Smart Structures and Systems, *Vol. 15, No. 3 (2015) 735-749*
DOI: http://dx.doi.org/10.12989/sss.2015.15.3.735

Posterior density estimation for structural parameters using improved differential evolution adaptive Metropolis algorithm

Jin Zhou*, Akira Mita[a] and Liu Mei[b]

Department of System Design Engineering, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan

(Received November 27, 2014, Revised January 20, 2015, Accepted January 28, 2015)

Abstract. The major difficulty of using Bayesian probabilistic inference for system identification is to obtain the posterior probability density of parameters conditioned by the measured response. The posterior density of structural parameters indicates how plausible each model is when considering the uncertainty of prediction errors. The Markov chain Monte Carlo (MCMC) method is a widespread medium for posterior inference but its convergence is often slow. The differential evolution adaptive Metropolis-Hasting (DREAM) algorithm boasts a population-based mechanism, which runs multiple different Markov chains simultaneously, and a global optimum exploration ability. This paper proposes an improved differential evolution adaptive Metropolis-Hasting algorithm (IDREAM) strategy to estimate the posterior density of structural parameters. The main benefit of IDREAM is its efficient MCMC simulation through its use of the adaptive Metropolis (AM) method with a mutation strategy for ensuring quick convergence and robust solutions. Its effectiveness was demonstrated in simulations on identifying the structural parameters with limited output data and noise polluted measurements.

Keywords: structural identification; differential evolution; adaptive metropolis-hastings; Markov chain Monte Carlo; structural parameter estimation; Bayesian posterior probability density

1. Introduction

Most of the structural identification approaches based on heuristic optimization algorithms are formulated as deterministic models in which the residual errors are built-up optimization functions to be minimized so that the candidate model can provide relatively accurate predictions of the system response. Heuristic algorithms, such as the genetic algorithm (GA) (Koh *et al*. 2007 and Perry *et al*. 2006), particle swarm optimization (PSO) (Xue *et al*. 2009), and differential evolution (DE) (Tang *et al*. 2008), have been used as global-optimum searches for the fitness function of the prediction errors. Recently, there has been an increasing need for structural identification to quantify the uncertain prediction errors associated with noise or model errors.

As is shown by Beck *et al*. (2010), no candidate model can exactly represent the I/O behavior of an actual system. It is a quixotic notion to choose only one biased model in a quest for the

*Corresponding author, Ph.D. Student, E-mail: zhoujin5120@gmail.com
[a] Professor, E-mail: mita@sd.keio.ac.jp
[b] Ph.D. Student, E-mail: meiliumail@gmail.com

http://www.techno-press.org/?journal=sss&subpage=8 ISSN: 1738-1584 (Print), 1738-1991 (Online)

output of the true system with uncertain prediction errors. Bayesian probabilistic inference provides a rigorous way of quantifying this uncertainty based on a probabilistic model that is defined by stochastic model classes. The model set, M, is a class of parameterized probability models, each of which predicts the behavior of the actual system with a prior probability density. The prior probability of each model indicates the initial plausibility of the individual model. In Bayesian parametric posterior density estimation, the identification problem is to infer the plausibility of each candidate model with a posterior density conditioned by the measured data; it is not a quest for the true structural parameters. It is usually difficult for Bayesian probabilistic inference identification to obtain the posterior probability density function (PDF) of the structural parameters, $P(\hat{x}(\theta)|Y^m)$, conditioned by the measured response, Y^m, where $\hat{x}(\theta)$ is a stochastic parameter vector defining each possible model within the model set (θ is a random variable in a probability space Ω). The posterior PDF describes how plausible each model is if one accounts for the uncertainty of the prediction errors. The posterior density, $P(\hat{x}(\theta)|Y^m)$, is needed to make robust predictions of the performance of the system based on past observations, as was illustrated by Papadimitriou *et al*. (2001) and Beck *et al*. (2002). Many studies have focused on obtaining the posterior PDF because its calculation often requires an evaluation of multidimensional integrals that cannot be easily obtained. In particular, Laplace's method of asymptotic approximation was utilized by Beck *et al*. (2002) to obtain a posterior PDF with a small-dimensional parameter space. To solve higher dimensional problems, an adaptive Markov chain Monte Carlo (MCMC) simulation method, the Metropolis-Hastings (MH) algorithm, was developed to be used in the Bayesian model update (Muto and Beck 2008). Since the advent of the MH algorithm, MCMC methods have become the primary means to obtain the posterior PDF in structural identification. Gibbs sampling and transitional Markov Chain Monte Carlo (TMCMC) were used by Ching *et al*. (2007). Cheung *et al*. (2009) used a hybrid Monte Carlo method, known as the Hamiltonian Markov chain method, to solve the higher dimensional Bayesian model updating problems.

However, all of these MCMC-based identification methods use a single Markov Chain, which may be inefficient and unreliable when the posterior surface is complicated. It is known that because of the noise corrupted system response, the surface of the prediction error lies in a hyper-surface of a multi-dimensional parametric space. The complicated surface of the prediction error will definitely cause the surface of the posterior model samples to have multiple regions of attraction and numerous local optima. To surmount this difficulty, the ability of heuristic algorithms to search for the global optimum will have to be merged with the advantage of the MH algorithm for inferring the posterior PDF. The first combining of DE and MCMC was proposed by Ter Braak. (2006) in Bayesian computing; it was named the DE-MC algorithm. Vrugt *et al*. (2009) proposed a differential evolution adaptive metropolis algorithm for solving the problem of flood forecasting.

In this paper, an improved differential evolution adaptive Metropolis-Hastings algorithm (IDREAM) is proposed for updating the posterior PDF of the structural identification model. Compared with other MCMC based identification methods, it runs different parallel Markov chains simultaneously and the posterior samples mutually exchange information along the iteration. Numerical examples of updating the posterior PDF of a linear structural system are presented, with which the effectiveness and efficiency of IDREAM are investigated. The influence of the incomplete measurements and noise errors on the posterior PDF of the parameters is discussed.

