Particle relaxation method of Monte Carlo filter for structure system identification

Tadanobu Sato [1,*]

Phone +86-25-83794567
Fax +86-25-83790575
Email satotdnbseu@yahoo.co.jp

[1] International Institute of Urban Engineering, Southeast University, 2 Sipailou Street, Nanjing, 210096 Jiangsu, China

Abstract

In this paper, we apply Monte Carlo filter (MCF) to identify dynamic parameters of structural systems and improve the efficiency of its algorithm. The algorithms using MCF so far have not been practical for applying to structural identification of large-scale systems because computation time increases exponentially as the degrees of freedom of the system increases. To overcome this problem, we developed a method with the ability to reduce the number of particles which express possible structural response state vector. In MCF, there are two steps which are the prediction and filtering processes. The idea is very simple. The prediction process remains intact but the filtering process is conducted at each node of structural system in the proposed method. We named this algorithm as relaxation Monte Carlo filter (RMCF) and demonstrated its efficiency to identify large degree of freedom systems. Moreover, to increase searching field and speed up convergence time of structural parameters, we proposed an algorithm combining the genetic algorithm with RMCF and named GARMCF.

Keywords

Monte Carlo filter

Particle filter
Genetic algorithm
System identification
Particle relaxation

1. Introduction

Recently many large-scale structures have been constructed. Those are essential infrastructures to support urban functions of megacities. Because to secure their seismic reliability is indispensable for the safety of the whole society, it is important to detect even a minor damage to these structures as early as possible after an earthquake occurrence. One of the methods to detect damage to structures is the system identification technique to identify dynamic parameters of structural systems using observed responses, in which the Kalman filter [1] has been a well-known technique. This is a recursive algorithm that estimates the first and second moments of the state vector for a linear system under the assumption of Gaussian uncertainty of observation and system noises. To overcome the limitations imposed by the linearity, a series of studies have been carried out by many researchers, including Yun and Shinozuka [2], Hoshiya and Saito [3], Loh and Chung [4], Smyth et al. [5], Sato and Qi [6], Sato and Takei [7] and Takaba and Katayama [8]. One of the most well-known algorithms is the extended Kalman filter (EKF) in which we use a linearization technique for the system and observation equations using the first-order Taylor expansion at the mean value of system parameters. To overcome instability problems caused by linearization of nonlinear problems in the method, the unscented Kalman filter is proposed [9] in which several points distributing near mean value are used to calculate a new mean value and variance to the next step. However, these methods still use the Gaussian assumption to calculate the likelihood of state vector from a given observation. The boot strap method [10] known as a particle filter [11] was proposed to estimate the state variables with nonlinear and non-Gaussian distribution characteristics. The basic concept of this method is that the probability density function of state variables conditioned with observation data is approximated by particles. At the almost same time when the boot strap method developed, Kitagawa [12] proposed independently the Monte Carlo

filter (MCF) with the same concept. In this paper, we use a word of MCF instead of the particle filter. This method has great potential for nonlinear and non-Gaussian system identification and provides versatile filtering approaches to estimate the system parameters. Many system identification algorithms using the MCF have been developed and have achieved good successes in structural identification fields by many researchers, such as Sato and Kaji [13], Yoshida and Sato [14], Ching et al. [15], Chowdhury et al. [16]: identification of structure–vehicle interaction problems [17], even a nonlinear system identification problem [18–20] and an active control of nonlinear structural systems [21], three-dimensional ground water flow problem [22] and a machine fault detection problem [23]. However, these methods have not been practical to apply to structural identification for large degrees of freedom systems because of exponential increase of computation time as the system becomes very large. This is because the classical MCF had the following problem: In the MCF, the probability density function of the state vector is expressed by many realizations, called particles. The number of combinations of state values increases exponentially as the degrees of freedom increases. Therefore, we have to generate exponential order of particles to assign enough variation to particles. This results in extreme increase of computation time. In this paper, to overcome this problem, we developed an efficient structural identification algorithm applicable to large degrees of freedom systems by improving the filtering process of MCF algorithm. Moreover, we improved convergence speed to the true value by combining genetic algorithm (GA) [24] with the proposed method.

2. Brief explanation of classical Monte Carlo filter

The general state space model is described by the state transfer and observation equations as follows:

$$x_t = F(x_{t-1}, w_{t-1}) \tag{1}$$

$$y_t = H(x_t, v_t) \tag{2}$$

in which t is the discrete time step, x_t the state vector, y_t the observation vector, w_t the system noise vector and v_t is the observation noise vector which is assumed to be expressed by

$$v_t = H^{-1}(y_t, x_t) = G(y_t, x_t) \tag{3}$$

The MCF can be applied even if the state space model is nonlinear and non-Gaussian. In the MCF, the probability density function of the state vector is expressed by many realizations, called particles and of which time marching behavior is calculated step by step. The MCF is therefore an algorithm to identify particles which express the conditional probability density function $p(x_t|Y_t)$ instead of identifying the estate vector x_t directly in which $Y_t = \{y_t, y_{t-1}, \ldots, y_1\}$. We called $p(x_t|Y_{t-1})$ as the prediction distribution and $p(x_t|Y_t)$ as the filter distribution, and each probability density function (PDF) is approximated by n particles of realization.

If we have n particles of realization from the prediction distribution $p(x_{t-1}|Y_{t-1})$ at $(t-1)$th time step as follows:

$$x^{(j)}_{t-1|t-1} \sim p(x_{t-1}|Y_{t-1}) \quad j = 1, 2, \ldots, n \tag{4}$$

Cumulative distribution of x_{t-1} can be approximated by

$$P(x_{t-1}|Y_{t-1}) = \frac{1}{n}\sum_{j=1}^{n} U\left(x_{t-1} - x^{(j)}_{t-1|t-1}\right) \tag{5}$$

where U is the step function. The approximated PDF of the state vector at $(t-1)$th time step is given by the derivation of cumulative distribution function with respect to x_{t-1}

$$p(x_{t-1}|Y_{t-1}) = \frac{1}{n}\sum_{j=1}^{n} \delta\left(x_{t-1} - x^{(j)}_{t-1|t-1}\right) \tag{6}$$

where δ is the Dirac delta function.

