Smart Structures and Systems, Vol.11, No. 1 (2013) 53-67
DOI: http://dx.doi.org/10.12989/sss.2013.11.1.053

Particle relaxation method for structural parameters identification based on Monte Carlo Filter

Tadanobu Sato*[1] and Youhei Tanaka[2]

[1]*International Institute of Urban engineering, Southeast University, Nanjing, China*
[2]*Minicipality of Nishinomiya, Nishinomiya, Japan*

(Received June 11, 2012, Revised November 25, 2012, Accepted November 30, 2012)

Abstract. In this paper we apply Monte Carlo Filter to identifying dynamic parameters of structural systems and improve the efficiency of this algorithm. The algorithms using Monte Carlo Filter so far has not been practical to apply to structural identification for large scale structural systems because computation time increases exponentially as the degrees of freedom of the system increase. To overcome this problem, we developed a method being able to reduce 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 demonstrate 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. Using shaking table test data of a model structure we also demonstrate the efficiency of proposed algorithm.

Keywords: Monte Carlo Filter; particle filter; system identification; genetic algorithm; relaxation technique; large scale system; shaking table test; model structure

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 (Kalman1960) 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 (1980), Hoshiya and Saito (1984), Loh and Chung (1993), Smyth *et al*. (1999), Sato and Kaji (1998), Sato and Takei (1998) and Takaba and Katayama (1996).

*Corresponding author, Professor, E-mail: satotdnbseu@yahoo.co.jp

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 this method the Unscented Kalman Filter is proposed (Simon *et al*. 1997) 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 (Gordon *et al*. 1993) known as a particle filter (Doucet *et al*. 2001), 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 the boot strap method developed, Kitagawa (1996) 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 achieved good successes in structural identification fields by many researchers such as Sato and Kaji (2001), Yoshida and Sato (2002), Ching *et al*. (2006), Chowdhury *et al*. (2012), even a nonlinear system identification problem (Namdeo and Manohar 2007), three dimensional grand water flow problem (Chang *et al*. 2012) and a machine fault detection problem (Samanta 2012). However, these methods have not been practical to apply to structural identification for large degree of freedom systems because of exponential incensement of computation time as the system becomes very large. This is because the classical MCF had a 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 degree 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) (Holland 1975) 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

$$x_n = F(x_{n-1}, w_n) \tag{1}$$

$$y_n = H(x_n, v_n) \tag{2}$$

in which, n is the discrete time step, x_n is the state vector, y_n is the observation vector, w_n is the system noise vector and v_n is the observation noise vector which is assumed to be expressed by

$$v_n = H^{-1}(y_n, x_n) = G(y_n, x_n) \tag{3}$$

The MCF can be applied even if the state space model is non-linear 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 $p(x_n|Y_n)$ instead of identifying the estate vector x_n directly. In which Y_n=$\{y_n, y_{n-1}, \cdots, y_1\}$. We called $p(x_n|Y_{n-1})$ as the prediction distribution and $p(x_n|Y_n)$ as the filter distribution, and each probability density function is approximated by m realizations as follows

$$b_n^{(j)} = \left\{b_n^{(1)},\ b_n^{(2)},\ \cdots, b_n^{(m)}\ \right\} \sim p(x_n|Y_{n-1}) \tag{4}$$

$$f_n^{(j)} = \left\{f_n^{(1)},\ f_n^{(2)},\ \cdots, f_n^{(m)}\ \right\} \sim p(x_n|Y_n) \tag{5}$$

in which $b_n^{(j)}$ and $f_n^{(j)}$ are respectively the jth prediction and filter particles which approximate prediction and filter distributions. The MCF algorithm is summarized as follows

1. Generate m initial particles $f_0^{(j)}$ $(j = 1,2,\cdots,m)$ using the given initial probability density function $p_0(x_0)$. m is the number of particles used in MCF.
2. Repeat the following steps until the end of steps after setting $n = 1$.

(a) Generate m samples of system noise $w_n^{(j)}$ using the system noise probability distribution function $p(w_n)$.

(b) Calculate the prediction particle by

$$b_n^{(j)} = F\left(f_{n-1}^{(j)}, w_n^{(j)}\right) \tag{6}$$

(c) Calculate the likelihood of each particle by

$$\alpha_n^{(j)} = p\left(y_n\middle|b_n^{(j)}\right) = r\left(G\left(y_n, b_n^{(j)}\right)\right)\left|\frac{\partial G}{\partial y_n}\right| \tag{7}$$

in which r is the probability density function of observation noise.

(d) Generate $f_n^{(j)}$ by resampling $b_n^{(j)}$ based on $\alpha_n^{(j)}$ as follows

$$f_n^{(k)} = b_n^{(k)} \text{ with the probability } \frac{\alpha_n^{(k)}}{\sum_{j=1}^{m}\alpha_n^{(j)}} \tag{8}$$

in which the total sum of k is restricted to be equal to m.

(e) Set n= $n + 1$ and return to (a) until the end of time step.

