as such, it is given a brief introduction in this section. For more information, nice introductions to importance sampling can be found in the books [5, 6] as well as numerous tutorials.

Here, $\pi(\boldsymbol{\theta})$ is used to represent the target distribution which, in the context of this paper, is the posterior parameter distribution in equation (1). Consider the situation where one wishes to estimate the expected value of some function, $f(\boldsymbol{\theta})$, (where $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$). With importance sampling, one first generates the set of samples, $\{\boldsymbol{\theta}^1, ..., \boldsymbol{\theta}^N\}$ from a 'target distribution', $q(\boldsymbol{\theta})$ (which is relatively easy to sample from). One then calculates a set of 'importance weights' according to

$$w^i = \frac{\pi(\boldsymbol{\theta}^i)}{q(\boldsymbol{\theta}^i)}, \qquad i = 1, ..., N. \tag{2}$$

It can then be shown that

$$\mathrm{E}[f(\boldsymbol{\theta})] \approx \sum_{i=1}^{N} f(\boldsymbol{\theta}^i)\tilde{w}^i \tag{3}$$

where

$$\tilde{w}^i = \frac{w^i}{\sum_j w^j}, \qquad i = 1, ..., N \tag{4}$$

are referred to as 'normalised importance weights'. If one then generates a new set of samples, $\{\bar{\boldsymbol{\theta}}^1, ..., \bar{\boldsymbol{\theta}}^N\}$, by sampling with replacement from the original set while ensuring that

$$\Pr(\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}^i) = \tilde{w}^i \tag{5}$$

then it can also be shown that the new set of samples will be an approximate discrete representation of $\pi(\boldsymbol{\theta})$. The weights of the new samples are, as a result, approximately equal. This method, known as 'resampling', helps to remove samples with low weights and produce replicas of those with large weights. Resampling is often performed when one's estimates are dominated by relatively few samples (in other words, when the majority of samples have low weights). This is often referred to as 'degeneracy' in particle filter literature. Based on the work in [7], resampling occurs if the 'effective sample size', N_f, drops below a threshold ($N/2$ in the current paper), where the effective sample size is defined as

$$N_{eff} = \frac{1}{\sum_i (\tilde{w}^i)^2}. \tag{6}$$

It is important to note that the resampling procedure can be readily parallelised (see [8] for more details) and that this suitability for modern, fine grained computer architectures is a property of the algorithm that is proposed in the current paper.

2.2 Previous work

In [9] it was suggested that importance sampling could help to address the issues raised at the beginning of the current paper. In the situation where one has obtained measurements up to time t, the target distribution is $p(\boldsymbol{\theta} \,|\, \boldsymbol{z}_{1:t})$. Using the prior, $p(\boldsymbol{\theta})$, as a proposal density, one's importance weights at time t can then be calculated according to

$$w_t = \frac{p(\boldsymbol{z}_{1:t}|\,\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} = p(\boldsymbol{z}_{1:t}|\,\boldsymbol{\theta}). \tag{7}$$

Assuming that the probability of witnessing separate measurements is independent, such that $p(\boldsymbol{z}_{1:t} \,|\, \boldsymbol{\theta}) = \prod_{i=1}^{t} p(\boldsymbol{z}_i \,|\, \boldsymbol{\theta})$, allows these importance weights to be calculated sequentially by

$$w_t = p(\boldsymbol{z}_t|\,\boldsymbol{\theta})w_{t-1} \tag{8}$$

which prevents one from having to analyse the entire dataset every time a new measurement arrives. Unfortunately however, as more data arrives, the posterior tends to become very concentrated relative to the prior and many of the samples generated will be associated with very low importance weights. While resampling can help to remove these samples, it doesn't help further exploration of the parameter space in regions of high probability density. In [9] it was suggested that a 'move step' could be inserted to address this (possibly using MCMC). This step is, however, rather expensive to implement. In the current work it is suggested that this move step can be conducted efficiently using a SMC sampler, which is introduced in the next section.