The particles of *t*th time step before observation updating are obtained by simply substituting $(t-1)$th time step particles into the state transfer equation

$$x_{t|t-1}^{(j)} = F\left(x_{t-1|t-1}^{(j)}, w_{t-1}^{(j)}\right) \tag{7}$$

The approximated PDF of the state vector before updating at *t*th time step is estimated by particles of realization

$$p(x_t|Y_{t-1}) = \frac{1}{n}\sum_{j=1}^{n}\delta\left(x_t - x_{t|t-1}^{(j)}\right) \tag{8}$$

The PDF of the state vector after updating by adding observation data y_t is obtained through Bayesian theorem

$$\begin{aligned} p(x_t|Y_t) &= p(x_t|y_t, Y_{t-1}) = \frac{p(x_t, y_t|Y_{t-1})}{p(y_t|Y_{t-1})} \\ &= \frac{p(y_t|x_t, Y_{t-1})p(x_t|Y_{t-1})}{\int p(y_t|x_t, Y_{t-1})p(x_t|Y_{t-1})dx_t} \end{aligned} \tag{9}$$

Substituting Eq. (8) into above equation, we have

$$p(x_t|Y_t) = \frac{p(y_t|x_t, Y_{t-1})\,\frac{1}{n}\sum_{j=1}^{n}\delta\left(x_t - x_{t|t-1}^{(j)}\right)}{\int p(y_t|x_t, Y_{t-1})\,\frac{1}{n}\sum_{j=1}^{n}\delta\left(x_t - x_{t|t-1}^{(j)}\right)dx_t} \tag{10}$$

Integration appearing in the denominator can be performed and we obtain

$$\begin{aligned} p(x_t|Y_t) &= \sum_{j=1}^{n}\frac{q_t^{(j)}}{\sum_{i=1}^{n}q_t^{(i)}}\delta\left(x_t - x_{t|t-1}^{(j)}\right) \\ &= \sum_{J=1}^{n}\alpha_t^{(j)}\delta\left(x_t - x_{t|t-1}^{(j)}\right) \end{aligned} \tag{11}$$

where

$$q_t^{(i)} = p\left(y_t | x_{t|t-1}^{(i)}\right) \tag{12}$$

$$\alpha_t^{(i)} = \left(\frac{q_t^{(i)}}{\sum_{j=1}^{n} q_t^{(j)}}\right) \tag{13}$$

$q_t^{(i)}$ is the likelihood of $x_{t|t-1}^{(i)}$ after the observation data y_t is given.

There two ways to use Eq. (11) for updating the probability density of particles. The first is keeping the initial particles and tracking the particle loci through the state transfer equation, only the weight of each particle is changed based on its likelihood value calculated by using observation data at each time step. This method is called as the sequential important sampling (STS). If we apply this method to identify PDF of state vector, the confidence in the estimated PDF deteriorates because only small number of particles has meaningful weight values and the remaining particle's weight converges to zero. This phenomenon is called as weight degeneracy or particle impoverishment. To overcome this deficiency, a procedure is proposed to resample particles at each time step based on the approximated particles PDF [25]. In the following analysis, we use the resampling method.

3. An algorithm combining genetic algorithm and Monte Carlo filter

A study to investigate the similarity between GA and Monte Caro filter was firstly done by Higuchi [26]. Both MCF and GA are algorithms to reconstruct a set of realization expressing the state values from a random set of initial values. In the MCF, the particles representing the state vector are always generated obeying the conditional probability density function that is influenced by the past information of the structural response. The MCF has, therefore, a function to decrease the influence of past observation noise. On the contrary, it cannot perform

high tracking ability for an abrupt change of dynamic characteristics of structural systems.

We have developed an algorithm combining the mutation procedure of genetic algorithm with the MCF to speed up convergence for identifying non-stationary structural parameters called GAMCF. The proposed method is defined by the following steps:

1. Generate n initial particles $x_{0|0}^{(j)}\ (j = 1, 2, \ldots, n)$ using the given initial probability density function $p_0\left(x_{0|0}\right)$.

2. Set $t = 1$ and repeat the following steps until the end of time steps.

(a) Generate n particles of system noise $w_{t-1}^{(j)}\ (j = 1, 2, \ldots, n)$ using the system noise probability distribution function $p(w_{t-1})$.

(b) Calculate the prediction particles by

$$x_{t|t-1}^{(j)} = F\left(x_{t-1|t-1}^{(j)}, w_{t-1}^{(j)}\right) \qquad 14$$

(c) Select k particles with large likelihood from the mother set of $x_{t-1|t-1}^{(j)}$ and assemble a set of filter particles $\Gamma = \{x_{t-1|t-1}^{(1)}, \ldots, x_{t-1|t-1}^{(k)}\}$. Remaining $n - k$ particles are intact and just go through the MCF process. A set of prediction particles obtained by substituting Γ into Eq. (14) is defined as $B = \{x_{t|t-1}^{(1)}, \ldots, x_{t|t-1}^{(k)}\}$.

(d) Generate a new set of filter particles by applying GA operation to Γ and calculate a new set of prediction particles $\acute{B} = \{\acute{x}_{t|t-1}^{(1)}, \ldots, \acute{x}_{t|t-1}^{(k)}\}$ using Eq. (14)

(e) Calculate the likelihood $\alpha_t^{(j)}$ and $\acute{\alpha}_t^{(j)}$ of $x_{t|t-1}^{(j)}$ and $\acute{x}_{t|t-1}^{(j)}$ by

15

$$\begin{aligned} \alpha_t^{(j)} &= r\left(G\left(y_t, x_{t|t-1}^{(j)}\right)\right)\left|\frac{\partial G}{\partial y_t}\right| \\ \acute{\alpha}_t^{(j)} &= r\left(G\left(y_t, \acute{x}_{t|t-1}^{(j)}\right)\right)\left|\frac{\partial G}{\partial y_t}\right| \end{aligned} \tag{16}$$

(f) Compare $\alpha_t^{(j)}$ and $\acute{\alpha}_t^{(j)}$. When $\acute{\alpha}_t^{(j)}$ is larger than $\alpha_t^{(j)}$, then replace $x_{t|t-1}^{(j)}$ and $\alpha_t^{(j)}$ with $\acute{x}_{t|t-1}^{(j)}$ and $\acute{\alpha}_t^{(j)}$, respectively.

(g) Generate $x_{t|t}^{(j)}$ by resampling $x_{t|t-1}^{(j)}$ based on $\alpha_t^{(j)}$ as follows:

$$x_{t|t}^{(j)} = x_{t|t-1}^{(j)} \quad \text{with the probability} \quad \frac{\alpha_t^{(j)}}{\sum_{l=1}^{m} \alpha_t^{(l)}} \tag{17}$$

(h) Return to (a) by setting $t = t + 1$ until the end of time steps.