3. System and observation equations

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

$$m_i\left(\ddot{z}_i + \ddot{X}_g\right) + c_i\dot{u}_i + k_i u_i - (1 - \delta_{iN})(c_{i+1}\dot{u}_{i+1} + k_{i+1}u_{i+1}) = 0 \tag{9}$$

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

$$x_n = \{\cdots,\ z_i, \dot{z}_i, c_i, k_i,\ \cdots\}^T \tag{10}$$

$$y_n = \{\cdots,\ z_i, \dot{z}_i,\ \cdots\}^T \qquad i = 1, 2, \cdots, N \tag{11}$$

The state transfer equation is expressed by the following equation

$$x_n = x_{n-1} + \int_{n-1}^{n} g(x_{n-1})dt + w_n \tag{12}$$

in which g is expressed as follows

$$g = \begin{Bmatrix} \vdots \\ \dot{z}_i \\ -\frac{c_1}{m_i}\dot{u}_i - \frac{k_i}{m_i}u_i + \frac{1-\delta_{iN}}{m_i}(c_{i+1}\dot{u}_{i+1} + k_{i+1}u_{i+1}) - \ddot{X}_g \\ 1 \\ 1 \\ \vdots \end{Bmatrix} \tag{13}$$

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

$$y_n = Hx_n + v_n \tag{14}$$

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

$$H = \begin{bmatrix} D & & & \\ & D & & \\ & & \ddots & \\ & & & D \end{bmatrix} \qquad D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{15}$$

We can apply the MCF to structural identification using Eqs. (12) and (14) as the state space model.

4. Proposed algorithm

In the past application of MCF the state vector is a vector with the dimension of $4N$ in the case of a N degrees of freedom system, so each particle is composed of N set of 4 state values; displacement and velocity of each node, as well as stiffness and damping coefficient of each layer. As the degree of freedom increases, the number of combinations of $4N$ state values increases exponentially. Therefore, to assign enough variation to particles to choose 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 to apply to structural identification

for large degree of freedom systems.

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 resampling particles based on the likelihood of prediction particles of whole system we conduct resampling at each node or layer 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 m initial particles $f_0^{(j)}$ $(j = 1,2,\cdots,m)$ using the given initial probability density function $p_0(x_0)$.

2. Repeat the following steps until the end of time steps after setting $n = 1$.

(a) Generate m samples of system noise $w_n^{(j)}$ using the system noise probability distribution function $p(w_n)$.

(b) Calculate the prediction particle by

$$b_n^{(j)} = F\left(f_{n-1}^{(j)}, w_n^{(j)}\right) \tag{16}$$

(c) Subdivide the predicted particles $b_n^{(j)}$ into N components composed of the state variables at each node and layer $b_{n,i}^{(j)}$ $(i = 1,2,\cdots,N)$ in which $b_n^{(j)}$ and $b_{n,i}^{(j)}$ are defined by

$$b_n^{(j)} = \left\{b_{n,1}^{(j)}, \cdots, b_{n,i}^{(j)}, \cdots, b_{n,N}^{(j)}\right\}^T \tag{17}$$

$$b_{n,i}^{(j)} = \{z_i, \dot{z}_i, c_i, k_i\}_n^{(j)} \quad (i = 1,2,\cdots,N) \tag{18}$$

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

i. Calculate the likelihood of $b_{n,i}^{(j)}$ by

$$\alpha_{n,i}^{(j)} = r\left(G\left(y_{n,i}, b_{n,i}^{(j)}\right)\right)\left|\frac{\partial G}{\partial y_{n,i}}\right| \tag{19}$$

in which $y_{n,i} = \{z_i, \dot{z}_i\}$ $(i = 1,2,\cdots N)$

ii. Generate $f_{n,i}^{(j)}$ by resampling $b_{n,i}^{(j)}$ based on $\alpha_{n,i}^{(j)}$ as follows

$$f_{n,i}^{(k)} = b_{n,i}^{(k)} \text{ with the probability } \frac{\alpha_{n,1}^{(k)}}{\sum_{j=1}^{m}\alpha_{n,i}^{(j)}} \tag{20}$$

iii. Return to i.

(d) Recompose $f_n^{(j)}$ as expressed by $f_n^{(j)} = \left\{f_{n,1}^{(j)}, \cdots, f_{n,i}^{(j)}, \cdots, f_{n,N}^{(j)}\right\}^T$ for each index j (we call this is the assembling process) in which we consider two cases. Case 1 is just to recompose $f_n^{(j)}$ by assembling $f_{n,i}^{(j)}$ $(i = 1,2,\cdots,N)$ for an arbitrary assigned order of j, Case 2 is to recompose $f_n^{(j)}$ by assembling $f_{n,i}^{(j)}$ $(i = 1,2,\cdots,N)$ for j which is arrayed in order of lage likelihood of

each particle.

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

In this method each prediction particle at a node and layer i, i.e., $b_{n,i}^{(j)}$ $(j = 1,2,\cdots,m)$, is a 4 dimensional vector and only composed of 4 state variables as defined by Eq. (18). This means that the necessary number of particles in the filtering process defined by Eq. (20) 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 m the total number of particles applying RMCF to a N degree of freedom structural system becomes enough with N multiply m. If we use the original MCF for identifying the N degree of freedom structural system we need to generate m^N combination of particles. Based on this proposed algorithm we can reduce dramatically the computation time. However the deficit is that a recomposed filtered particle of the structural system $f_n^{(j)}$ dose not satisfy exactly the equation of motion at the time step n and has a small residual error. This deficiency 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 $n + 1$ using Eq. (16).