3 Sequential Monte Carlo samplers

A brief introduction to SMC samplers is given here. A more detailed description is given in [10] while a more applied introduction can be found in [11].

Say $\boldsymbol{\theta}_k$ is the state of a system at iteration k and that $\pi_k(\boldsymbol{\theta}_k)$ is defined as the kth target distribution - the current distribution of interest. (The observant reader may notice a change in notation here, where k's have been employed instead of t's - the reasons for this are explained later in the text.) $\pi(\boldsymbol{\theta}_{1:k})$ is used to represent the joint distribution over all previous states. With an SMC sampler, rather than targeting $\pi_k(\boldsymbol{\theta}_k)$ directly, one instead targets

$$\pi_k(\boldsymbol{\theta}_k) \prod_{k'=2}^{k} L(\boldsymbol{\theta}_{k'-1}|\boldsymbol{\theta}_{k'}) \tag{9}$$

where $L(\boldsymbol{\theta}_{k'-1}|\boldsymbol{\theta}_{k'})$ (known as the 'L-kernel') is defined such that

$$\int \left\{ \pi_k(\boldsymbol{\theta}_k) \prod_{k'=2}^{k} L(\boldsymbol{\theta}_{k'-1}|\boldsymbol{\theta}_{k'}) \right\} d\boldsymbol{\theta}_{1:k-1} = \pi_k(\boldsymbol{\theta}_k). \tag{10}$$

Generating samples from the proposal distribution $q(\boldsymbol{\theta}_{1:k}) = q(\boldsymbol{\theta}_k \,|\, \boldsymbol{\theta}_{k-1})q(\boldsymbol{\theta}_{1:k-1})$, importance weights are assigned according to:

$$w_k^i = \frac{\pi_k(\boldsymbol{\theta}_k^i) \prod_{k'=2}^{k} L(\boldsymbol{\theta}_{k'-1}|\boldsymbol{\theta}_{k'})}{q(\boldsymbol{\theta}_k \,|\, \boldsymbol{\theta}_{k-1})q(\boldsymbol{\theta}_{1:k-1})}. \tag{11}$$

From this one can then go on to show that

$$w_k^i = w_{k-1}^i \frac{\pi_k(\boldsymbol{\theta}_k^i)}{\pi_{k-1}(\boldsymbol{\theta}_{k-1}^i)} \frac{L(\boldsymbol{\theta}_{k-1}^i|\boldsymbol{\theta}_k^i)}{q(\boldsymbol{\theta}_k^i|\boldsymbol{\theta}_{k-1}^i)} \tag{12}$$

such that, in other words, the importance weights can now be updated in a sequential manner as k increases. At first sight one may consider exchanging the indices k and t such that the target distribution is

$$\pi_t(\boldsymbol{\theta}_t) = p(\boldsymbol{\theta}\,|\boldsymbol{z}_{1:t}) \tag{13}$$

and using the SMC sampler to tackle the problem outlined in Section 1. This would, however, still require analysis of the full dataset every time new measurements arrive. In the current paper it is suggested that the SMC sampler can actually be used to facilitate the 'move step' of the method proposed in [9]. As will be shown, the advantages of this approach are that

- The move step will only be conducted if the effective sample size drops below a pre-defined threshold.
- The move step will be suitable for parallelisation.

4 Proposed methodology

Say one has a model of the form

$$\boldsymbol{y}_t = f(\boldsymbol{x}_t, \boldsymbol{\theta}) \tag{14}$$

where $\boldsymbol{x}_t$ represents the input to the model at time t. It is assumed that measurements are made according to

$$\boldsymbol{z}_t = h(\boldsymbol{y}_t) + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_\epsilon) \tag{15}$$

where the covariance matrix $\boldsymbol{\Sigma}_\epsilon$ is known and h can be a nonlinear function. The proposed methodology for sequential parameter estimation is outlined in algorithm 1.