2. Problem statement

In the Bayesian probability logic identification framework, the measured response is used to estimate the posterior density of the plausibility for each I/O model in a model set instead of estimating biased parameters for only misrepresentative models (models with biased parameters that cannot represent the actual behavior of the true system). Here, let $y^m(t)$ denote the measured response at each time step ($t = 1, \dots, N_t$). The stochastic model set, M, is defined by a structural parameter vector, $\hat{x}(\theta) = \left(\hat{x}_1(\theta), \hat{x}_2(\theta), \cdots, \hat{x}_{n_m}(\theta)\right) \in \mathbb{R}^n$ ($\theta_{N_s} \in \Omega$, θ is a random variable in a probability space Ω), where n_m is the number of parameters for model $M_m \in M$, and N_s is the number of stochastic samples. The initial plausibility of each model parameterized by $\hat{x}(\theta_k)$ $(k = 1,2, \dots, N_s)$ is defined as a prior density function, $P(\hat{x}(\theta)|M)$. The updated plausibility of each I/O model considering the uncertainty of the measured response is defined as the posterior density, $P(\hat{x}(\theta)|y^m, M)$, which from Bayes' Theorem gives

$$P(\hat{x}(\theta)|y^m, M) = P(y^m|\hat{x}(\theta), M) \cdot P(\hat{x}(\theta)|M)/P(y^m|M) \tag{1}$$

where $P(y^m|\hat{x}(\theta), M)$ is obtained from a probabilistic model that accounts for the uncertainty of the prediction errors between the measured response and the stochastic output of the model set specified by each random model variable, $\hat{y}(\hat{x}(\theta), t)$. Let $e_j^k(t) = y_j^m(t) - \hat{y}_j^k(\hat{x}(\theta_k), t)$ denote the prediction error at each time interval ($t = 1, 2, \dots, N_t$, $k = 1,2, \dots, N_s$, and $j=1, \dots, m$, where m is the number of available measurements). The predictive PDF for the model output (white noise is considered as the measurement error; it thus obeys a normal distribution) at each time interval is

$$P\left(y_j^m(t)\middle|\hat{x}(\theta), M\right) = P\left(e_j(t)\middle|\hat{x}(\theta), M\right) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{\left[-\frac{1}{2\sigma_j{}^2}(y_j^m(t) - \hat{y}_j(\hat{x}(\theta),t))^2\right]}, \; j=1,\dots, m \tag{2}$$

Hence, the predictive PDF (which is the likelihood function) seen from the whole time history is

$$P(y^m|\hat{x}(\theta), M) = \frac{1}{(\prod_{j=1}^m \sqrt{2\pi}\sigma_j)^{N_t}} e^{\left[-\sum_{j=1}^m \frac{1}{2\sigma_j{}^2} \sum_{t=1}^{N_t} (y_j^m(t) - \hat{y}_j(\hat{x}(\theta),t))^2\right]} \tag{3}$$

The vector of the prediction error variance, σ_j^2, is an independent parameter corresponding to each candidate model in the model class, M. The term, $P(y^m|M)$, is called the evidence of the model class, and it equals

$$P(y^m|M) = \int P(y^m|\hat{x}(\theta), M) \cdot P(\hat{x}(\theta)|M) d\theta \tag{4}$$

The difficulty in estimating the Bayesian posterior density is none other than approximating this multi-modal and high-dimensional integral. A direct and easy-to-understand approach would be the Monte-Carlo method. However, a direct MC simulation usually requires a large number of MC samples, which makes it inefficient. On the other hand, the Metropolis-Hastings (MH) algorithm (Chib *et al*. 1995) is a new Markov-chain Monte-Carlo simulation tool, and it can meet this challenge. In the MH algorithm, it is unnecessary to accurately calculate the model evidence in Eq. (4). The Markov chain samples are initially distributed according to the prior PDF and updated with a jumping distribution. The samples in the Markov chain are selected according to the MH acceptance rate and eventually converge to a stationary probability density which can be seen to be the desired posterior PDF of the model parameters. However, the scale of the jumping distribution

is hard to decide for the MH algorithm. This is an important problem because choosing a suitable transition strategy for the samples in the Markov chain influences not only the convergence speed but also the robustness of the solutions.

3. IDREAM for Bayesian inference of parameter estimation

This study presents an improved differential evolution adaptive Metropolis-Hastings algorithm (IDREAM) aimed at overcoming the difficulty of choosing a suitable jumping scale in the algorithm. The IDREAM algorithm starts by choosing samples represented as a dimensional vector $\hat{x}$, and the k^{h} sample is denoted as $\hat{x}^k$ $(k = 1,2, \ldots N_s)$. The number of samples is twice the parametric dimension, as is proposed in J. Vrugt *et al*. (2009). The initial states of the Markov chain samples are drawn from the search domain by using Latin hypercube sampling (LHS). The density function of each sample in the initial state can be computed as a prior density $p(\hat{x}_0^k)$ for $k = 1,2, \ldots N_s$.

3.1 DE mutation of the MC samples

In the differential evolution adaptive Metropolis-Hastings (DREAM) algorithm (Vrugt *et al.* 2009), the samples are updated by using the difference between randomly chosen pairs of samples in the current state. Let $\Delta_{\hat{x}}^k = \hat{x}_{s+1}^k - \hat{x}_s^k$ denote the jump scale between the updating state (s+1) and current state (s) of the sequence k in the Markov chain. In the DREAM algorithm, the samples are updated as

$$\Delta_{\hat{x}}^k = (u_d + e_d)\gamma(\delta, d)\left[\sum_{j=1}^{\delta} \hat{x}_s^{r_1(j)} - \sum_{n=1}^{\delta} \hat{x}_s^{r_2(n)}\right] + \varepsilon \quad (5)$$

where ε is a small random vector that is drawn from $N_d(0, \hat{\Sigma})$. This variable is the same as the jumping scale vector in the classic MH algorithm, which is called the random-walk strategy. In the MH algorithm, the jump scale ε obeys a Gaussian distribution in which the variance, $\hat{\Sigma}$, decides the jumping direction and scale of the MC samples. It is clear that the efficiency of this algorithm is strongly affected by ε. So how we choose an appropriate jumping scale for the samples transition is a difficulty especially for high-dimensional problems. This problem can be solved with Eq. (5), where the jumping scale equals an adaptive multiple of the difference between pairs of randomly chosen samples in the current state. δ is the number of chosen pairs, and $r_1(j)$ and $r_2(n)$ are respectively different and random integers that are chosen from the integer set {1,2,…k-1,k+1,…,N_s}. The term u_d us the d-dimensional unit vector, and e_d signifies a small random vector drawn from a uniform distribution to assure the ergodicity of each individual Markov chain. The scaling factor γ is decided by the values of δ and d, where d is the parametric dimension. From Eq. (5), it is clear that in DREAM, only the sequences in the current state are used to update the samples in the Markov chain. In this study, we propose a new update pattern in which the sequence having the largest plausible density in the current state k and the one with the maximum posterior PDF from the initial state to the current state are used for the updating of the Markov chain samples