4.1 Numerical example 1

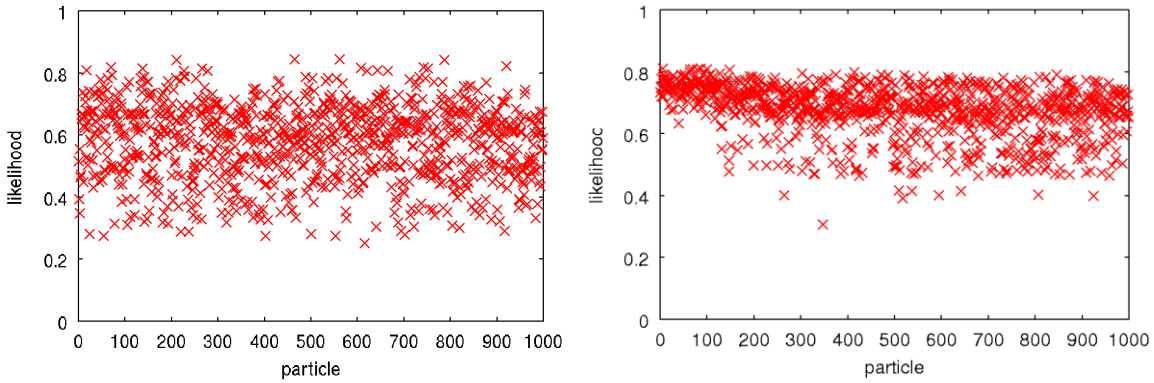
Through the following numerical example, we demonstrate the efficiency of the RMCF algorithm. The structural model used is a 10 degrees of freedom system. The all nodes have the same mass, damping coefficient and stiffness which are respectively 123.02 (kg), 68 .606 (N·sec/m), 24008.69 (N/m). The input motion is the El Centro (NS1940) accelerogram whose maximum acceleration is modified to 25(gal). First, we simulate the structural responses and we 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 1000 for each node. We compare effect of two assembling process to recompose the filtered particles of the whole system, $f_n^{(j)}$, on the numerical convergence speed.

Fig. 1 shows the distribution of likelihood of particles at a node and layer, as example $f_{n,7}^{(j)}$ (paticle at 7 node and layer), at 15 seconds.

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 characteristic are also seen in the relationship between the particle number and likelihood at all other nodes. Fig. 2 shows the distribution of likelihood of $f_n^{(j)}$ (a recomposed particle expressing filtered state vector) at 15 seconds. 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 layer $f_{n,i}^{(j)}$ before assembling to recompose $f_n^{(j)}$. Fig. 3 shows the time history of mean of the identified particles for stiffness in the layer with odd numbers. Fig. 4 shows the time history of probability density function of stiffness in the first layer. 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 minutes 36 seconds in Case 1 and 5

minutes 56 seconds in Case 2. Computation time in Case 2 is a little bit longer than Case 1 because of rearranging the array of $f_{n,i}^{(j)}$ before recomposing $f_n^{(j)}$. But considering improvement of convergence speed, this increase of computation time is not disadvantage.

Although we conducted structural parameter identification for the same structure model using the classical MCF, we could not identify even if we used 40000 particles with a computation time of 2 hours and 30 minutes. This shows that the proposed method can identify a large degrees of freedom system very efficiently.

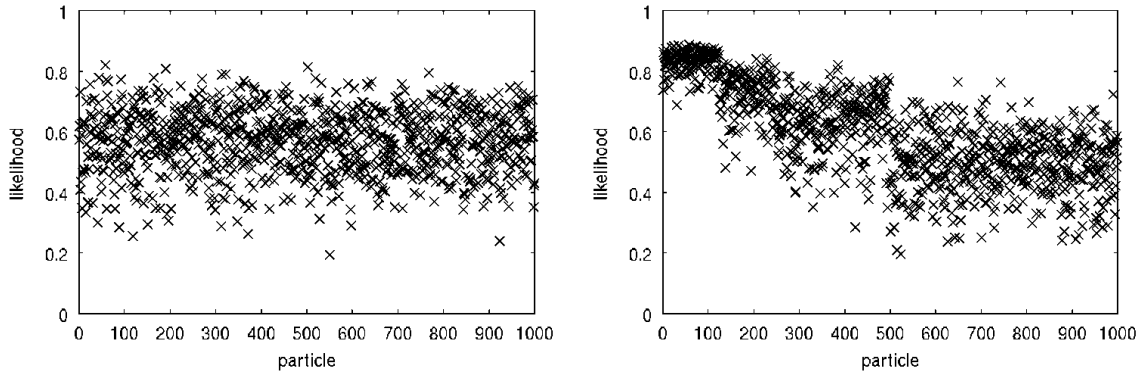


(a) Case 1 (b) Case 2

Fig. 1 Distribution of likelihood of $f_{n,7}^{(j)}$ (7 node and layer) at 15seconds

Case 1: No ordering of particles $f_{n,i}^{(j)}$ at each node and layer (random distribution)

Case 2: With ordering of particles $f_{n,i}^{(j)}$ at each node and layer with respect to their likelihood values

