Iranian Journal of Science & Technology, Transaction B, Engineering, Vol. 29, No. B6
Printed in The Islamic Republic of Iran, 2005

FINITE ELEMENT MODEL UPDATING OF ROTATING STRUCTURES USING DIFFERENT OPTIMISATION TECHNIQUES*

S. ZIAEI-RAD

Dept. of Mechanical Engineering, Isfahan University of Technology, Isfahan, I. R. of Iran
Email:szrad@cc.iut.ac.ir

Abstract– This paper is concerned with finite element model updating of rotating structures using measured vibration test data. The use of both deterministic and stochastic optimisation techniques was investigated in order to minimise the difference between the measured and analytical data. First, a theoretical basis was developed for frequency response functions (FRFs) updating techniques. The standard linear least-square (LLS) formulation was applied to the FRF updating formulation where the element mass, damping, gyroscopic and stiffness matrices are corrected by using a single multiplier, the so-called *p*-value. A new residue was then proposed and formulated to improve the convergence rate of the FRF-based model updating in the presence of noise. Next, two well-known stochastic optimisation methods that require no gradient and can achieve a global optimal solution in solving non-smooth and highly non-linear optimisation problems, namely genetic algorithm (GA) and adaptive simulated annealing (ASA), were introduced and implemented to the developed rotating structure code. The findings were illustrated in the case of a test rotor and the advantages and disadvantages of the proposed techniques were discussed in some detail.

Keywords– Model updating, modal analysis, optimization techniques

1. INTRODUCTION

Model updating can be defined as the adjustment of an existing analytical model in light of a measured vibration test. After adjustment, the updated model is expected to represent the dynamic behaviour of the structure more accurately.

Over the last twenty years, a significant number of model updating techniques have been proposed which have become important tools for correcting models of vibrating structures [1-3]. However, no reliable and generally applicable procedures have been formulated so far.

Updating models of rotating structures did not receive the same attention as non-rotating structures. Models for non-rotating structures are usually self-adjoint and thus possess certain symmetries which considerably simplify their analysis and the experimental extraction of dynamic models. However, high-speed rotating structures exhibit gyroscopic forces which create non-self-adjoint (non-symmetric) matrices. A non self-adjoint system is represented by natural frequencies and both right and left-hand eigenvectors. This additional set of vectors makes experimental extraction of a model difficult, or even impossible [4-6]. As an immediate consequence of this different form of modal model, conventional model updating procedures can not be applied directly without taking into account the basic differences in the model structure. Therefore, an FRF based model updating seems to be more suitable for rotating machinery as it uses the data directly from measurements.

In this study, in order to do model updating, minimization between measured and analytical FRFs of rotating structures were carried out using three different optimization techniques, namely linear-least-squares (LLS), Genetic Algorithm (GA), and adaptive simulated annealing (ASA). The standard FRF

*Received by the editors November 22, 2004; final revised form October 8, 2005.

based formulation was modified to cope with ill-conditioning and noise in a better manner. Although GA was used by some researchers for model updating of non-rotating structures, to the best knowledge of the author, developing and using a real-number GA code for model updating of a rotating structure has not been reported so far. Simulated annealing has also been used in different branches of engineering as an optimization tool. However, combining an adaptive version with an FRF based model updating code for a rotating structure is a novel approach which is used here.

2. FRF BASED MODEL UPDATING OF ROTATING STRUCTURE USING LEAST-SQUARES METHOD

The use of frequency response functions instead of modal parameters for model updating is relatively recent and presents some advantages [7-12]. Since the FRF is a measured quantity, errors due to modal parameter extraction are avoided. Furthermore, a large amount of data can be used to improve the stability of the updating equations. In other words, the problem can be made overdetermined due to the availability of FRF data at a large number of frequency points. Although it is not possible to write down as many independent equations as there are frequency points.