4. System and observation equations

We formulate the state transfer and observation equations for a M degrees of freedom shear frame structure. The equation of motion for this system at the time t is given by

$$m_i\left(\ddot{z}_i + \ddot{X}_g\right) + c_i\dot{u}_i + k_i u_i - (1 - \delta_{iM})(c_{i+1}\dot{u}_{i+1} + k_{i+1}u_{i+1}) = 0 \tag{18}$$

in which m_i is the mass of node i, c_i and k_i are, respectively, the damping coefficient and stiffness of structural element i. Node and structural element number are assigned from the bottom. $\ddot{z}_i$, $\dot{z}_i$ and z_i are the relative acceleration, velocity and displacement of node i to the ground. u_i is the relative displacement between neighboring nodes defined by $u_i = z_i - z_{i-1}$. $\ddot{X}_g$ is the ground acceleration. For the case of $i = 1$ we assign $z_{i-1} = 0$ and $\dot{z}_{i-1} = 0$. δ_{iN} is the Kronecker's delta. We define the state and observation vectors as follows in which i is the node and structural element index:

$$x_t = \{\ldots, z_i, \dot{z}_i, c_i, k_i, \ldots\}_t^T \tag{19}$$

$$y_t = \{\ldots,\ z_i, \dot{z}_i, \ldots\}_t^T \quad i = 1, 2, \ldots, M \tag{20}$$

There is no time index t for structural responses and the ground acceleration as well as structural parameters directly except being necessary but each value is evaluated at the time t. The state transfer equation is expressed by the following equation:

$$x_t = x_{t-1} + \int_{t-1}^{t} g(x_{t-1})dt + w_t \tag{21}$$

in which g is expressed as follows:

$$g = \begin{Bmatrix} \vdots \\ \dot{z}_i \\ -\dfrac{c_1}{m_i}\dot{u}_i - \dfrac{k_i}{m_i}u_i + \dfrac{1-\delta_{iM}}{m_i}\left(c_{i+1}\dot{u}_{i+1} + k_{i+1}u_{i+1}\right) - \ddot{X}_g \\ 1 \\ 1 \\ \vdots \end{Bmatrix}_t \tag{22}$$

When the relative displacement and velocity are observed, the observation equation is

$$y_t = Hx_t + v_t \tag{23}$$

in which H is a $(2 \times 4M)$ matrix given by

$$H = \begin{bmatrix} D & & & \\ & D & & \\ & & \ddots & \\ & & & D \end{bmatrix}, D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{24}$$

We can apply the MCF to structural identification using Eqs. (21) and (23) as the state space model.

5. Numerical example 1

To demonstrate the efficiency of the proposed GAMCF to identify the non-stationary structural parameters, we assume a shear type five degrees of freedom system. The weight of each node, damping ratio and stiffness of each layer are 12.056 kN, 0.02 and 240.09 kN/m, respectively. We simulate a set of observed structural responses of which stiffness of the first layer decreases 20 % at 10 s (1,000 time steps) after the earthquake motion input to the structure. The input motion is the El Centro (NS, 1940) acceleration time history whose peak amplitude is adjusted to be 25 gal. As observation noise we added a white noise with 3 % root mean square level of structure response to the simulated structural response. All node responses are used for the identification. The total number of particles is 5,000. In GAMCF, the number of selected particles used for mutation is 100. For mutation purpose, we added a normally distributed random number with the characteristic of $N(0, 0.01)$ to each selected particle. Figure 1 shows comparison of the time histories of the identified distribution of circular frequency ~~of~~ at the first layer ($\omega_1 = \sqrt{k_1/m_1}$, where k_1 is the stiffness of the first layer, m_1 is the mass of the first node) obtained by the normal MCF and that by GAMCF. The identified result obtained by the GAMCF has higher tracking ability for the non-stationary change of structural parameters than that by the normal MCF. Figure 2 shows the number of particles that are replaced by new particles generated in step (f) of GAMCF algorithm. This means that mutations in GA work well to track abrupt change of structural parameters. The number of renewal particles is large especially when the structural system becomes non-stationary.

Fig. 1

Comparison of the time histories of the identified distribution of circular frequency ~~of~~at the first structural element ($\omega = \sqrt{k_1/m_1}$ in which k_1 is the stiffness of the first structural element and m_1 the mass of the first node)

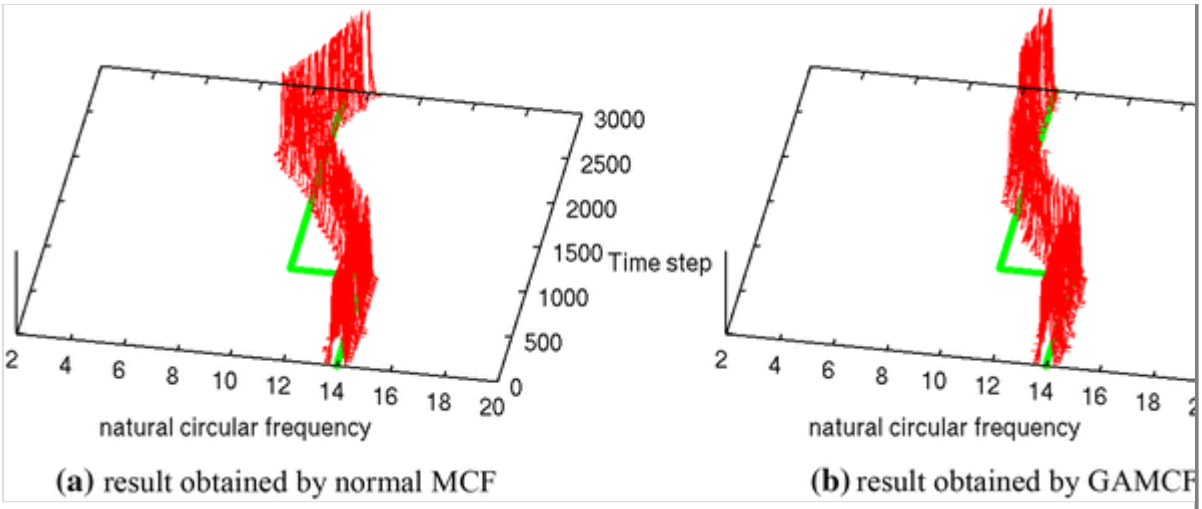
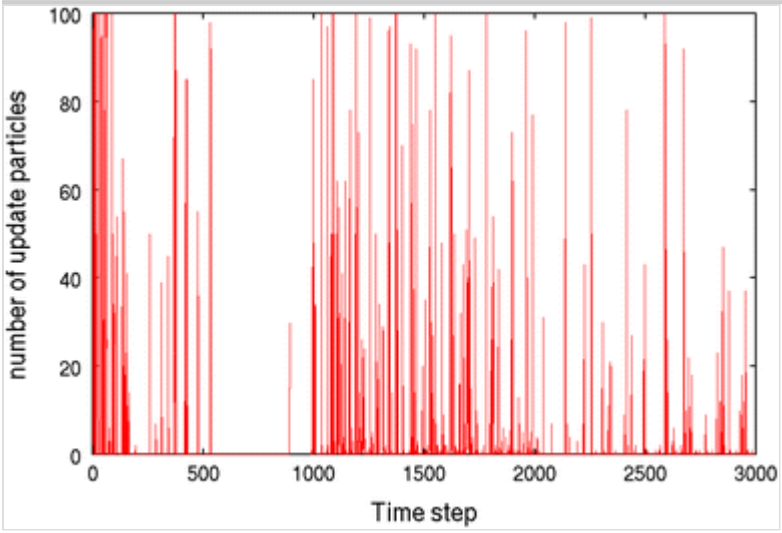