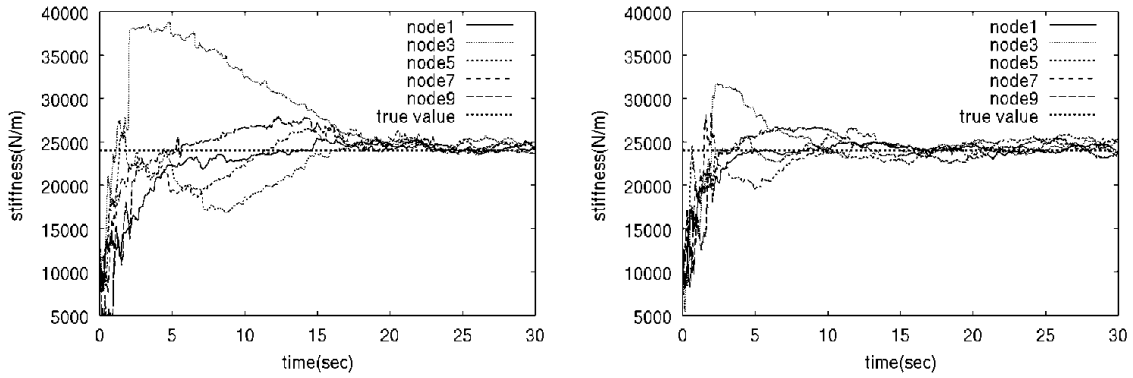


(a) Case 1 (b) Case 2

Fig. 2 Distribution of likelihood of recomposed particles $f_n^{(j)}$ from each node and layer particles $f_{n,i}^{(j)}$ (i=1,2,$\cdots$, N)

Case 1: No ordering of particles $f_{n,i}^{(j)}$ at each node and layer (random distribution)

Case 2: With ordering of particles $f_{n,i}^{(j)}$ at each node and layer with respect to their likelihood values

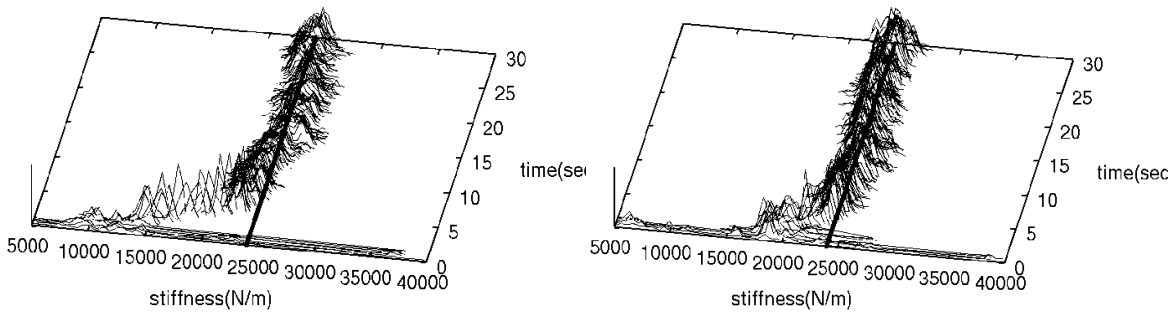


(a) Case 1 (b) Case 2

Fig. 3 Time histories of mean of identified stiffness at odd number nodes (10 degree of freedom system)

Case 1: No ordering of particles $f_{n,i}^{(j)}$ at each node and layer (random distribution)

Case 2: With ordering of particles $f_{n,i}^{(j)}$ at each node and layer with respect to their likelihood values



(a) Case 1 (b) Case 2

Fig. 4 Time history of probability density function of particles at node and layer 1

Case 1: No ordering of particles $f_{n,i}^{(j)}$ at each node and layer (random distribution)

Case 2: With ordering of particles $f_{n,i}^{(j)}$ at each node and layer with respect to their likelihood values

5. Combining genetic algorithm with the proposed algorithm

A study to investigate the similarity between Genetic Algorithm (GA) and Monte Caro Filter was firstly done by Higuchi (1996). Both MCF and GA are algorithms to reconstruct a set of realization expressing the state values from a random set of initial values. We improved convergence speed to the true value by combining GA with the proposed RMCF (abbreviate as GARMCF). In this method we introduce the mutation and crossover operations into RMCF to widen the searching 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 generate particles which cannot be generate by RMCF and can improve convergence speed of 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 is consisted of the following steps assuming that the Case 2 process in RMCF is effective:

1. Generate the initial particle distribution $f_0^{(j)}$ base on the initial probability distribution function $p_0(x_0)$ and set $n = 1$

2. Repeat the following steps until the end of time steps.

(a) Generate the system noise $w_n^{(j)}$ based on the probability density function $q(w_n)$

(b) Calculate the prediction particle by

$$b_n^{(j)} = F\left(f_{n-1}^{(j)}, w_n^{(j)}\right) \tag{21}$$

(c) Chose k particles with large likelihood from the mother set of $f_{n-1}^{(j)}$ and assemble a set of filter particles $\Gamma = \left\{f_{n-1}^{(1)}, \cdots, f_{n-1}^{(k)}\right\}$. Remaining $m-k$ particles are intact and just go through the RMCF process. A set of prediction particles obtained by substituting Γ into Eq. (21) is defined as $B = \{b_n^1, \cdots, b_n^k\}$.