$$\Delta_{\hat{x}}^k = w_1^k\left(\hat{x}_s^{cbest} - \hat{x}_s^k\right) + w_2^k\left(\hat{x}_{1-s}^{gbest} - \hat{x}_s^k\right) + (u_d + e_d)\gamma(\delta, d)\sum_{j=1}^{\delta} w^{*k}{}_j\left(\hat{x}_s^{r_1(j)} - \hat{x}_s^{r_2(j)}\right) + \varepsilon \quad (6)$$

where $\hat{x}_s^{cbest}$ denotes the sample with the maximum density in the current state (s^{th} state) and is called the "current-best individual", and $\hat{x}_{1-s}^{gbest}$ denotes the sequence with the maximum density in the previous states (from the initial state to the s^{th} state) and is called the "global-best individual". $\hat{x}_s^{r_1(j)}$ and $\hat{x}_s^{r_2(j)}$ are randomly chosen individuals in the current state and are different from each other. The weighting factors, w_1^k, w_2^k and w_j^k, are the distance between the individuals

$$\begin{cases} w_1^k = d_1^k / d_{sum}^k \, , w_2^k = d_2^k / d_{sum}^k , w^{*k}_j = d_j^k / d_{sum}^k \\ d_{sum}^k = d_1^k + d_2^k + \sum_{j=1}^{\delta} d_j^k \end{cases} \tag{7}$$

where d_1^k and d_2^k are respectively the Euclidean distance from sequence $\hat{x}_s^k$ to the "current-best" sample $\hat{x}_s^{cbest}$ and the distance from the sequence $\hat{x}_s^k$ to the "global-best" sample $\hat{x}_s^{gbest}$, and d_s^k is the Euclidean distance of are randomly chosen sample pairs excluding the individual $\hat{x}_s^k$ in the Markov chain. From Eq. (6), one can see that the biggest difference of IDREAM from DREAM is that the posterior samples are updated using both the maximum PDF in the current state and that of the previous states, while the updated samples of DREAM are orientated between the chosen pairs of samples only in the current state. IDREAM enhances the convergence speed especially in the early stage because the difference between the individual and the "best" sample is definitely large owing to the diversity of samples in the initial state. From Eq. (7), it is clear that samples farther away from the "best" sample possess a larger jumping scale because the weight factor is positively propositional to the distance. As for the "best" sample, the update method is the same as in DREAM, i.e., Eq. (5). The convergence becomes slower in the later stages when the diversity of the samples in the Markov chain decreases. Because of this, a crossover strategy (Storn *et al.* 1997) is used to keep the diversity of the MC samples high.

Assertion: IDREAM yields a Markov chain which is irreducible and aperiodic with a unique stationary distribution with the target *pdf* $\pi(\cdot)^{N_s}$. *Proof*: the proof consists of two parts:

1. If the sample of the k^{th} chain coincides with the position of the global-best and the current-best individual, the probability of a jump from sample in the state s of the chain to the sample in the state s+1 is the same as in the DREAM algorithm, in which the conditional probability of jumping forward, $\mathrm{p}\left(\hat{\mathrm{x}}_{\mathrm{s}}^{\mathrm{k}} \rightarrow \hat{\mathrm{x}}_{\mathrm{s}+1}^{\mathrm{k}}\right)$, is equal to that of jumping back, $\mathrm{p}\left(\hat{\mathrm{x}}_{\mathrm{s}+1}^{\mathrm{k}} \rightarrow \hat{\mathrm{x}}_{\mathrm{s}}^{\mathrm{k}}\right)$. For the individuals who are different from the "best" sample in the current state, *s*, the distance and the corresponding weighting factor decrease quickly in the later stages (especially following a sufficient burn-in period) as the populations converge. The deviations (random walks) of the randomly chosen sample pairs mainly decide the transitions of each chain. The $N_s - 1$ samples can been seen as updated conditionally on the other chains obeying the reversibility property of the Markov chain, because the Jacobian of the transformation (Hastie 2012) implied in Eq. (6) is close to 1 in absolute value, (Vrugt *et al.* 2009), and the first two items of Eq. (6) play limited roles in the transition Markov chains during the later stages (*Note*: $\hat{x}_s^{cbest}$ is equal to $\hat{x}_{1-s}^{gbest}$ in the later stages). 2. Detailed balance is achieved with an accepting rule having probability $min(p(\hat{x}_{s+1})/p(\hat{x}_s), 1)$. Each chain maintains conditional detailed balance, because the chains are aperiodic and not transient with the random walk generated by Eq. (6). The N_s chains are irreducible with the unbounded support of the distribution of ε in Eq. (6). This concludes the assertion.

3.2 DE crossover strategy for the MC samples

The crossover strategy is implemented in d dimensions of the current samples, $\hat{x}_s^k$, and the updated posterior samples, $\hat{x}_{s+1}^k$, and a trial sample $\hat{x}_{s+1}^{kj}$ is generated with

$$\hat{x}_{s+1}^{kj}=\begin{cases}\hat{x}_s^{kj} & if\ U_j \le 1-CR\\ \hat{x}_{s+1}^{kj} & otherwise\end{cases} \tag{8}$$

where j=1,2,…,d; U_j is the j^{th} independent random number uniformly distributed in the range of [0,1]. *CR* is a crossover probability defined by the user.