a) Formulation of FRF based model updating using linear-least-squares

A fundamental difference between various methods of model updating is related to the definition of the residual terms. In a specified frequency, one can define the residual as the difference between the forces acting on the real structure and those predicted by the analytical model [13]. Based on the exact dynamically reduced matrix, it is possible to define such a residue as [13]

$$\{\varepsilon(\{p\},\omega)\} = \{I\}_j - [Z_A(\{p\})]^{Red}\{H_X\}_j$$
$$[Z_A(\{p\})] = -\omega^2[M_A(\{p\})] + [K_A(\{p\})] + i\omega([C_A(\{p\})] + \Omega[G_A(\{p\})]) \quad (1)$$

The $[M_A]$, $[K_A]$, $[C_A]$ and $[G_A]$ are mass, stiffness, damping and gyroscopic matrices respectively. $[Z_A]$ is called the analytical impedance matrix and is a function of updating parameters $\{p\}$ and frequency (ω). In Eq. (1) Ω is the speed of rotation and $\{H_X\}$ is a vector of measured FRFs. One can use any condensation technique for the impedance matrix $[Z_A]$. Many such techniques have been developed and used for different purposes [14-15]. The reason for such condensation is to reduce the order of matrix $[Z_A]$ to the order of measured FRFs. An exact dynamic condensation was considered here as [13]

$$[Z_A]^{Red} = [Z_{nn}] - [Z_{ns}][Z_{ss}]^{-1}[Z_{sn}] \quad (2)$$

The proposed algorithm is effective and easy to implement. Here n and s are referred to measured and unmeasured DOFs respectively. By expanding $[Z_A]^{Red}$ in Eq. (1) via a Taylor series, and setting the residual vector to the null vector, after some mathematical manipulation one can obtain

$$\left(\sum_{i=1}^{N}[Z_A]^{Red}\left([H_A]\frac{\partial[Z_A]^{Red}}{\partial p_i}[H_A]\right)^{Red}[Z_A]^{Red}\Delta p_i\right)\{H_X\}_j = \{I\}_j - [Z_A^0]^{Red}\{H_X\}_j, \qquad j = 1,\ldots,J \quad (3)$$

Where J is the number of load cases, while $[Z_A^0]$ is the impedance matrix calculated at initial $\{p\}$ values. This set of equations can be written for different FRFs measured at different frequency points to obtain an overdetermined set of equations for unknown updating paramerters (Δp_i). Although many frequency points are available, it does not mean that it is possible to write down as many independent equations as frequency points. It seems that the updating frequencies that produce independent equations and their actual position are case dependent, the number of measured modes in the frequency range being the most important factor.

b) Error modelling

In the case of co-ordinate incompleteness, the solution of the updating equations is non-unique. Additional constraints, taking into account physical connectivities, can be introduced. In order to reduce the number of unknowns, one possible solution is to force the zero elements of the matrices to remain zero after updating. Further constraints can be introduced by considering the mass, stiffness and damping matrices of individual finite elements. Let us assume that modelling errors can be expressed as a linear combination of the individual element mass, stiffness, damping and gyroscopic matrices. In this way, the derivatives of matrices are constant and the maximum number of unknowns per element is reduced to four (See Eq. (1)). After some rearrangment, Eq. (3) can be written in final form as [13]

$$[A(\omega)]\{p\} = \{b\} \tag{4}$$

The overdetermined set of linear equations can be solved by the linear-least-square (LLS) technique to calculate the unknown vector {p}. The calculated parameters were then used to find an updated estimate for the stiffness, mass, damping and gyroscopic matrices. To obtain more accurate results, the whole process continues until the difference between two consecutive values of vector {p} is less than a prescribed threshold [3, 13].