Algorithm 1 Proposed algorithm.

Set $k = 1$, $t = 1$
Sample $\{\boldsymbol{\theta}_{t,k}^1, ..., \boldsymbol{\theta}_{t,k}^N\}$ from the prior, $p(\boldsymbol{\theta})$
Find initial weights: $w_{t,k}^i = p(\boldsymbol{z}_1 \,|\, \boldsymbol{\theta}_{t,k}^i), \qquad i = 1, ..., N$
while **do**
  Normalise weights: $\tilde{w}^i = \frac{w_{t,k}^i}{\sum_j w_{t,k}^j}$
  Estimate quantities of interest.
  Find effective sample size: $N_{eff} = \frac{1}{\sum_i (\tilde{w}^i)^2}$
  while $N_{eff} < N/2$ **do**
    Resample to get $\{\boldsymbol{\theta}_{t,k}^1, ..., \boldsymbol{\theta}_{t,k}^N\}$
    Reset weights: $w_{t,k}^i = 1, \qquad i = 1, ..., N$
    Set $k = k + 1$
    Generate new samples $\{\boldsymbol{\theta}_{t,k}^1, ..., \boldsymbol{\theta}_{t,k}^N\}$ from $q(\boldsymbol{\theta}_{t,k} \,|\, \boldsymbol{\theta}_{t,k-1})$
    Find new weights: $w_{t,k}^i = \frac{p(\boldsymbol{z}_{1:t} \,|\, \boldsymbol{\theta}_{t,k}^i)}{p(\boldsymbol{z}_{1:t} \,|\, \boldsymbol{\theta}_{t,k-1}^i)} \frac{L(\boldsymbol{\theta}_{t,k-1}^i \,|\, \boldsymbol{\theta}_{t,k}^i)}{q(\boldsymbol{\theta}_{t,k}^i \,|\, \boldsymbol{\theta}_{t,k-1}^i)}, \qquad i = 1, ..., N$
    Normalise weights: $\tilde{w}^i = \frac{w_{t,k}^i}{\sum_j w_{t,k}^j}$
    Find effective sample size: $N_{eff} = \frac{1}{\sum_i (\tilde{w}^i)^2}$
  end while
  $t = t + 1$
  Find new weights: $w_{t,k}^i = w_{t-1,k}^i p(\boldsymbol{z}_t \,|\, \boldsymbol{\theta}_{t,k})$
end while

Key points to note are that:

- If degeneracy doesn't occur (such that the effective sample size remains sufficiently large) then the SMC sampler will never be employed, and the algorithm will proceed in a similar manner to that proposed in [9] (except without 'move steps').
- The SMC sampler is only employed if degeneracy occurs. The SMC sampler helps to provide a further exploration of the parameter space in a way which is suitable for parallel processing.
- The SMC sampler will run until the effective sample size has increased above the predefined threshold.
- k is used to index the number of times the SMC sampler is run, while t indexes time.

One of the advantages of this approach is that, if new data has relatively little effect on the geometry of the posterior, then the samples simply need to be reweighted. It is only when 'information rich' data arrives, and a large change in the posterior occurs, that the parameter samples need to be moved (and the entire dataset needs to be reanalysed). The method therefore embodies the idea of 'highly informative training data' outlined in [1].

With regard to the L-kernel, it should be noted that a *symmetric proposal distribution is utilised throughout the following examples.* The L-kernel is chosen to be

$$L(\boldsymbol{\theta}_{t,k-1}^i | \boldsymbol{\theta}_{t,k}^i) = q(\boldsymbol{\theta}_{t,k-1}^i | \boldsymbol{\theta}_{t,k}^i) \tag{16}$$

which then allows the weights of the samples generated using the SMC sampler to be calculated from