Fig. 2

Number of the renewal particles in GAMCF



6. Particle relaxation algorithm

In the past application of MCF, the state vector is a vector with the dimension of 4 *M* in the case of a *M* degrees of freedom system, so each particle is composed of *M* set of 4 state values, displacement and velocity of each node, as well as stiffness and damping coefficient of each structural element. As the degrees of freedom increases, the number of combinations of 4 *M* state values increases exponentially. Therefore, to assign enough variation to particles for choosing proper candidates of the state vector, we have to generate exponential order of particles. This results in extreme increase of computation time.

Therefore, the classical MCF is not practical for applying to identify a system with large degrees of freedom.

In this paper, we developed a method to overcome this problem. In the proposed algorithm, we modified the calculation of likelihood and resampling of particles in the MCF algorithm. Not to resample particles based on the likelihood of prediction particles of whole system, we conducted resampling at each node or structural element based on the likelihood of particles composed of system variable at a node or layer. We name this algorithm as the relaxation MCF (RMCF) algorithm. The RMCF algorithm consists of the following steps:

1. Generate *n* initial particles $x_{0|0}^{(j)} \ (j = 1, 2, \ldots, n)$ using the given initial probability density function $p_0\left(x_{0|0}\right)$.

2. Set t = 1 and repeat the following steps until the end of time steps.

(a) Generate *n* particles of system noise $w_{t-1}^{(j)} \ (j = 1, 2, \ldots, n)$ using the system noise probability distribution function $p(w_{t-1})$.

(b) Calculate the prediction particle by

$$x_{t|t-1}^{(j)} = F\left(x_{t-1|t-1}^{(j)}, w_{t-1}^{(j)}\right) \tag{25}$$

(c) Subdivide the predicted particles $x_{t|t-1}^{(j)}$ into M components composed of the state variables at each node and structural element $x_{t|t-1;i}^{(j)} (i = 1, 2, \ldots, M)$ in which $x_{t|t-1}^{(j)}$ and $x_{t|t-1;i}^{(j)}$ are defined by

$$x_{t|t-1}^{(j)} = \left\{x_{t|t-1;1}^{(j)}, \ldots, x_{t|t-1;i}^{(j)}, \ldots, x_{t|t-1;N}^{(j)}\right\}^T \tag{26}$$

$$x_{t|t-1;i}^{(j)} = \{z_i, \dot{z}_i, c_i, k_i\}_{t|t-1}^{(j)} \quad (i = 1, 2, \ldots, M) \tag{27}$$

And repeat the following steps for node index from $i = 1$ to $i = M$

(i) Calculate the likelihood of $x_{t|t-1;i}^{(j)}$ by

$$\alpha_{t,i}^{(j)} = r\left(G\left(y_{t,i}, x_{t|t-1;i}^{(j)}\right)\right)\left|\frac{\partial G}{\partial y_{t,i}}\right| \tag{28}$$

in which $y_{t,i} = \{z_i, \dot{z}_i\}, (i = 1, 2, \ldots M)$

(ii) Generate $x_{t|t;i}^{(j)}$ by resampling $x_{t|t-1;i}^{(j)}$ based on $\alpha_{t,i}^{(j)}$ as follows:

$$x_{t|t;i}^{(j)} = x_{t|t-1;i}^{(j)} \quad \text{with the probability} \quad \frac{\alpha_{t,i}^{(j)}}{\sum_{l=1}^{m} \alpha_{t,i}^{(l)}} \tag{29}$$

(iii) Return to (i).

(d) Recompose $x_{t|t}^{(j)}$ as expressed by

$x_{t|t}^{(j)} = \left\{x_{t|t-1;1}^{(j)}, \ldots, x_{t|t-1;i}^{(j)}, \ldots, x_{t|t-1;M}^{(j)}\right\}^T$ for each index j (we call this as the assembling process) in which we consider two cases. Case 1 is just to recompose $x_{t|t}^{(j)}$ by assembling $x_{t|t-1;i}^{(j)} (i = 1, 2, \ldots, M)$ for an arbitrary assigned order of j and Case 2 is to recompose $x_{t|t}^{(j)}$ by assembling $x_{t|t-1;i}^{(j)} (i = 1, 2, \ldots, M)$ for j which is arranged in order of large likelihood of each particle.

(e) Set $t = t + 1$ and return to (a) until the end of time steps

In this method, each prediction particle at the node and structural element i, i.e., $x_{t|t-1;i}^{(j)} (j = 1, 2, \cdots, n)$, is a 4-dimensional vector and only composed of 4 state variables as defined by Eq. (27). This means that the necessary number of particles in the filtering process defined by Eq. (29) is equal to the number of particles which can filter the particles for a single degree of freedom system. If the total number of

particles for identifying a single degree of freedom system is n, the total number of particles applying RMCF to a M degrees of freedom structural system becomes enough with M multiply n. If we use the original MCF for identifying the M degrees of freedom structural system, we need to generate n^M combination of particles. Based on this proposed algorithm, we can reduce dramatically the computation time. However, the ~~deficit~~defect is that a recomposed filtered particle of the structural system $x_{t|t}^{(j)}$ does not satisfy exactly the equation of motion at the time step t and has a small residual error. This ~~deficiency~~defect is not so essential because we can assume that this error is a part of the system noise to obtain prediction particles in the next time step $t + 1$ using Eq. (25).

7. Numerical example 2

Through the following numerical example, we demonstrate the efficiency of the RMCF algorithm. The structural model used is a shear type ten degrees of freedom system. All the nodes have the same mass, damping coefficient and stiffness which are, respectively, 123.02 (kg), 68.606 (N s/m), 24,008.69 (N/m). The input motion is the El Centro (NS1940) acceleration time history whose maximum acceleration is modified to 25 (gal). First, we simulate the structural responses and use these structural responses of all nodes as observation data adding a time history of white noises with 3 % noise of signal ratio to root mean square of the structural responses. The initial values of the unknown parameters c_i and k_i are defined as one half of the true values. The number of particles used is 1,000 for each node and structural element. We compare the effect of two assembling process to recompose the filtered particles of the whole system, $x_{t|t}^{(j)}$, on the numerical convergence speed.