(d) Generate a new set of filter particles by applying GA operation to Γ and calculate a new set of prediction particles $\acute{B} = \left\{\acute{b}_n^1, \cdots, \acute{b}_n^k\right\}$ by using Eq. (21)

(e) Subdivide the prediction particles $b_n^{(j)}$ and $\acute{b}_n^{(j)}$ into each node state variables $\left\{b_{n,1}^{(j)}, \cdots, b_{n,i}^{(j)}, \cdots, b_{n,N}^{(j)}\right\}$ and $\left\{\acute{b}_{n,1}^{(j)}, \cdots, \acute{b}_{n,i}^{(j)}, \cdots, \acute{b}_{n,N}^{(j)}\right\}$. And repeat the following steps for node index i from 1 to N:

i. Calculate the likelihood $\alpha_{n,i}^{(j)}$ and $\acute{\alpha}_{n,i}^{(j)}$ of $b_{n,i}^{(j)}$ and $\acute{b}_{n,i}^{(j)}$ by

$$\alpha_{n,i}^{(j)} = r\left(G\left(y_{n,i}, b_{n,i}^{(j)}\right)\right)\left|\frac{\partial G}{\partial y_{n,i}}\right| \tag{22}$$

$$\acute{\alpha}_{n,i}^{(j)} = r\left(G\left(y_{n,i}, \acute{b}_{n,i}^{(j)}\right)\right)\left|\frac{\partial G}{\partial y_{n,i}}\right| \tag{23}$$

ii. Compare $\alpha_{n,i}^{(j)}$ and $\acute{\alpha}_{n,i}^{(j)}$. When $\acute{\alpha}_{n,i}^{(j)}$ is larger than $\alpha_{n,i}^{(j)}$ then replace $b_{n,i}^{(j)}$ with $\acute{b}_{n,i}^{(j)}$.

iii. Generate $f_{n,i}^{(j)}$ by resampling $b_{n,i}^{(j)}$ based on $\alpha_{n,i}^{(j)}$ as follows

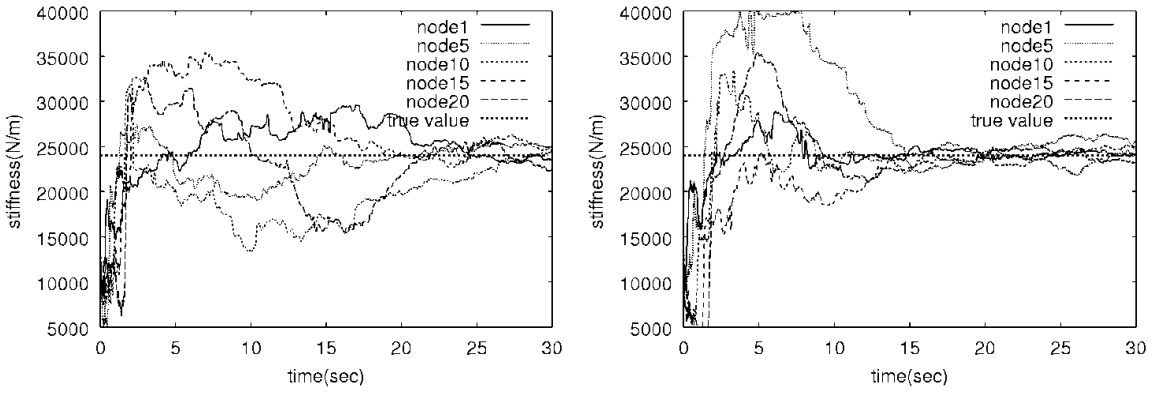
$$f_{n,i}^{(k)} = b_{n,i}^{(k)} \text{ with the probability } \frac{\alpha_{n,1}^{(k)}}{\sum_{j=1}^{m}\alpha_{n,i}^{(j)}} \tag{24}$$

(f) Rearranging array of $f_{n,1}^{(j)}$ as the order of large likelihood (original numbering of j is arranged as the order of large likelihood) and recompose $f_n^{(j)}$ by assembling each node particles as $\left\{f_{n,1}^{(j)}, \cdots, f_{n,i}^{(j)}, \cdots, f_{n,N}^{(j)}\right\}$.

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

5.1 Numerical example 2

Through the following numerical example we demonstrate the efficiency of the GARMCF. The structural model used is 20 degrees of freedom system. Conditions for identifying structural parameters are the same as given in the first numerical example (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 1000 at each node and layer. The number of particles to adapt the GA operation is 100 and crossover rate is 30% at each node and layer. Fig. 5 shows the time history of mean of the identified particles for stiffness. Fig. 6 is the time history of probability density function of the stiffness at the first layer. From both of 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 first numerical example are 19 minutes and 49 seconds in Case 1 and 21 minutes and 38 seconds 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 improvement of convergence speed this increase of computation time is not disadvantage of GARMCF algorithm. These results show us that just simple implement of GA algorithm into RMCF algorithm improves dramatically computation efficiency.