$$w_{t,k}^i = \frac{p(\boldsymbol{z}_{1:t} \,|\, \boldsymbol{\theta}_{t,k}^i)}{p(\boldsymbol{z}_{1:t} \,|\, \boldsymbol{\theta}_{t,k-1}^i)} \frac{L(\boldsymbol{\theta}_{t,k-1}^i \,|\, \boldsymbol{\theta}_{t,k}^i)}{q(\boldsymbol{\theta}_{t,k}^i \,|\, \boldsymbol{\theta}_{t,k-1}^i)} = \frac{p(\boldsymbol{z}_{1:t} \,|\, \boldsymbol{\theta}_{t,k}^i)}{p(\boldsymbol{z}_{1:t} \,|\, \boldsymbol{\theta}_{t,k-1}^i)}. \tag{17}$$

5 Results

5.1 Example 1 - linear system

In this first example, a linear model is considered, where

$$y_t = \theta x_t \tag{18}$$

and measurements are made according to

$$z_t = y_t + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \beta_\epsilon^{-1}) \tag{19}$$

where β_ϵ is the noise precision. This is a convenient problem as, by choosing a Gaussian prior, $p(\theta) = \mathcal{N}(\theta; \mu_0, \beta_0^{-1})$, closed form expressions for the posterior distribution can be obtained. This therefore allows one to check that the proposed method can follow the true solution. 100 'measurements' were generated (sampling x from $U[0, 1]$) and the data was analysed sequentially. The prior moments, noise precision and true value of θ were

$$\mu_0 = 0, \qquad \beta_0 = 1, \qquad \beta_\epsilon = 100, \qquad \theta = 1 \tag{20}$$

while $N = 100$ samples were employed. Proposals were drawn according to

$$q(\theta_{t,k}^i | \theta_{t,k-1}^i) = \mathcal{N}(\theta_{t,k}^i; \theta_{t,k-1}^i, 1). \tag{21}$$

Figure 1 shows that the algorithm performed well, and was able to closely match the true solution. It should be noted that resampling - the reanalysis of the full dataset - was only needed when particularly 'informative' data points were analysed (in other words, when large changes in the estimated mean and variance occurred). The remainer of the analysis could be performed very cheaply, without the SMC sampling step.

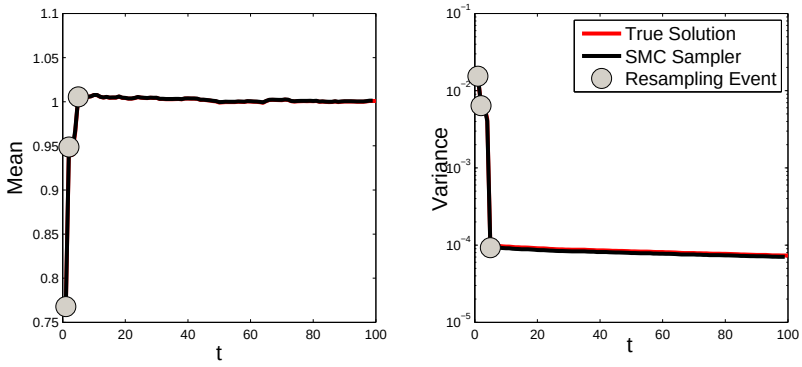

Figure 1: Applying the proposed methodology to the identification of a linear system.

5.2 Example 2 - comparison with MCMC

In this section, the previous example is reanalysed using a MCMC based method. Specifically, a variant of the well-known Transitional MCMC (TMCMC) [12] algorithm is proposed and applied. Here, when the effective sample size drops below the predefined threshold, new samples are generated in a manner similar to that employed in the TMCMC algorithm. Specifically, Markov chains (which are ergodic and whose stationary distributions are equal to the target), are 'grown' from the existing samples. The probability that each chain will grow is equal to the normalised importance weights of the corresponding existing sample. This

therefore encourages new samples to be generated in the region where existing samples are highly weighted. The problem described in the previous section was analysed 1000 times using the proposed SMC methodology as well as this new, TMCMC based methodology.