Figure 3 shows the distribution of likelihood of particles at a node and structural element, for example, $x_{t|t;7}^{(j)}$ (particle at 7 node and structural element), at 15 s. In Case 1, there is no relationship between the array number of particle j and the likelihood of the particle, whereas we can see a strong correlation between them if we use the process to rearrange particle array with respect to the order of their likelihood as shown in

Case 2. These distribution characteristics are also seen in the relationship between the particle number and likelihood at all other nodes. Figure 4 shows the distribution of likelihood of $x_{t|t}^{(j)}$ (a recomposed particle expressing filtered state vector) at 15 s. The width of likelihood variation is large and there are few particles with large likelihood in Case 1, whereas there are more particles with larger likelihood in Case 2 than those in Case 1. From this figure, we can see that RMCF can intentionally generate particles with large likelihood by rearranging the array of filtered particle of each node and structural element $x_{t|t;i}^{(j)}$ before assembling to recompose $x_{t|t}^{(j)}$. Figure 5 shows the time history of mean of the identified particles for stiffness in the structural element with odd numbers. Figure 6 shows the time history of probability density function of stiffness in the first structural element. Convergence time of Case 2 to the true value is faster than that of Case 1. Execution time of the developed program depends on a computer used but with CPU speed of 1.2 GHz it needs 5 min 36 s in Case 1 and 5 min 56 s in Case 2. Computation time in Case 2 is a little bit longer than Case 1 because of rearranging the array of $x_{t|t;i}^{(j)}$ before recomposing $x_{t|t}^{(j)}$. But considering improvement of convergence speed, this increase of computation time is not a disadvantage.

Fig. 3

Distribution of likelihood of $f_{n,7}{}^{(j)}$ (~~7 nodes~~ 7th node and and structural element) at 15 s. Case 1: No ordering of particles $f_{n,i}^{(j)}$ at each node and structural element (random distribution). Case 2: With ordering of particles $f_{n,i}^{(j)}$ at each node and structural element with respect to their likelihood values

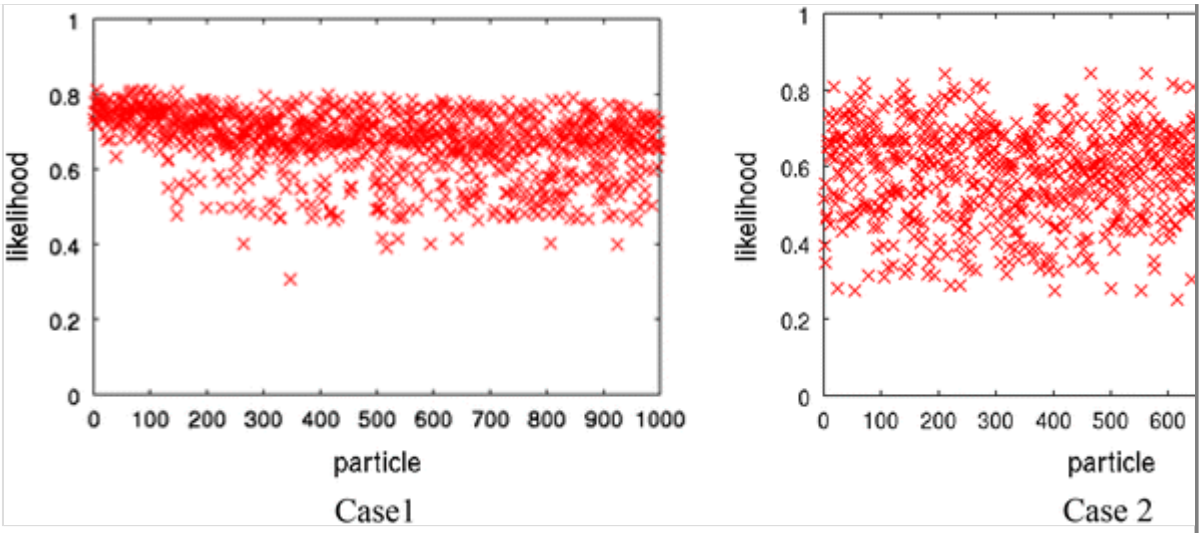


Fig. 4

Distribution of likelihood of recomposed particles $f_n^{(j)}$ from each node and structural element particles $f_{n,i}^{(j)}\,(i=1,2\ldots,M)$ at 15 s. Case 1: No ordering of particles $f_{n,i}^{(j)}$ at each node and structural element (random distribution). Case 2: With ordering of particles $f_{n,i}^{(j)}$ at each node and structural element with respect to their likelihood values

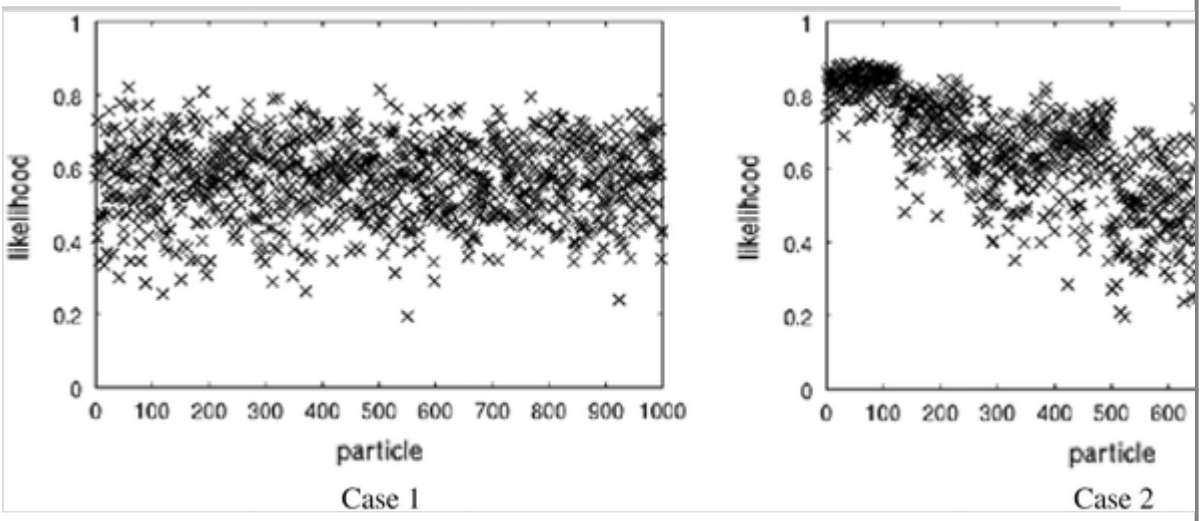


Fig. 5

Time histories of mean of identified stiffness at odd number nodes (10 degrees of freedom system). Case 1: No ordering of particles $f_{n,i}^{(j)}$ at each node and structural element (random distribution). Case 2: With ordering of particles $f_{n,i}^{(j)}$ at each node and structural element with respect to their likelihood values