c) Noise sensitivity reduction

The minimisation of the force-balance residue is prone to numerical problems since Eq. (1) is inherently ill-conditioned., i.e. $\{\varepsilon(\{p\},\omega)\}$ is a vector with some small elements. As a consequence, the cost function is very sensitive to measurement errors in $\{H_X\}_j$ and leads to biased parameters. Therefore, it is recommended to weight $\{\varepsilon(\{p\},\omega)\}$ by pre-multiplying it by the dynamic flexibility matrix of the analytical model calculated at initial {p} values, i.e. $[H_A^0]^{Red}$

$$\{\tilde{\varepsilon}\} = [H_A^0]^{Red}\{\varepsilon\} \tag{5}$$

Applying the same procedure explained in Section 2.a, one can arrive at

$$[H_A^0]^{Red}\left(\sum_i \frac{\partial [Z_A]^{Red}}{\partial p_i}\Delta p_i\right)\{H_X\}_j = \{H_A^0\}_j - \{H_X\}_j \tag{6}$$

Equation (6) is better conditioned than Eq. (3). Following the procedure in Section 2.2, an overdetermined set of equations will be obtained which can be solved using LLS. It can be seen from Eq. (6) that if one puts $[H_A^0] = [H_A]$, the new force residue will be the same as the receptance residue in [3]. Therefore, in this final form, the minimisation of the new receptance residue leads to the minimisation of the receptance residue. However, for greater selection of updating parameters, this minimisation is more stable [13].

Standard model updating can be regarded as a gradient-based optimization technique. In these methods, one needs to calculate the derivatives of objective function at every iteration, i.e. $\frac{\partial [Z_A]^{Red}}{\partial p_i}$. The stochastic optimization techniques, on the other hand, do not need the calculation of derivatives. Two such techniques will be explained in the following sections.

3. FRF BASED MODEL UPDATING OF ROTATING STRUCTURE USING GENETIC ALGORITHM

Optimisation problems with non-smooth, non-differentiable, highly non-linear and many local minima cost functions are commonly encountered in many model updating applications. Conventional gradient-

based algorithms are ineffective in these applications due to the problem of local minima or the difficulty in calculating gradients.

Optimisation methods that require no gradient and can achieve a global optimal solution offer considerable advantages in solving these difficult optimisation problems. The two best-known classes of such global optimisation methods are the genetic algorithm (GA) [16-18] and the simulated annealing (SA) [19-21].

Many model updating applications pose the following general optimisation problem:

$$Min \;\; \varepsilon(\{p\},\omega) = \sum_{i=1}^{N_f} log\left\| \{H_x\} - \{H_A(\{p\}, \omega_i)\} \right\|_2 , \qquad L_j \le p_j \le U_j \qquad 1 \le j \le N \tag{7}$$

Where $\{p\} = \{p_1, p_2, \ldots, p_N\}^T$ is the *N*-dimensional parameter vector to be optimised, N_f is the number of selected frequency in the frequency range of interest, and L_j and U_j are the lower and upper bounds of p_j , respectively. The cost function $\{\varepsilon(\{p\}, \omega)\}$ can be multidimensional and non-smooth and will be used for fitness calculation in both GA and ASA.

The GA may be thought of as an evolutionary process, where a population of solutions evolves over a sequence of generations. During each generation, the fitness (goodness) of each solution is calculated, and solutions are selected for reproduction on the basis of their fitness. The probability of survival of a solution is proportional to its fitness value. The reproduced solutions then undergo recombination, which consists of crossover and mutation.

A simple GA is really easy to use. It uses three basic generic operators: reproduction, crossover and mutation. Reproduction is a process in which individual solutions are copied according to their fitness value (objective function values). Crossover requires a mating of two randomly selected strings of solution. The information on the strings is partly interchanged according to a randomly chosen crossover site. Crossover is applied to take valuable information from the parents, and applied with a certain probability. Mutation is the occasional random alteration of the value of a string position. Mutation insures against bit loss, and can be a source of new bits.

The genetic algorithm used here requires determination of six fundamental issues: chromosome representation, selection function, the genetic operators making up the reproduction function, the creation of the initial population, termination criteria and the evolution function. Each of these issues will be described very briefly.