The statistics of the ensemble of results are shown in Figure 2. It is clear that both methodologies are very closely matched. The advantage of the SMC approach, however, is that it will always be suitable for parallel processing (while the parallelisation of TMCMC depends on the distribution of importance weights). In an extreme case, for example, where one sample has a normalised weight equal to 1 and all others are equal to zero, the TMCMC approach would result in the generation of a single Markov chain which would be impossible to parralelise.

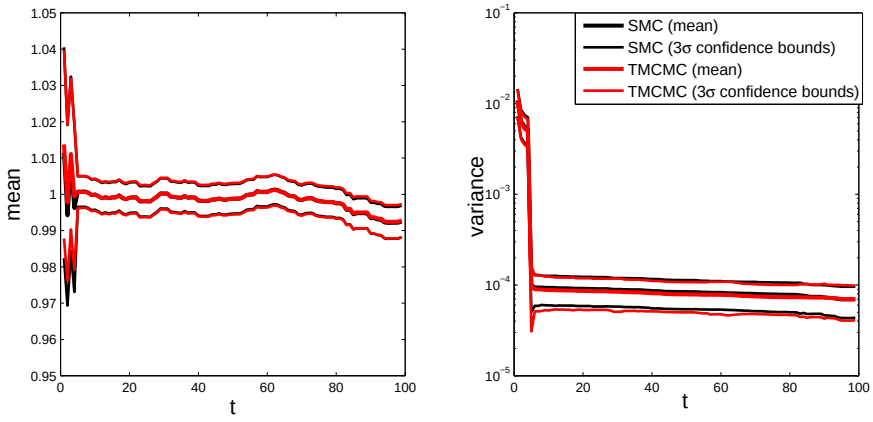

Figure 2: Identification of a linear system using the proposed SMC approach and the proposed MCMC approach. Statistics are taken from an ensemble of 1000 simulations.

5.3 Example 3 - simulated two-storey structure

As the final example in this paper, 'measurement' data is generated from a simulation of a linear two-storey structure. The mass of each floor was set equal to 10000 kg. The vector of parameters to be estimated were $\boldsymbol{\theta} = (k_1, k_2, c_1, c_2)^T$ where the k's and c's represent interstorey stiffness and damping terms respectively. Their true values, and details of the prior distributions that were utilised, are given in Table 1.

Parameter	True value	Prior
k_1	30×10^6	$\mathcal{N}(k_1; 25 \times 10^6, 2, 5 \times 10^{12})$
k_2	20×10^6	$\mathcal{N}(k_2; 25 \times 10^6, 2, 5 \times 10^{12})$
c_1	30×10^3	$\Gamma(c_1; 1, 5 \times 10^4)$
c_2	20×10^3	$\Gamma(c_2; 1, 5 \times 10^4)$

Table 1: True parameter values and corresponding prior (SI units).

To generate training data, the structure was subjected to a band-limited white noise base acceleration, with standard deviation equal to 0.01 m/s^2. The displacement time histories of the two floors were then corrupted with Gaussian noise, such that the signal to noise ratio was approximately 10. Figure 3 shows how the proposed algorithm successfully converges to the true parameter values as more data is analysed. It should be noted that, again, as more data is analysed resampling is needed less frequently. This is because, after an initial phase, new measurement data is contributing relatively little additional information.