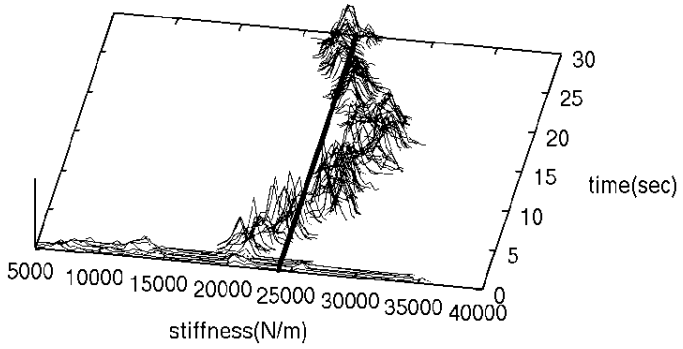
(a) Results only applying RMCF (b) Result applying GARMCF

Fig. 5 Time histories of mean of identified stiffness at every five nodes (20 degrees of freedom system)

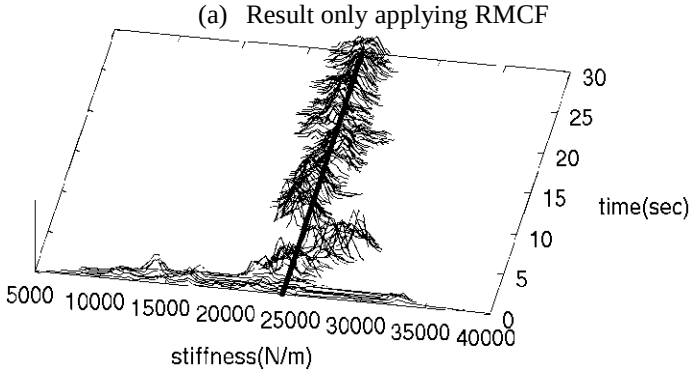
5.2 Application to experimental result

5.2.1 Brief introduction of the experiment

Here we apply the proposed GARMCF algorithm to identify model structural parameters using shaking table test data. A shaking table test was conducted for a model structure as shown in Fig. 7(a). This model is a five layered frame structure, and placed a base isolator as a column of model structure at four corners of each layer. As shown in Fig. 7(b) accelerometers and laser displacement meters are placed at symmetry points of model front and back sides in each layer. Sampling speed was 500Hz. In the experiment two types of base isolator were selected. One is a simple rubber bearing to obtain linear response behavior of structure and the other is a lead plugged rubber isolator to extract nonlinear behavior. In this paper we only use the experimental data of linear case. Input motion is an observed earthquake record at East Kobe Bridge during 1995 Kobe earthquake of which amplitude modified to 60 gal.



(a) Result only applying RMCF



(b) Result applying GARMCF

Fig.6 Time history of probability density function of particles at node and layer 1

5.2.2 Structural parameter identification of experimental model structure

We used only absolute acceleration data of all layers and base of which frequency components were modified by using a band pass filter with $0.5\sim10Hz$ band. Acceleration relative to base was obtained by extracting the base acceleration from the absolute acceleration of each layer. The relative velocity and displacement of each layer were calculated from relative accelerations by integrating using Newmark β method.

The mass of fifth layer is 2.6224 t, and the other layer mass has same value of 2.0968 t. We use calculated relative velocity and displacement as the observed values to identify the structural parameters. We assume uniform initial distributions of structural parameters, $U(0.0,5.0)$ for damping coefficients and $U(0.0,1000.0)$ for stiffness. The number of particles at each node is 1000. The number of particles applying GA operation is 100 and crossover rate is 20%.

Fig. 8 shows the time histories of mean value of filtered particles for stiffness and damping coefficient of each layer. Fig. 9 is the time history of probability density function of both values at the first layer. For damping coefficients time history is rather fluctuate but for stiffness it becomes stable during the time range from 15 to 25 seconds.

We also conduct analyses to check the efficiency of identified results by simulating model responses to the given input earthquake motion. The model structural parameters used for analyses are mean value of stiffness and damping coefficient between 15 and 25 seconds. Fig. 10 show the comparison between observed and calculated time history of relative acceleration at node 1 and 5 as well as hysteresis curve of layer 1 and 5.

Fig. 7(a) A five layered structural model on a shaking table. Each mass of layer is supported by four base isolators at the four corners of each layer