3.3 Metropolis acceptance

The density of the new sample, $p\left(\hat{x}_{s+1}^k\right)$, and the Metropolis acceptance (Haario *et al.* 2001) can be calculated with the updated samples in the Markov chain

$$\alpha(\hat{x}_s,\hat{x}_{s+1}) = min\left[\frac{p(\hat{x}_{s+1})}{p(\hat{x}_s)},1\right] \tag{9}$$

The algorithm accepts the candidate state $\hat{x}_{s+1}$ with probability min(1,$\alpha(\hat{x}_{s+1},\hat{x}_s)$), and keeps the current state $\hat{x}_s$ with probability 1-min(1, $\alpha(\hat{x}_{s+1},\hat{x}_s)$). This process is repeated in several iterations, and after a burn-in period, the chain of samples approaches a stationary distribution. The desired posterior PDF can be obtained from these Markov-chain samples, excluding the ones in the burn-in period.

3.4 Gelman-Rubin convergence condition

The IDREAM algorithm ends by checking the Gelman-Rubin convergence condition (A. Gelman *et al.* 1992), and calculates the $\widehat{\mathrm{R}}_j$-statistic by using the last 50% of the samples in each chain. Let k be the number of the sequences used to calculate $\widehat{\mathrm{R}}_j$, and let B be the variance between the sequence means and W signify the average of the within sequence variances (B=$N_s \times \sum_{i=1}^{k}(\hat{x}_i-\bar{\hat{x}})^2/(\mathrm{k}-1)$, and W=$\sum_{i=1}^{k} s_i^2/k$, where s_i^2 is the variance of the sequence). Furthermore, the target posterior variance is estimated as $\hat{\sigma}^2$=$\frac{n-1}{n}W+\frac{1}{n}B$. The Gelman-Rubin convergence parameter, $\widehat{\mathrm{R}}$, can be calculated as

$$\widehat{\mathrm{R}}=\sqrt{\frac{k+1}{k}\times\frac{\hat{\sigma}^2}{W}-\frac{N_s-1}{k\times N_s}} \tag{10}$$

If $\widehat{\mathrm{R}}_j < 1.2$ for all dimensions (Gelman *et al.* 1992), it means that the Markov chain has converged to a stationary distribution; otherwise, the posterior sample is updated with Eq. (6).

3.5 IDREAM based parametric posterior density estimation

The procedure of IDREAM-based Bayesian probability inference parameter estimation is as follows:

Step 1: The initial N_s sequences for the Markov chain are drawn by Latin hypercube sampling with the predefined maximum and minimum boundary of the structural parameters and number of

samples, N_s, while respecting the prescribed limits of the search space. Calculate the prior density and the likelihood function using Eqs. (2) and (3).

Step 2: Update the posterior sample of the Markov chain by using Eqs. (6) and (7).

Step 3: The samples are updated according to the crossover probability calculated with Eq. (8). Calculate the density for the each updated sample in the Markov chain.

Step 4: Use the Metropolis acceptance (Eq. (9)) to decide whether to accept the new samples.

Step 5: Return to Step 2 to Step 4, considering the burn-in period and calculate the Gelman statistic condition using Eq. (10) for each dimension of the stochastic parameter. If the stopping criteria are met, ($\widehat{\mathrm{R}}_j < 1.2$ or the max. iteration number is reached) stop the algorithm; otherwise, return to Step 2.

4. Numerical study

A numerical simulation of a 5-DOF linear time invariant (LTI) system was carried out to verify the IDREAM algorithm. The actual system output was simulated from a linear structural system (a one-dimensional shear frame structure). For sake of clearly exhibiting that only the measurement uncertainty is considered, the second-order differential dynamic equation of the structural system is described by the state-space representation as

$$\begin{bmatrix}\dot{v}_1(t)\\ \dot{v}_2(t)\end{bmatrix} = \begin{bmatrix}0 & \mathrm{I}\\ -M^{-1}K & -M^{-1}C\end{bmatrix}\begin{bmatrix}v_1(t)\\ v_2(t)\end{bmatrix} + \begin{bmatrix}0\\ -\mathrm{I}\end{bmatrix}\Gamma^T u(t) \tag{11}$$

where *M*, *C*, and *K* are mass, damping, and stiffness matrices, I is a $n \times n$ identity matrix, $\Gamma = [1,1,\ldots,1]^T$ is a $1 \times n$ position vector. $v_1(t)$ and $v_2(t)$ are state space vectors respectively representing the displacement and velocity, and $u(t)$ is the input of the system. Equation (11) includes a Rayleigh damping matrix *C*, where the modal damping ratio (ζ_r) is set to 5% in the first two modes (*r*=1, 2) (Mita 2003).

$$C = \alpha M + \beta K, \quad \zeta_r = \frac{\alpha}{2\omega_r} + \frac{\beta\omega_r}{2} \tag{12}$$

The system output is an acceleration value that is assumed to be contaminated by Gaussian white noise w_j(t)~N(0, $\sigma_j(\theta)$), (*j*=1,.. m). The measured output vector is thus

$$y(t) = [-M^{-1}K \quad -M^{-1}C]\begin{bmatrix}v_1(t)\\ v_2(t)\end{bmatrix} - \Gamma^T \cdot u(t) + w(\sigma(\theta), \mathrm{t}) \tag{13}$$

The input was the 1940 El Centro ground motion (N-S acceleration measured at the Imperial Valley Irrigation District substation in El Centro, CA, during the 1940 Imperial Valley earthquake) lasting 40 s. It was normalized so that its peak was 10 cm/s^2, and the sampling frequency was decimated as 50 Hz (Fig. 1). The Newmark-beta method was used to calculate the structural response. An output acceleration (*acc.*) with different Gaussian white noise levels (Eq. (13)) was assumed. To show the effectiveness of the method for an identification problem with a large noise level, the noise level (*nl.*) was chosen to have a standard deviation of 30% or 100% of the corresponding signal; i.e., if $\sigma_{acc.,i}$ is the standard deviation of the i^{th} floor acceleration (relative to the ground), then the noise affecting the measurement of that floor's acceleration has an RMS $\sigma_i = nl. \times \sigma_{acc.,i}$. The response of the 5$^{\text{th}}$ DOF with and without 100% noise is shown in Fig. 1.