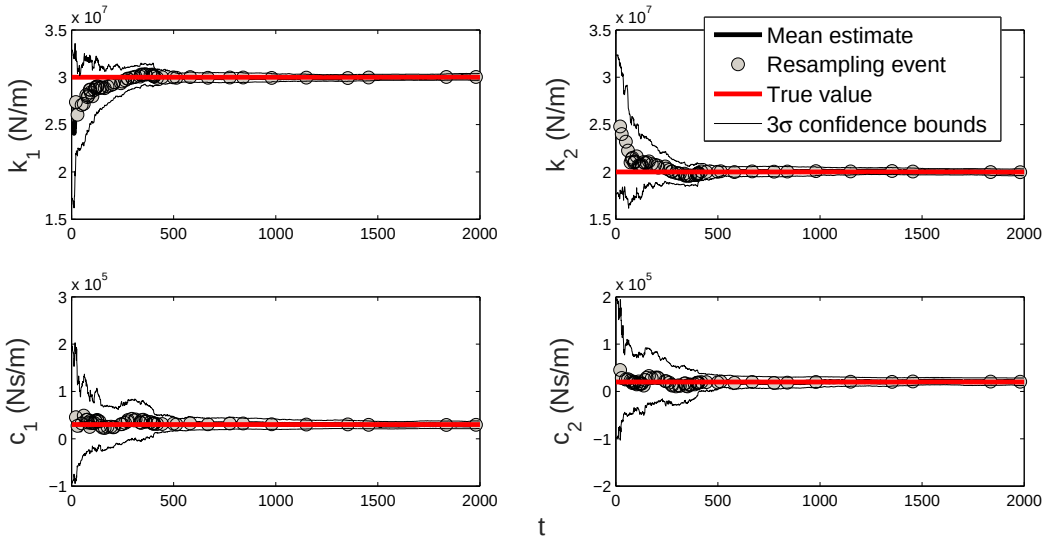

Figure 3: Parameter estimation of two-storey simulated structure using the proposed SMC methodology.

6 Conclusions

In this paper the authors investigate the parameter identification of systems from data sets which are continuously growing. The aim is to provide an efficient methodology which can be used to track how one's parameter estimates converge as more data is analysed, thus allowing one to decide when a sufficient quantity of data has been analysed. An algorithm is proposed which utilises a Sequential Monte Carlo sampler. The efficiency of the method stems from its ability to quickly analyse data which adds little additional information, thus allowing it to focus on data which is 'information rich'. It is demonstrated that the proposed method is able to achieve a similar level of repeatability when compared to an alternative, Markov chain Monte Carlo method. The proposed Sequential Monte Carlo method is, however, better suited to parallel implementation on modern computer architectures.

References

[1] P.L. Green, E.J. Cross, K. Worden. *Bayesian system identification of dynamical systems using highly informative training data*. Mechanical Systems and Signal Processing, 56:109–122, 2015.

[2] P.L. Green. *Bayesian system identification of a nonlinear dynamical system using a novel variant of simulated annealing*. Mechanical Systems and Signal Processing, 52:133–146, 2015.

[3] P.L. Green. *Bayesian system identification of dynamical systems using large sets of training data: A MCMC solution*. Probabilistic Engineering Mechanics, 42:54–63, 2015.

[4] D. Crisan, A. Doucet. *A survey of convergence results on particle filtering methods for practitioners*. Signal Processing, IEEE Transactions on, 50(3):736–746, 2002.

[5] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[6] D.J.C. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[7] A. Kong. *A note on importance sampling using standardized weights*. University of Chicago, Dept. of Statistics, Tech. Rep, 348, 1992.

[8] S. Maskell, B. Alun-Jones, M. Macleod. *A single instruction multiple data particle filter*. In Nonlinear Statistical Signal Processing Workshop, 2006 IEEE, pages 51–54. IEEE, 2006.

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

[10] P. Del Moral, A. Doucet, A. Jasra. *Sequential monte carlo samplers*. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 68(3):411–436, 2006.

[11] S. Maskell. *An application of Sequential Monte Carlo samplers: An alternative to particle filters for non-linear non-Gaussian sequential inference with zero process noise*. In Data Fusion & Target Tracking Conference (DF&TT 2012): Algorithms & Applications, 9th IET, pages 1–8. IET, 2012.

[12] J. Ching, Y.C. Chen. *Transitional Markov chain Monte Carlo method for Bayesian model updating, model class selection, and model averaging*. Journal of engineering mechanics, 133(7):816–832, 2007.