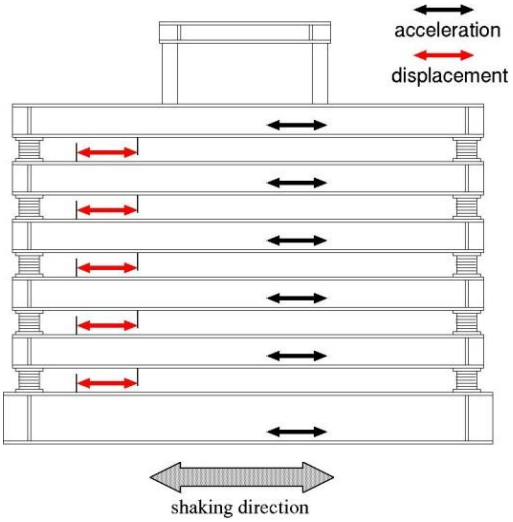
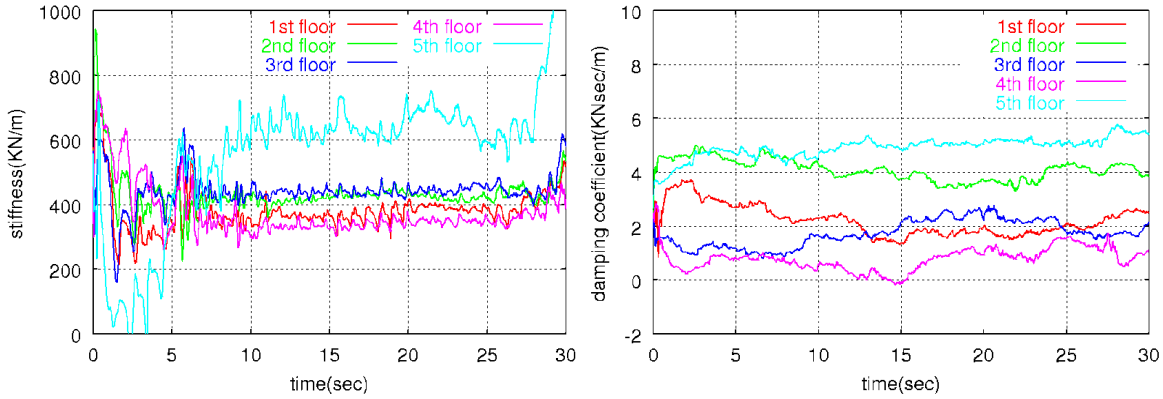
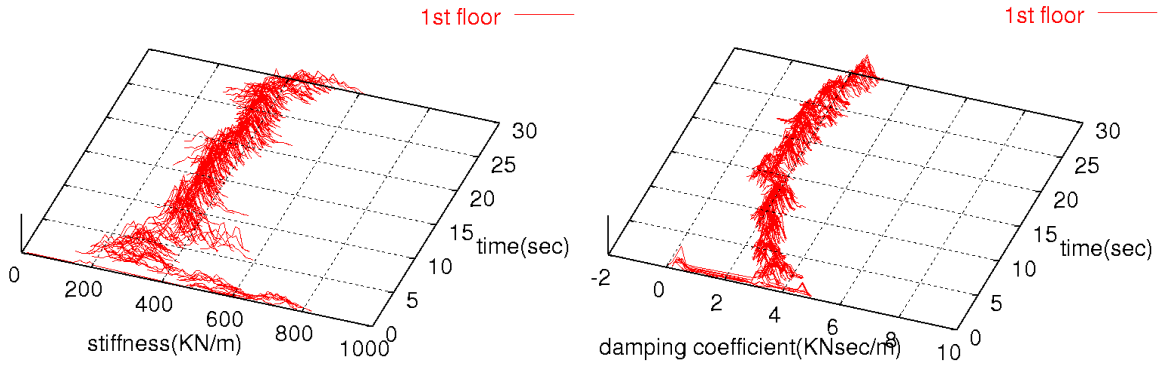


Fig. 7(b) Schematic positioning of accelerometers and displacement meters



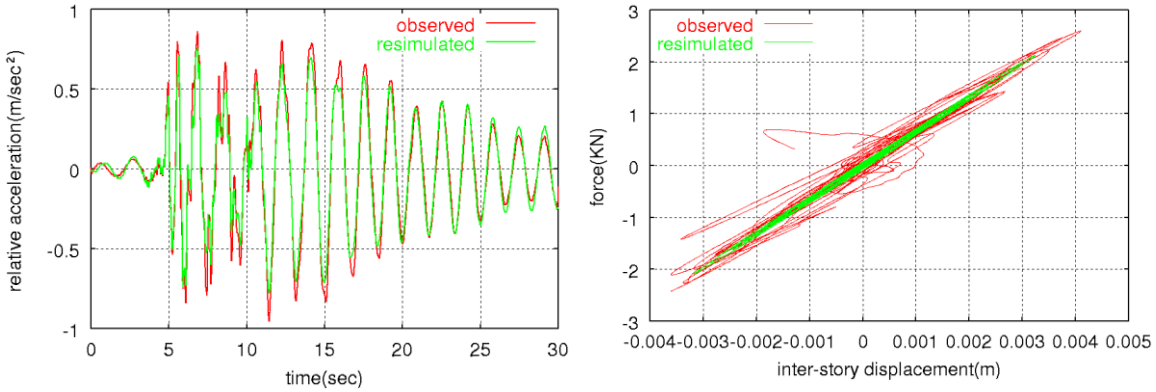
(a) Stiffness at each floor (b) Damping coefficient at each floor

Fig. 8 Time histories of mean values calculated from filtered particles using GARMCF



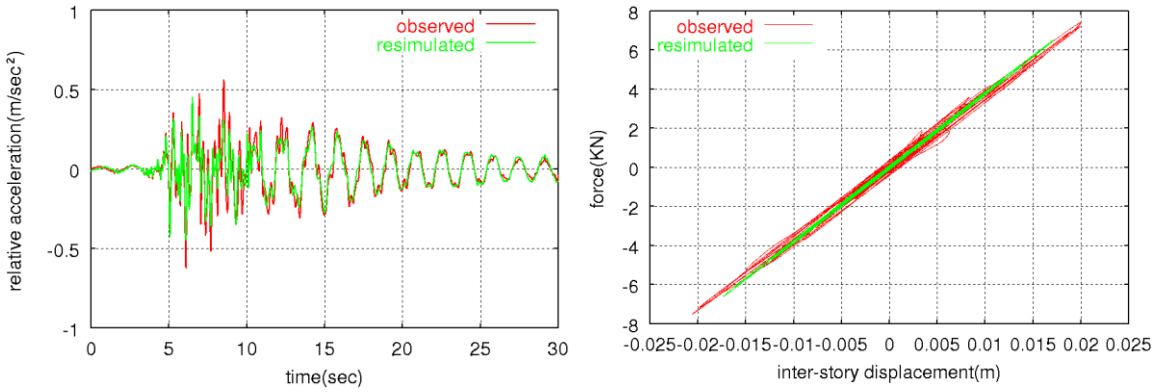
(a) Stiffness (b) Damping coefficient

Fig. 9 Time histories of probability density function of stiffness and damping coefficient at the first floor