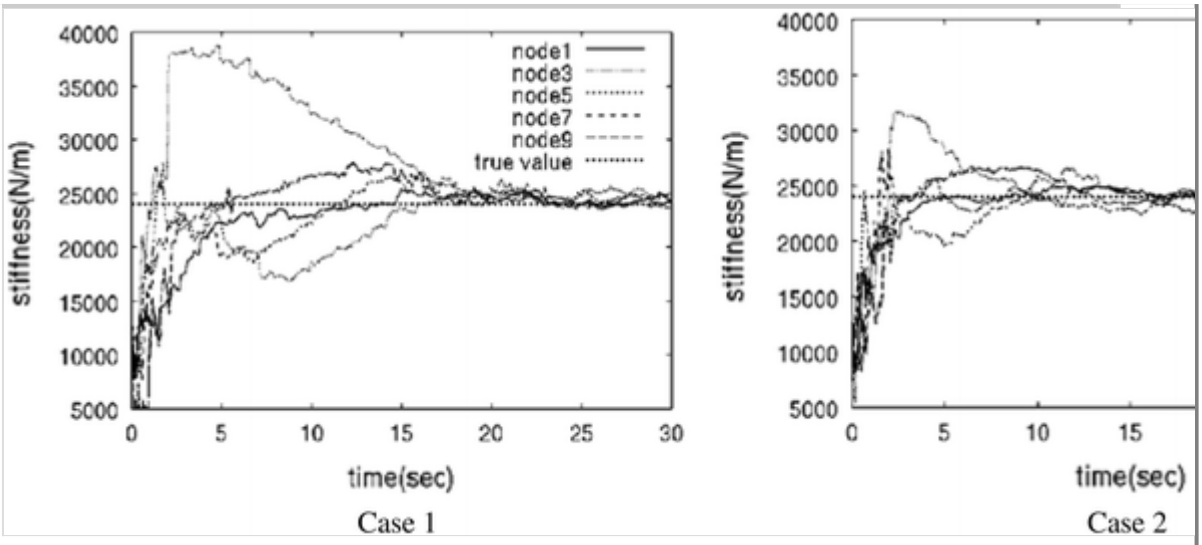
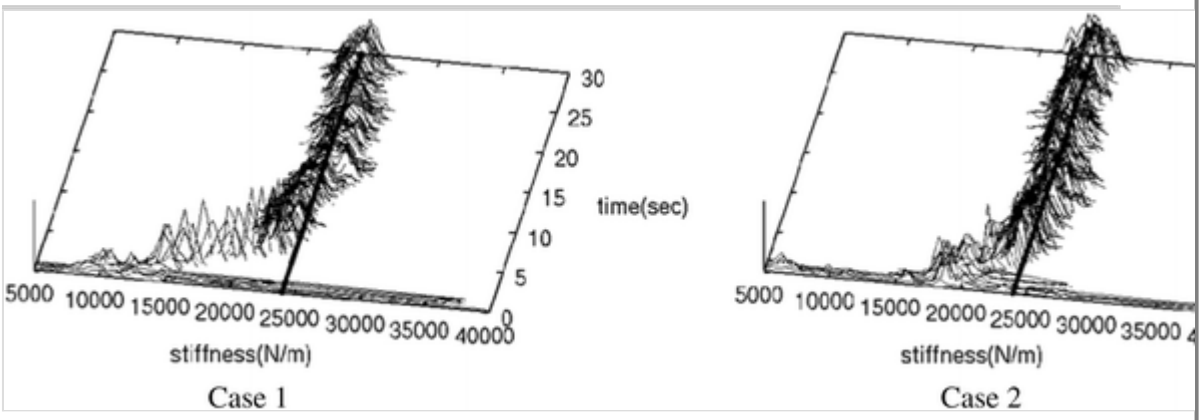


Fig. 6

Time history of probability density function of particles at structural element 1. Case 1: No ordering of particles $f_{n,i}^{(j)}$ at each node and structural element (random distribution). Case 2: With ordering of particles $f_{n,i}^{(j)}$ at each node and structural element with respect to their likelihood values



Although we conducted structural parameters identification for the same structure model using the classical MCF, we could not identify even if we used 40,000 particles with a computation time of 2 h and 30 min. This shows that the proposed method can identify a large degrees of freedom system very efficiently.

8. Combining genetic algorithm with the particle relaxation algorithm

We improved convergence speed to the true value by combining GA with the proposed RMCF (abbreviated as GARMCF). In this method, we introduce the mutation and crossover operations into RMCF to widen the search area of structural parameters in RMCF. Mutation generates particles beyond the limit of particle distribution in the proposed RMCF. Crossover gives more variation to particles by exchanging components between two state values. Based on these GA operations, the GARMCF generates particles which cannot be generated by RMCF and can improve convergence speed of the proposed algorithm. In this study, we apply mutation operation for both stiffness and damping coefficient and crossover operation only for damping coefficient. The proposed method consisted of the following steps assuming that the Case 2 process in RMCF is effective:

1. Generate *n* initial particles $x_{0|0}^{(j)}\ (j=1,2,\ldots,n)$ using the given initial probability density function $p_0\left(x_{0|0}\right)$.

2. Set $t = 1$ and repeat the following steps until the end of time steps.

(a) Generate *n* particles of system noise $w_{t-1}^{(j)}\ (j=1,2,\ldots,n)$ using the system noise probability distribution function $p(w_{t-1})$.

(b) Calculate the prediction particle by

$$x_{t|t-1}^{(j)} = F\left(x_{t-1|t-1}^{(j)}, w_{t-1}^{(j)}\right) \tag{30}$$

(c) Choose *k* particles with large likelihood from the mother set of $x_{t-1|t-1}^{(j)}$ and assemble a set of filter particles $\Gamma = \{x_{t-1|t-1}^{(1)}, \ldots, x_{t-1|t-1}^{(k)}\}$. Remaining $n - k$ particles are intact and just go through the RMCF process. A set of prediction particles obtained by substituting Γ into Eq. (30) is defined as $B = \{x_{t|t-1}^{(1)}, \ldots, x_{t|t-1}^{(k)}\}$.