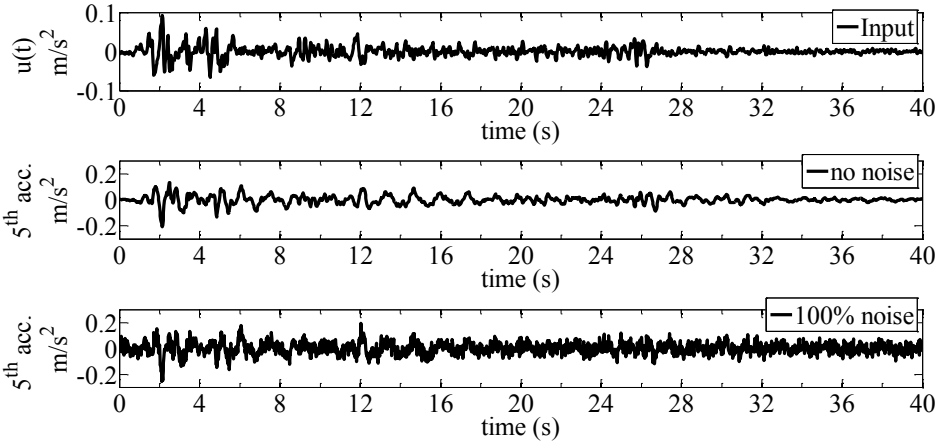


Fig. 1 Input and measured noise

The influence of the limited availability of measurements was also assessed. In the "full output" scenario, measurements of all DOFs were available, whereas in the "partial output" scenario, only data from floors 3 and 5 were available. The mass was assumed to be known and deterministic; hence, an *n*-DOF system with *m*-available measurements can be described by a model set, of which the stochastic parameterized vector for each model is

$$\hat{\boldsymbol{x}}(\theta)=\{k_1(\theta),\dots,k_n(\theta),\zeta_1(\theta),\zeta_2(\theta),\sigma_1(\theta),\dots,\sigma_m(\theta)\} \tag{14}$$

where k_i denotes the stiffness of i^{th} -DOF, n is the number of structural DOFs; ζ_1 and ζ_2 are the damping ratio of the first two modes; σ_i denotes the standard deviation of the i^{th} measurement noise, m is the number of measurements; θ denotes the random variables in the sample space, Ω_θ.

Table 1 lists the structural properties together with the simulated measurement error for each available DOF. The parametric dimension was 12 for the full output scenario and 9 for the partial output scenario. The parameters of the IDREAM algorithm were set as follows: the number of Markov chain samples (N_s) was 20, the crossover probability (CR) was 0.85, and the number of sample pairs (δ) was 5. These prescribed parameters were tested and also suggested by the developer of the algorithm. If the number of samples is increased, such as to 50 or 100, the results may improve a little at the expense of a higher computing cost. The search domain was 0.5 to 2.0 times the true value, which depends on prior knowledge about the structural system. The prior distribution followed a uniform distribution at the search boundary.

The results obtained by the original DREAM algorithm are shown for comparison in the full output scenario (Tables 2(a) and 2(b)) and partial output scenario (Tables 3(a) and 3(b)). From Tables 2 (a) and 2(b), it is clear that both algorithms performed very well in the noise-free scenario. As the magnitude of the measured noise increased, the relative errors of the identified solutions remained small. The maximum relative error for the most plausible value of structural stiffness, i.e., the sample with the maximum posterior density ($\hat{x}_{MAP}$), ranged from zero in the no-noise case to 0.502% in the 30% noise case and up to 1.987% in the 100% noise case when using the DREAM algorithm.

Table 1 Structural properties and measurement errors

Stiffness	(10^4N/m)		(σ_i) m/s^2	30% noise	100% noise
Levels 1	2.000	Case 1	Levels 1	0.0049	0.0162
Level 2-5	1.000		Levels 2	0.0074	0.0248
Mass	(kg)		Levels 3	0.0076	0.0252
Levels 1-4	50		Levels 4	0.0077	0.0256
Level 5	45		Levels 5	0.0095	0.0317
Damping ratio		Case 2			
ζ_1	0.05		Levels 3	0.0076	0.0252
ζ_2	0.05		Levels 5	0.0095	0.0317

Table 2 (a) Identified results of structural parameters (full output scenario)

		no noise		30% noise		100% noise	
		DREAM	IDREAM	DREAM	IDREAM	DREAM	IDREAM
k_1	*error*	0.000	0.000	0.352	0.164	0.656	0.250
	cov.	0.000	0.000	0.906	0.533	3.263	1.641
k_2	*error*	0.000	0.000	0.324	0.242	1.558	0.114
	cov.	0.000	0.000	0.683	0.399	2.462	1.286
k_3	*error*	0.000	0.000	0.298	0.353	1.987	0.198
	cov.	0.000	0.000	0.812	0.459	2.946	1.474
k_4	*error*	0.000	0.000	0.172	0.045	0.658	0.717
	cov.	0.000	0.000	0.813	0.465	2.789	1.564
k_5	*error*	0.000	0.000	0.502	0.299	1.155	1.083
	cov.	0.000	0.000	0.937	0.530	3.227	1.835
ζ_1	*error*	0.000	0.000	1.307	0.813	3.442	2.035
	cov.	0.000	0.000	1.591	0.871	5.545	2.946
ζ_2	*error*	0.000	0.000	0.887	1.217	1.677	0.499
	cov.	0.000	0.000	1.158	0.635	3.951	2.095

* the error is in %; the cov. (the ratio of the standard deviation to the mean) is in %

Table 2 (a) shows that the accuracy of $\hat{x}_{MAP}$ when using IDREAM was better than that of DREAM. The maximum relative error fell to 0.353% in the 30% noise case and 1.083% in the 100% noise case. The parameter with the largest uncertainty obtained by the two algorithms was in the damping ratio. The largest coefficient of variance (*cov.*) of the damping ratio was 1.591% in the 30% noise case, and it increased to 5.545% in the 100% noise case for the DREAM algorithm. For IDREAM, the coefficient of variance (*cov.*) of the damping ratio ranged from 0.871% to 2.946%. It is clear that the parametric uncertainty was additive as the measurement error increased. Table 2 (b) shows that the two algorithms can identify the exact value of the noise standard

deviation, σ_i in Table 1. In the noise-free scenario, both solutions were as small as zero. For the 100% noise case, the maximum error emerged in the third DOF at 2.341% for DREAM and at 2.562% for IDREAM. The identified results for the partial output case are listed in Tables 3(a) and 3(b).

From Table 3(a), it can be seen that when there was a noise error, the maximum relative errors of $\hat{x}_{MAP}$ were smaller for IDREAM than for DREAM. In the 30% noise case, the maximum relative error for DREAM was 1.855%, but only 1.501% for IDREAM, while in the 100% noise case, the maximum error decreased from 5.602% to 5.214%.