For any GA, a chromosome representation is needed to describe each individual in the population of interest. Each chromosome is made up of a sequence of genes in the form of binary digits (0 and 1), floating point number, integers, symbols (i.e. A, B, C, D), matrices and etc. In Holland's original paper [18], chromosomes are represented in binary digits. However, it was shown that natural representation is more efficient in terms of CPU times and offers more precision with more consistent results across replication [22]. This natural representation was used for the model updating example in this paper.

There are several schemes for the selection process. In this study, Roulette Wheel, which is the traditional selection function with the probability of surviving equal to the fitness of individual *i* divided by the sum of the fitness of all individuals was used.

Crossover takes two individuals and produces two new individuals, while mutation alters one individual to produce a single new solution. The application of these two basic types of operators and their derivatives depends on the chromosome representation used. The most important ones that can be used in the developed program are: 1- Arithmetic crossover, 2- Heuristic crossover, 3- Simple crossover.

Mutation is the occasional random alteration of the value of a string position. The applied mutation methods in the program are: 1-Boundary mutation, 2-Multi-non-uniform mutation, 3-Non-uniform

mutation, 4-Uniform mutation. More information on these and other related subjects can be found in Reference 16.

4. FINITE ELEMENT MODEL UPDATING OF ROTATING STRUCTURES USING ADAPTIVE SIMULATED ANNEALING

As mentioned in Section 3, the GA and SA belong to a class of so-called guided random search methods. The underlying mechanisms for guiding optimisation search process are, however, very different for the two methods. The GA is population-based, and evolves a population-based solution according to the principles of the evolution of species in nature. The SA, on the other hand, evolves a single solution in the parameter space with certain guiding principles that imitate the random behaviour of molecules during the annealing process.

Simulated annealing (SA) is an optimisation method initially developed for problems involving discrete variables. The ideas that form the basis of SA were first published by Metropolis *et al*. [23] in the early 1950s, while developing an algorithm to simulate the metallurgical annealing process. If a metal block is heated to a temperature below its melting point and then cooled back to the solid state, the structural properties of the cooled solid are a function of the rate of cooling. For example, if the cooling is very fast (quenching), then the metal develops several imperfections or faults, since the system converges to a higher energy state. On the other hand, if a sufficiently slow cooling schedule is employed, then the crystalline state is reached, since the system converges to a state of minimum energy. SA simulates the process of slow cooling of metals to achieve the minimum function value in a minimisation problem. The cooling phenomenon is modelled by controlling a temperature-like parameter introduced with the concept of Boltzmann probability distribution. By a controlled reduction in this temperature as the algorithm proceeds, the convergence of the algorithm can be controlled. Kirkpatrick *et al*. [20] suggested the use of SA as a technique for discrete optimisation in the early 1980s, and several researchers have successfully extended this method to problems involving continuous variables. SA has been mathematically proven to converge to the global optimum independent of starting values, given enough time to converge by using an appropriate 'temperature' reduction schedule [21].

An attractive feature of SA is that it is very easy to program and the algorithm typically has few parameters that require tuning. Furthermore, its statistical guarantee of convergence should make SA very appealing. However, a serious drawback of SA is that it is often very slow. In many diverse applications, a standard SA algorithm often requires many more cost-function evaluations to converge, compared with a carefully designed and tuned GA. However, an improved version of SA, referred to as the ASA (Adaptive simulated annealing), is very efficient in this sense [24-27]. The implemented ASA algorithm for model updating of rotating structures is briefly described here. More information regarding these matters can be found elsewhere [26, 28-29].

a) Adaptive simulated annealing objective function

The ASA evolves a single point p in the parameter or state space P. The seemingly random search is guided by certain underlying probability distributions. The objective function for this case is the same as Eq. (7). In general, the ASA algorithm can be described by three important functions. These functions are explained very briefly here.

b) Generating probability density function

$$G(p_i, T_{i,gen}) = \frac{1}{2} + \frac{\mathrm{sgn}(\gamma_i)}{2} \times \frac{\log(1+\gamma_i / T_{i,gen})}{\log(1+1/T_{i,gen})} \quad 1 \le i \le n \tag{8}$$