(d)

Generate a new set of filter particles by applying GA operation to Γ and calculate a new set of prediction particles $\acute{B} = \left\{ \acute{x}_{t|t-1}^{(1)}, \ldots, \acute{x}_{t|t-1}^{(k)} \right\}$ using Eq. (30)

(e) Subdivide the prediction particles $x_{t|t-1}^{(j)}$ and $\acute{x}_{t|t-1}^{(j)}$ into each node and structural element state variables $\left\{ x_{t|t-1;1}^{(j)}, \ldots, x_{t|t-1;i}^{(j)}, \ldots, x_{t|t-1;M}^{(j)} \right\}$ and $\left\{ \acute{x}_{t|t-1;1}^{(j)}, \ldots, \acute{x}_{t|t-1;i}^{(j)}, \ldots, \acute{x}_{t|t-1;M}^{(j)} \right\}$. And repeat the following steps for node index i from 1 to M:

(i) Calculate the likelihood $\alpha_{t,i}^{(j)}$ and $\acute{\alpha}_{t,i}^{(j)}$ of $x_{t|t-1;i}^{(j)}$ and $\acute{x}_{t|t-1;i}^{(j)}$ by

$$\alpha_{t,i}^{(j)} = r\left(G\left(y_{t,i}, x_{t|t-1;i}^{(j)} \right) \right) \left| \frac{\partial G}{\partial y_{t,i}} \right| \tag{31}$$

$$\acute{\alpha}_{t,i}^{(j)} = r\left(G\left(y_{t,i}, \acute{x}_{t|t-1;i}^{(j)} \right) \right) \left| \frac{\partial G}{\partial y_{t,i}} \right| \tag{32}$$

(ii) Compare $\alpha_{t,i}^{(j)}$ and $\acute{\alpha}_{t,i}^{(j)}$. When $\acute{\alpha}_{t,i}^{(j)}$ is larger than $\alpha_{t,i}^{(j)}$, then replace $x_{t|t-1;i}^{(j)}$ and $\alpha_{t,i}^{(j)}$ with $\acute{x}_{t|t-1;i}^{(j)}$ and $\acute{\alpha}_{t,i}^{(j)}$ respectively.

(iii) Generate $x_{t|t;i}^{(j)}$ by resampling $x_{t|t-1;i}^{(j)}$ based on $\alpha_{t,i}^{(j)}$ as follows:

$$x_{t|t;i}^{(j)} = x_{t|t-1;i}^{(j)} \quad \text{with the probability} \quad \frac{\alpha_{t,i}^{(j)}}{\sum_{l=1}^{m} \alpha_{t,i}^{(l)}} \tag{33}$$

(f) Rearranging array of $x_{t|t;i}^{(j)}$ as the order of large likelihood and recompose $x_{t|t}^{(j)}$ by assembling each node particles as $\left\{ x_{t|t;1}^{(j)}, \ldots, x_{t|t;i}^{(j)}, \ldots, x_{t|t;M}^{(j)} \right\}$.

(g) Return to (a) by setting $t = t + 1$.

9. Numerical example 3

Through the following numerical example, we demonstrate the efficiency of the GARMCF. The structural model used is a shear type 20 degrees of freedom system. Conditions for identifying structural parameters are the same as given in the numerical example 2 (10 DOF). Two cases are considered. Case 1 is an application of RMCF and the other is the case of using GARMCH (Case 2). The number of particles used is 1,000 at each node and structural element. The number of particles to adapt the GA operation is 100. The mutation process is the same as mentioned in the numerical example 1 and crossover rate is 30 % at each structural element. Figure 7 shows the time history of mean stiffness obtained from the filtered particles at each time step. Figure 8 is the time history of probability density function of the stiffness at the first structural element. From both figures, we can see that convergence times to the true value in Case 2 are earlier than Case 1. Execution times of programs using the same computer in the numerical example 2 are 19 min and 49 s in Case 1 and 21 min and 38 s in Case 2. Computation time in Case 2 is longer than Case 1 because of the GA operation and comparison of likelihood in the GARMCF, but considering the improvement of convergence speed, this increase of computation time is not a disadvantage of GARMCF algorithm. These results show us that just simple implement of GA algorithm into RMCF algorithm improves computation efficiency dramatically.

Fig. 7

Time histories of mean of identified stiffness at every five nodes (20 degrees of freedom system)

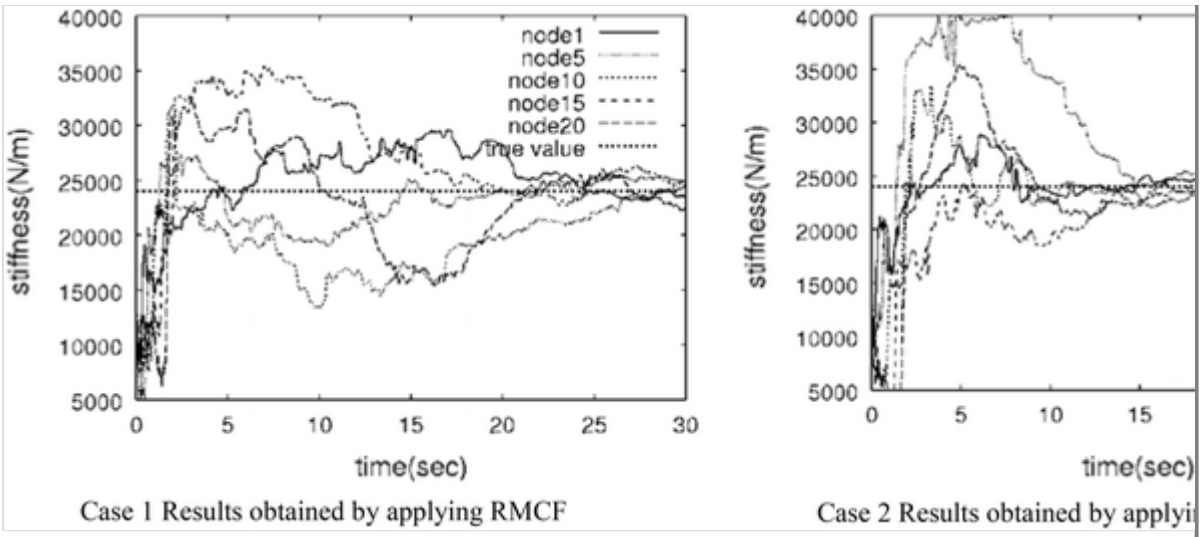
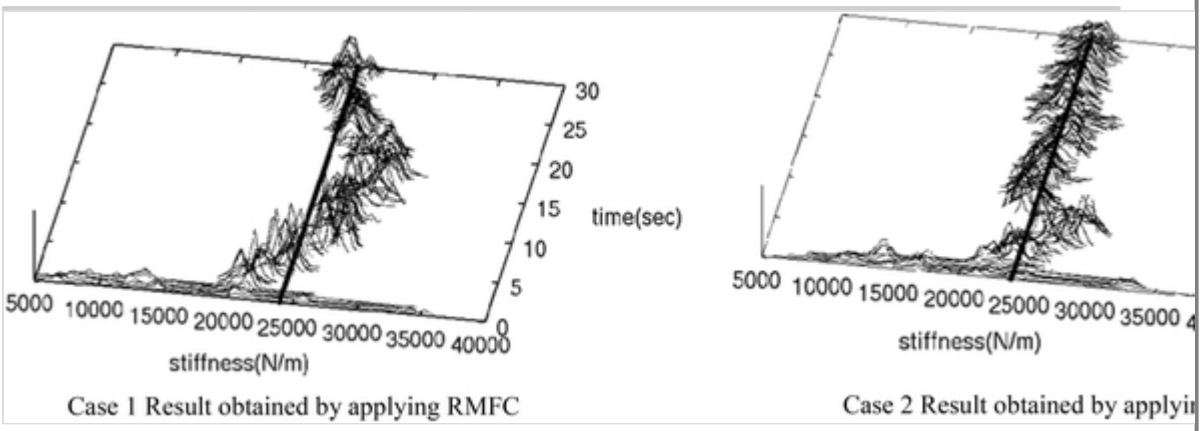