Table 2 (b) Identified results of prediction errors (full output scenario)

		no noise		30% noise		100% noise	
		DREAM	IDREAM	DREAM	IDREAM	DREAM	IDREAM
σ_1	*error*	0.000	0.000	0.967	0.080	2.060	0.204
	cov.	0.000	0.000	0.136	0.082	0.172	0.055
σ_2	*error*	0.000	0.000	0.966	0.034	0.725	2.562
	cov.	0.000	0.000	0.074	0.042	0.183	0.043
σ_3	*error*	0.000	0.000	0.445	0.731	2.341	0.678
	cov.	0.000	0.000	0.088	0.051	0.084	0.032
σ_4	*error*	0.000	0.000	0.185	1.879	1.302	1.128
	cov.	0.000	0.000	0.075	0.049	0.070	0.042
σ_5	*error*	0.000	0.000	1.108	1.741	2.237	0.128
	cov.	0.000	0.000	0.096	0.022	0.111	0.048

* the error is in %; the cov. (the ratio of the standard deviation to the mean) is in %

Table 3 (a) Identified results of structural parameters (partial output scenario)

		no noise		30% noise		100% noise	
		DREAM	IDREAM	DREAM	IDREAM	DREAM	IDREAM
k_1	*error*	0.019	0.010	1.404	1.501	2.227	2.546
	cov.	0.000	0.000	4.267	2.606	16.58	7.761
k_2	*error*	0.082	0.025	1.025	0.794	4.372	1.450
	cov.	0.000	0.000	2.216	1.351	10.33	4.584
k_3	*error*	0.072	0.018	1.855	0.196	5.602	0.725
	cov.	0.000	0.000	2.710	1.530	12.69	5.538
k_4	*error*	0.089	0.014	1.041	0.511	2.722	2.376
	cov.	0.000	0.000	2.145	1.222	8.633	4.224
k_5	*error*	0.106	0.028	1.233	1.189	0.859	4.485
	cov.	0.000	0.000	2.243	1.358	12.32	4.171
ζ_1	*error*	0.050	0.030	1.066	0.406	1.667	5.214
	cov.	0.000	0.000	2.110	1.179	6.774	4.024
ζ_2	*error*	0.016	0.006	1.564	1.067	0.273	0.972
	cov.	0.000	0.000	2.142	1.258	7.930	4.020

* the error is in %; the cov. (the ratio of standard deviation to the mean) is in %

Table 3(b) Identified results of prediction errors (partial output scenario)

		no noise		30% noise		100% noise	
		DREAM	IDREAM	DREAM	IDREAM	DREAM	IDREAM
σ_3	*error*	0.000	0.000	0.950	2.406	2.675	0.669
	cov.	0.000	0.000	0.145	0.114	0.207	0.089
σ_5	*error*	0.000	0.000	1.240	0.982	0.490	0.557
	cov.	0.000	0.000	0.096	0.094	0.116	0.091

* the error is in %; the cov. (the ratio of standard deviation to the mean) is in %

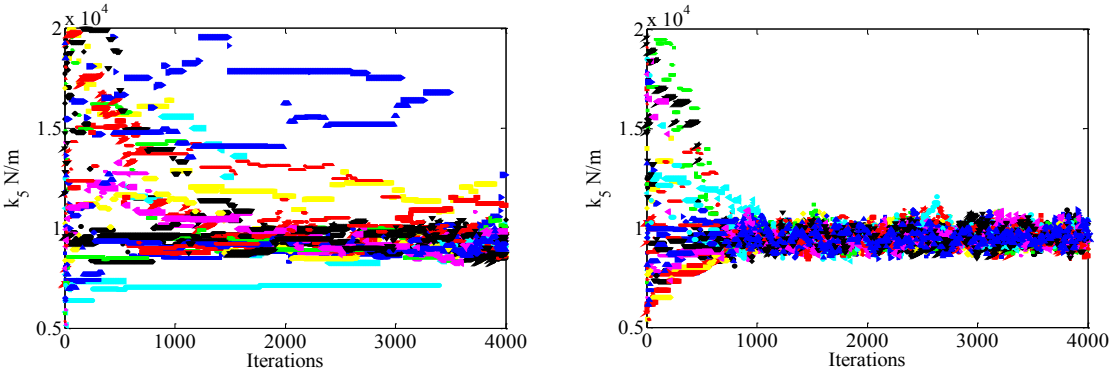


Fig. 2 Progress of identification of stiffness of 5th floor (partial output, 100% noise)

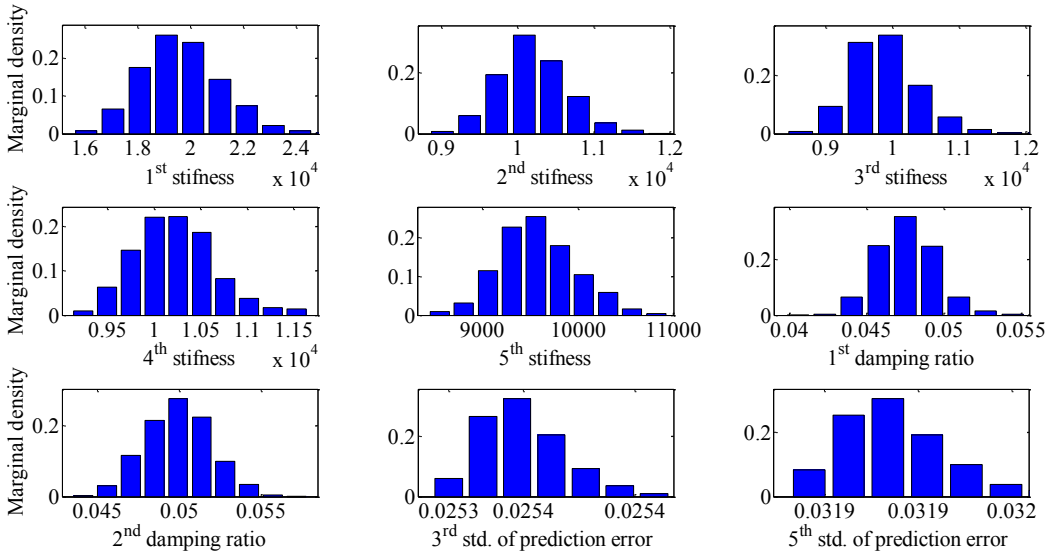


Fig. 3 Histogram of marginal density for each parameter (100% noise, partial output)