The above equation determines how a new state p_i^{new} is created, and from what neighbourhood and probability distributions it is generated, given the current state p_i^{old}. The generating temperatures $T_{i,gen}$ describe the widths or scales of the generating distribution along each dimension p_i of the state space. Often a cost function has different sensitivities along different dimensions of the state space. Ideally, the generating distribution used to search a steeper and more sensitive dimension should have a narrower width than that of the distribution used in searching a dimension less sensitive to change. The ASA adopts a so-called reannealing scheme to periodically re-scale $T_{i,gen}$, so that they optimally adapt to the current status of the cost function. This is an important mechanism which not only speeds up the search process, but also makes the optimisation process robust to different problems.

c) Acceptance function

$$h_{accept}(\varepsilon(p_i^{old}),\varepsilon(p_i^{new}), T_{accept}) = \frac{1}{1+\exp\left(\left(\varepsilon(p_i^{new})-\varepsilon(p_i^{old})\right)/T_{accept}(k_a)\right)} \tag{9}$$

This gives the probability of p_i^{new} being accepted. The acceptance temperature determines the frequency of accepting new states of poorer quality. Probability of acceptance is very high at very high temperature T_{accept}, and it becomes smaller as T_{accept} is reduced. At every acceptance temperature, there is a finite probability of accepting the new state. This occasionally produces an uphill move, enables the algorithm to escape from local minima, and allows a more effective search of the state space to find a global minimum. The ASA also periodically adapts T_{accept} to best suit the status of the cost function. This helps to improve convergence speed and robustness.

d) Reduce temperatures or annealing schedule

$$T_{i,gen}(k_a) = T_{i,gen}(0)\times\exp(-ck_a^{1/n}), \quad T_{i,gen}(k_i) = T_{i,gen}(0)\times\exp(-ck_i^{1/n}) \quad 1\le i\le n \tag{10}$$

where k_a and k_i are some annealing time indexes. The reduction of temperatures should be gradual enough to ensure that the algorithm finds a global minimum. This mechanism is based on the observations of the physical annealing process. When the metal is cooled from a high temperature, if the cooling is sufficiently slow, the atoms line themselves up and form a crystal, which is the state of minimum energy in the system. The slow convergence of many SA algorithms is rooted in this slow annealing process. The ASA, however, can employ a very fast annealing schedule, as it has self-adaptation ability to re-scale temperatures.

e) ASA algorithm implementation

Although there are many possible realisations of the ASA, an implementation of this algorithm is used in the study detailed here. How the ASA realises the above three functions will also become clear during the following description.

(i) In the initialisation, an initial $p\in P$ is randomly generated, the initial temperature of the acceptance probability function, $T_{accept}(0)$, is set to $\varepsilon(p)$, and the initial temperatures of the parameter generating probability functions, $T_{i,gen}(0)$, $1\le i\le n$, are set to 1.0. A user-defined annealing control parameter *c* is given, and the annealing times, k_i and k_a is all set to 0.

(ii) The algorithm generates a new point in the parameter space with

$$p_i^{new} = p_i^{old} + \gamma_i(U_i - L_i) \qquad 1\le i\le n \tag{11}$$

Where γ_i is calculated from Eq. (8) as

$$\gamma_i = \mathrm{sgn}(u_i - 0.5) \times T_{i,gen}(k_i) \times \left(\left(1 + \frac{1}{T_{i,gen}(k_i)} \right)^{|2u_i - 1|} - 1 \right) \tag{12}$$

where u_i is a uniformly distributed random variable in interval [0,1]. Notice that if a generated p^{new} is not in P, it is simply discarded and a new point is tried again until $p^{new} \in P$. The value of the cost function $\varepsilon(p^{new})$ is evaluated and the acceptance probability function of p^{new} is then calculated using Eq. (9). A uniform random variable h_{unif} is generated in [0, 1]. If $h_{unif} \le h_{accept}$, p^{new} is accepted, otherwise it is rejected.