(a) Acceleration response time history at node 5

(b) Hysteresis loop of fifth floor



(c) Acceleration response time history at node 1

(d) Hysteresis loop of first floor

Fig. 10 Comparison of time histories of acceleration response and hysteresis loops between observed and calculated results (Calculated results are obtained using mean value of identified time history of stiffness and damping coefficient during 15-25sec)

Because identification of damping coefficients is unstable we can see a little bit difference on the amplitude but because of good stability of identified stiffness overall response tendencies in phase and gradient of hysteresis loops are well agreed with experimental data. In the structural parameter identification process using RMFC damping coefficients go into sometimes negative region and those time history fluctuate strongly, even divergence phenomenon can be seen. Therefore we showed here only the results using GARMCF algorithm. One reason of occurring unstable identification process is that we modeled the complicated structural system by a simple shear vibration system. But searching process of solution by MCF has not been clearly investigated including convergence to local minimum solution we have to study more to apply MCF for practical problems.

6. Conclusions

By modifying algorithm 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 likelihood of prediction particles for resampling of filtered particle by splitting prediction particle components into each set of structural node and layer. We named it as Relaxation Monte Carlo Filter (RMCF) and demonstrate 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 proposed methods is also confirmed by not only applying it to the identification of large scale structural model but also identifying model structure parameters using shaking table test data. 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

Ching, J., Beck, J.L. and Porter, K.A. (2006), "Bayesian state and parameter estimation of uncertain dynamical systems", *Probabilist. Eng. Mech.*, **21**(1), 81-96.

Chang, S.Y., Chowhan, T. and 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. Enig. - ASCE*, posted ahead of print 30 April 2012.

Chowdhury, S.R., Roy, D. and Vasu, R.M. (2012), "Variance reduced particle filters for structural system identification problems", *J. Eng. Mech. - ASCE*, posted ahead of print 28 May 2012.

Doucet, A., De Freitas, N. and Gordon, N.J. (2001), *Sequential Monte Carlo Methods in Practice*, Springer

Gordon, N.J., Salmond, D.J. and Smith, A.F.M. (1993), "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", *IEEE Proc. Radar Signal Proc.*, **140**(2), 107-113.

Higuchi, T. (1996), "Genetic algorithm and Monte Carlo Filter", *Proceedings of the Institute Statistical Mathematics*, 44, 19-30. (in Japanese)

Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*, the University of Michigan Press, Ann Abor.

Hoshiya, M. and Saito, E. (1984), "Structural identification by extended Kalman Filter", *J. Eng. Mech.-ASCE*, **110**(12), 1757-1770.

Kalman, R.E. (1960), "A new approach to linear filtering and prediction problems", *J. Basic Eng.- ASCE* **82D** (1), 35-45.

Kitagawa, G. (1996), "Monte Carlo Filter and smoother for Non-Gaussian nonlinear state space model", *J. Comput.Graph. Stat.*, **5**(1), 1-25.

Loh, C.H. and Chung, S.T. (1993), "A three-stage identification approach for hysteretic systems", *Earthq. Eng. Struct. D.*, **22**, 85-97.

Namdeo, V. and Manohar, C.S. (2007), "Nonlinear structural dynamical system identification using adaptive particle filters", *J. Sound Vib.*, **306**(3-5), 524-563.

Samanta, B. (2009), "Engineering system fault detection using particle filters", *Proceedings of ASME 2009 International Design engineering Technical Conference and Computers and Information in engineering Conference*, Paper no. DEC2009-86809, 95-101.

Sato, T. and Qi, K. (1998), "Adaptive H infinite filter: Its application to structural identification", *J. Eng. Mech. - ASCE*, **124**(11), 1233-1240.

Sato, T. and Takei, K. (1998), "Development of a Kalman Filter with fading memory", *Proceedings of the Structural Safety and Reliability*, 387-394.

Sato, T. and Kaji, K. (2001)," Adaptive Monte Carlo Filter and structural identification", *Proceedings of the 1st International Conference on Monte Carlo Simulation*, Monte Carlo, Monaco.

Simon, J.J. and Uhlmann, J.K. (1997), "A new extension of the Kalman Filter to nonlinear systems", *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II, SPIE*.

Smyth, A.W. Marsi, S.F. Chassiakos, A.G. and Caughey, T.K. (1999), "On-line parametric identification of MDOF nonlinear hysteretic systems", *J. Eng. Mech.- ASCE,* **125**(2), 133-142.

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

Yoshida, I. and Sato, T. (2002), "Health monitoring algorithm by the Monte Carlo Filter based on non-Gaussian noise", *J. Natural Disaster Sci.*, **24**(2), 101-107.

Yun, C.B. and Shinozuka, M. (1980), "Identification of nonlinear structural dynamics systems", *J. Struct.Mech.*, **8**(2), 187-203.