Comparing Table 3(a) with Table 2(a), it is clear that the loss of available measurements leads to an increase in parametric uncertainty, because the maximum coefficient of variance for the structural parameters adds from 2.946% to 7.761% at the same noise level. Table 3(b) shows that the actual prediction error standard deviation can be well estimated even if only measurements of the 3rd and 5th DOFs are available. The maximum error of the estimated prediction errors, $\hat{x}_{MAP}$, was 2.406% in the 30% noise case and 0.669% in the 100% noise case. For the DREAM algorithm, the corresponding errors were 1.240% in the 30% noise case and 2.675% in the 100% noise case. (Note that the maximum error of the estimator prediction errors seems larger for the 30% noise scenario than for the 100% noise scenario. The reason for this phenomenon is the denominator for calculating the relative error in the case of 30% noise is smaller than that in the case of 100% noise.)

Fig. 2 shows the progress of identification of the stiffness of the 5th DOF at the 100% noise level in the partial output scenario. In Fig. 2, each plot with a different color denotes the progress of a Markov Chain, which means there were 20 posterior MC samples at each iteration. We can see that the posterior samples of the Markov chain obtained by IDREAM were more stable than those of DREAM, which leads to a smaller uncertain range. The marginal posterior density of the parameters using IDREAM can be obtained by using kernel density estimation on the stationary Markov chain samples excluding the sequence during the burn-in period, as is shown in Fig. 3.

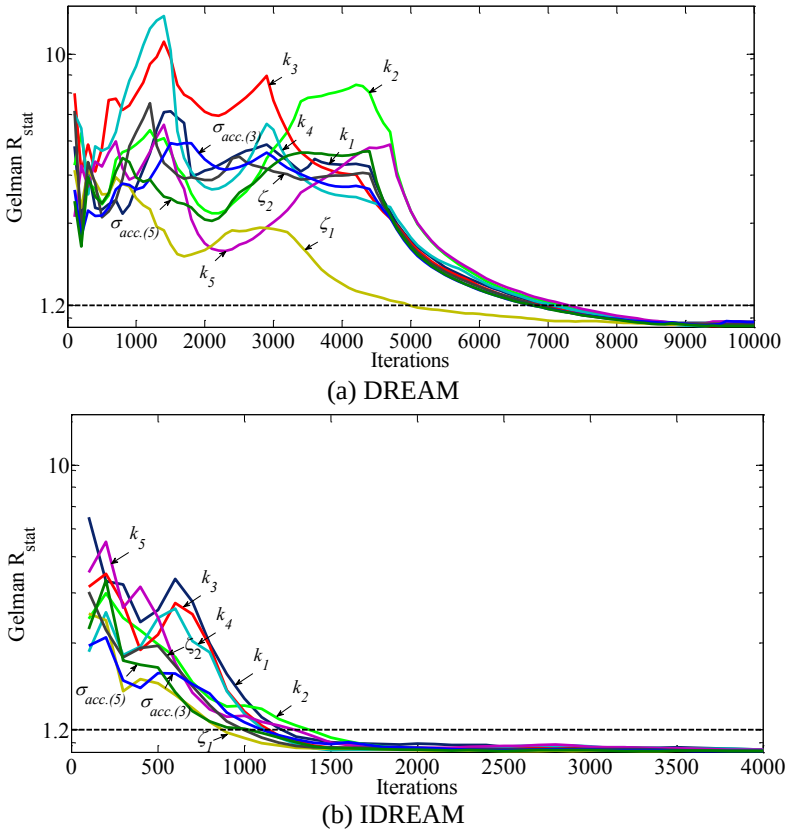
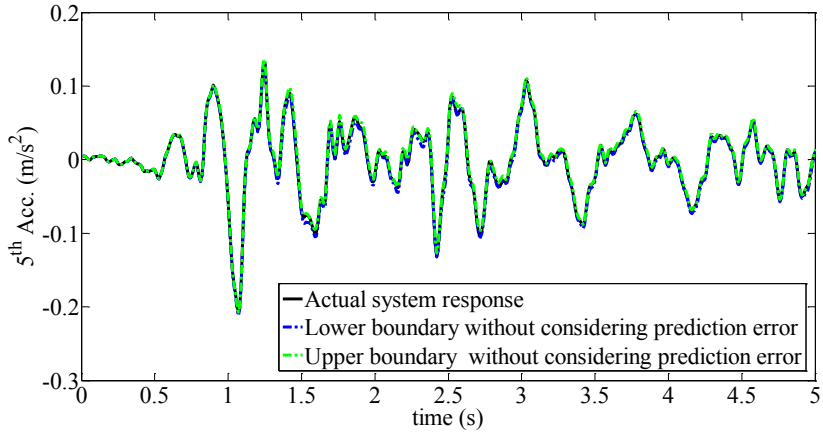
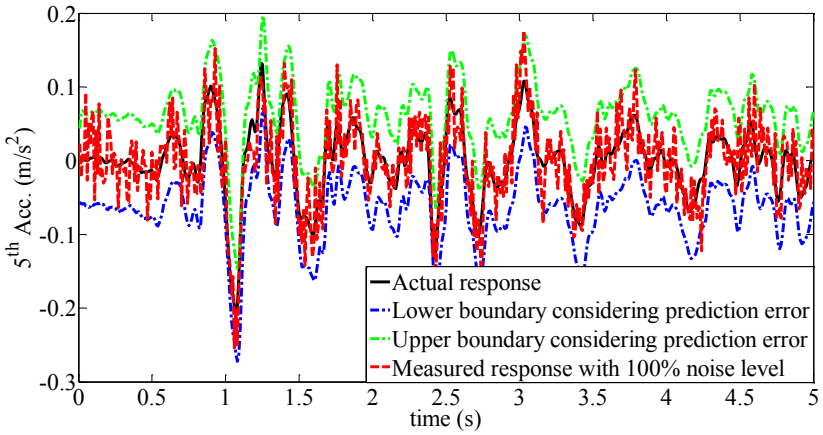


Fig. 4 Convergence of Markov Chain (100% noise, partial output)



(a) Response range considering the identified parametric uncertainty



(b) Response range considering the identified std. of prediction error

Fig. 5 95% uncertainty ranges for acceleration of 5th DOF (100% noise, partial output)

Figs. 4(a) and 4(b) indicate that the Markov chain converged for each identified parameter when using DREAM and IDREAM in the scenario of partial outputs and 100% noise. Comparing Fig. 4(a) with Fig. 4(b), however, makes it clear that IDREAM already converged by the time 4000 iterations were reached, but the Markov chains of DREAM were still unstable. Therefore, IDREAM converged faster than DREAM. Combing the solutions in Tables 2 and 3, we can conclude that IDREAM outperformed DREAM because of its earlier convergence and robustness of the posterior samples without decreasing the accuracy of the results.