Fig. 8

Time history of probability density function of particles at structural element 1



10. Concluding remark

By modifying algorithm and applying the classical Monte Carlo filter to structural identification problems, we developed a new algorithm applicable to large structural systems identification. The idea is to calculate the likelihood of prediction particles for resampling of filtered particle by splitting prediction particle components into each set of structural node and structural element. We named it as relaxation Monte Carlo filter (RMCF) and demonstrated its efficiency comparing its convergence characteristics with that of the classical MCF. Moreover, we improved convergence speed of RMCF by combining the genetic algorithm. The efficiency of the proposed methods is also confirmed by applying it to the identification of large-scale structural model. MCF

covers very wide area of system filtering problems. It can be applicable to solve not only nonlinear but also non-Gaussian problems. However, it is necessary a proper modeling of constitutive relationship expressing system response to apply proposed methods for identifying nonlinear characteristics, this subject is a future topic of GAMCF algorithm.

References

1. Kalman RE (1960) A new approach to linear filtering and prediction problems. ASCE J Basic Eng 82D(1):35–45

2. Yun CB, Shinozuka M (1980) Identification of nonlinear structural dynamics systems. J Struct Mech 8(2):187–203

3. Hoshiya M, Saito E (1984) Structural identification by extended Kalman Filter. J Eng Mech ASCE 110(12):1757–1770

4. Loh CH, Chung ST (1993) A three-stage identification approach for hysteretic systems. Earthq Eng Struct Dynam 22:85–97

5. Smyth AW, Marsi SF, Chassiakos AG, Caughey TK (1999) On-line parametric identification of MDOF nonlinear hysteretic systems. J Eng Mech ASCE 125(2):133–142

6. Sato T, Qi K (1998) Adaptive H infinite filter: its application to structural identification. J Eng Mech ASCE 124(11):1233–1240

7. Sato T, Takei K (1998) Development of a Kalman filter with fading memory. In: Proceedings of structural safety and reliability, pp 387–394

8. Takaba K, Katayama T (1996) Discrete-time H infinite algebraic Ricatti equation and parameterization of all H infinite filters. Int J Control 64(6):1129–1149

9. Simon JJ, Uhlmann JK (1997) A new extension of the Kalman filter to nonlinear systems. In: Proceedings of AeroSense: The 11th international symposium on aerospace/defense sensing, simulation

and controls, multi sensor fusion, tracking and resource management II, SPIE, Boston

10. Gordon NJ, Salmond DJ, Smith AFM (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEEE Proc Radar Signal Proc 140(2):107–113

11. Doucet A, De Freitas N, Gordon NJ (2001) Sequential Monte Carlo methods in practice, Springer, New York

12. Kitagawa G (1996) Monte Carlo filter and smoother for Non-Gaussian nonlinear state space model. J Comput Gr Stat 5(1):1–25

13. Sato T, Kaji K (2001) Adaptive Monte Carlo filter and structural identification. In: Proceedings of the first international conference on Monte Carlo simulation, Monte Carlo, Monaco, pp 441–447

14. Yoshida I, Sato T (2002) Health monitoring algorithm by the Monte Carlo filter based on non-Gaussian noise. J Nat Disaster Sci 24(2):101–107

15. Ching J, Beck JL, Porter KA (2006) Bayesian state and parameter estimation of uncertain dynamical systems. Probab Eng Mech 21:81–96

16. Chowdhury SR, Roy D, Vasu RM (2012) Variance reduced particle filters for structural system identification problems. J Eng Mech ASCE [posted ahead of print 28 May 2012]

17. Nasrellahm HA, Manohar CS (2010) A particle filtering approach for structural system identification in vehicle-structure interaction problems. J Sound Vib 329:1289–1309

18. Namdeo V, Manohar CS (2007) Nonlinear structural dynamical system identification using adaptive particle filters. J Sound Vib 306:524–563

19. Sajeeb R, Manohar CS, Roy D (2009) A conditionally linearized Monte Carlo filter in non-linear structural dynamics. Int J Non-Linear Mech 44:776–790

20. Sajeeb R, Manohar CS, Roy D (2010) A semi-analytical particle filter for identification of nolinear oscillators. Probab Eng Mech 25:35–48

21. Sajeeb R, Manohar CS, Roy D (2007) Use of particle filters in an active control algorithm for noisy nonlinear structural dynamical systems. J Sound Vib 306:111–135

22. Chang S-Y, Chowhan T, Latif S (2012) State and parameter estimation with a sequential importance resampling (SIR) particle filter in a three dimensional groundwater pollutant transport model. J Environ Eng ASCE [posted ahead of print 30 April 2012]

23. Samanta B (2009) Engineering system fault detection using particle filters. In: Proceedings of ASME 2009 international design engineering technical conference and computers and information in engineering conference, Paper no. DEC2009-86809, pp 95–101

24. Holland J~~H~~ (19~~75~~92) Adaptation in natural and artificial systems. ~~The University of Michigan Press, Ann Abor~~Cambridge, MA:MIT Press.

25. Ristic B, Arulampalam S, Gordon N, Beyond the Kalman Filter (2004) Particle filters for tracking applications, Artech House

26. Higuchi T (1996) Genetic algorithm and Monte Carlo filter. Proc Inst Stat Math 44:19–30 (in Japanese)