The posterior uncertain range that assures a reliability of 95% can be obtained from the posterior samples of the model class which denotes the plausibility of each I/O system. Fig. 5 shows the ranges for part of the time history (5 seconds). Fig. 5 (a) is the uncertain response range of a stochastic I/O system parameterized by identified parameters of the posterior candidate model set with 95% assurance and without prediction errors at each time interval. On the other hand, Fig. 5(b) shows the uncertain range of the response with 95% assurance and with measurement errors at each time interval by incorporating the identified standard deviation of the prediction error. Fig.

5(a) illustrates the effectiveness of the Bayesian identification method because the parametric uncertain response range is close to the actual system response. Moreover, in Fig. 5(b), the percentage of the response considering a 100% measurement error within the uncertain range that covers the prediction error is 94.75%.

5. Conclusions

A framework of Bayesian probability inference for identification based on an improved differential evolution adaptive Metropolis-Hasting (IDREAM) algorithm was proposed. Compared with the DREAM algorithm, its novelty lies in a new sample updating pattern that speeds convergence and improves the stability of the posterior samples. IDREAM runs MCMC simulations in parallel and keeps the diversity of samples by using a DE crossover strategy. This gives it a strong ability to search for the global optimum and to resolve the problem that the MH algorithm has in choosing an approximate jump scale. A numerical simulation of a 5-DOF system demonstrated its potential for solving identification problems with a high noise level and with partial output data. In conclusion, IDREAM is a new approach to obtaining the posterior density of a model class that cannot be easily found with the classic Monte-Carlo method due to the difficulty in calculating high-dimensional integrals.

Acknowledgments

This work was supported in part by a grant for the project “Implementation of shelter guidance system for commuters who are unable to return home based on Structural Health Monitoring of tall buildings after large-scale earthquake,” (FY2013-2016, PI: Mita) from the Japan Science and Technology Agency.

References

Beck, J.L. (2010), “Bayesian system identification based on probability logic”, *J. Struct. Control Health Monit.*, **17**(7), 825-847.

Beck, J.L. and Au, S.K. (2002), “Bayesian updating of structural models and reliability using Markov-Chain Monte-Carlo simulation”, *J. Eng. Mech. - ASCE*, **128**(4), 380-391.

Beck, J.L. and Yuen, K.V. (2004), “Model selection using response measurements: A Bayesian probabilistic approach”, *J. Eng.Mech. -ASCE*, **130**(2), 192-203.

Cheung, S.H. and Beck, J.L. (2009), “Bayesian model updating using Hybrid Monte Carlo simulation with application to structural dynamic models with many uncertain parameters”, *J. Eng. Mech. - ASCE*, **135**(4), 243-255.

Chib, S. and Greenberg, E. (1995), “Understanding the Metropolis-Hasting algorithm”, *Am. Statist. Association*, **49**(4), 327-335.

Ching, J. and Chen, Y.C. (2007), “Transitional Markov Chain Monte Carlo method for Bayesian model updating, model class selection and model averaging”, *J. Eng.Mech. - ASCE*, **133**(7), 816-832.

Gelman, A. and Rubin, D.B. (1992), “Inference from iterative simulation using multiple sequence”, *Statistical Sci.*, **7**(4), 457-472.

Haario, H., Saksman, E. and Tamminen, J. (2001), “An adaptive Metropolis algorithm”, *J. Bernouli*, **7** (2),

223-242.
Hastie, D.I. (2012), "Model choice using reversible jump Markov chain Monte Carlo", *Statistica Neerlandica*, **66**(3), 309-338.
Koh, C.G and Perry, M.J. (2007), "Structural damage quantification by system identification", *J. Earthq. Tsunami*, **1**(3), 211-231.
Mita, A. (2003), *Structural Dynamics for Health Monitoring*, Sankeisha Co., Ltd, Nagoya , Japan.
Muto, M. and Beck, J.L. (2008), "Bayesian updating and model class selection for hysteretic structural models using stochastic simulation", *J. Vib. Control*, **14**(1-2), 7-34.
Papadimitriou,C., Beck, J.L and Katafygiotis, L.S. (2001), "Updating robust reliability using structural test data", *Probabilist. Eng. Mech.*, **16** (2), 103-113.
Perry, M.J., Koh, C.G. and Choo, Y.S. (2006), "Modified genetic algorithm strategy for structural identification", *Comput. Struct.*, **84**(8-9), 529-540.
Storn, R. and Price, K. (1997), "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces", *J. Global Optim.*, **11**(4), 341-359.
Tang, H.S., Xue, S.T. and Fan, C.X. (2008), "Differential evolution strategy for structural system identification", *Comput. Struct.*, **86**(21-22), 2004-2012.
Ter Braak, C.J.F. (2006), "A Markov Chain Monte Carlo version of the genetic algorithm differential evolution: easy Bayesian computing for real parameter spaces", *Statistics Comput.*, **16**(3), 239-249.
Vrugt, J.A., Ter Braak, C.J.F., Diks, C.G.H. et al. (2009), "Acceleration Markov Chain MonteCarlo simulation by differential evolution with self-adaptive randomized subspace sampling", *Int. J. Nonlinear Sci. Numerical Simulation*, **10**(3), 273-290.
Xue, S., Tang, H. and Zhou, J. (2009), "Identification of structural systems using particle swarm optimization", *J. Asian Architect. Build. Eng.*, **8**(2), 101-112.