(iii) After every N_{accept} acceptance points, reannealing takes place by first calculating the sensitivities

$$s_i = \left| \frac{\varepsilon(p^{best} + \delta e_i) - \varepsilon(p^{best})}{\delta} \right| \qquad 1 \le i \le n \tag{13}$$

Where p^{best} is the best point found so far, δ is a small step size, the *n*-dimensional vector e_i has unit *i*-th element and the rest of the elements of e_i are all zeros. Let $s_{\max} = Max\{s_i,\ 1 \le i \le n\}$. Each parameter generating temperature $T_{i,gen}$ is scaled by a factor $s_{\max} / s_i$, and the annealing time k_i is reset

$$T_{i,gen}(k_i) \leftarrow \frac{s_{\max}}{s_i} T_{i,gen}(k_i) \quad , \quad k_i = \left(-\frac{1}{c} \log\left(\frac{T_{i,gen}(k_i)}{T_{i,gen}(0)} \right) \right)^n \tag{14}$$

Similarly, $T_{accept}(0)$ is reset to the value of the last accepted cost function, $T_{accept}(k_a)$ is reset to $\varepsilon(p^{bnest})$, and the annealing time k_a is rescaled accordingly

$$k_a = \left(-\frac{1}{c} \log\left(\frac{T_{accept}(k_a)}{T_{accept}(0)} \right) \right)^n \tag{15}$$

(iv) After every N_{gen} generated points, annealing takes place with

$$\begin{aligned} k_i &= k_i + 1 \\ T_{i,gen}(k_i) &= T_{i,gen}(0) \times \exp(-c k_i^{1/n}) \quad 1 \le i \le n \end{aligned} \tag{16}$$

And

$$\begin{aligned} k_a &= k_a + 1 \\ T_{accept}(k_a) &= T_{accept}(0) \times \exp(-c k_a^{1/n}) \end{aligned} \tag{17}$$

Otherwise, go to step (ii).

(v) The algorithm is terminated if the parameters have remained unchanged for a few successive reannealings or a pre-set maximum number of cost function evaluations have been reached; otherwise, go to step (ii).

The optimal values of N_{accept} are problem dependent, but experience suggests that an adequate choice for N_{accept} is in the range of tens to hundreds, and an appropriate value for N_{gen} is in the range of hundreds to thousands. The annealing rate control parameter can be determined from the chosen initial temperature, final temperature and predetermined number of annealing steps. We have found out that a choice of *c* in the 0.01 to 0.1 range is often adequate.

A necessary and sufficient condition for the convergence of different SA algorithms can been found in [26, 28-29]. Table 1 compares the convergence of different SA algorithms, i.e. standard SA, fast SA and ASA. In this table, the relations k_i is an annealing time index and n is the dimension of unknown parameters.

Table 1. Convergence rate of different SA algorithms

Simulation method	Standard SA	Fast SA	ASA
$T_{i,gen(k_i)}$	$\frac{T_{i,gen(0)}}{\ln(k_i)}$	$\frac{T_{i,gen(0)}}{k_i}$	$\frac{T_{i,gen(0)}}{\exp(ck_i^{1/n})}$

The convergence rate of an SA algorithm is determined primarily by its annealing schedule. The slow convergence of the standard SA is inherent from the annealing schedule (see Table 1). The ASA, therefore, has the fastest convergence rate among the three algorithms tabulated here.

5. NUMERICAL EXAMPLES

Figure 1 shows the overall flowchart of the developed program. The details of GA and ASA are not shown for brevity. The system shown in Fig. 2 represents the data from a real rotor with bearing, discs, added mass and etc (part of an active magnetic bearing system). It consists of 36 nodes and 59 elements. Eight sensors are located in x and y direction at different positions of the shaft (Fig. 3).

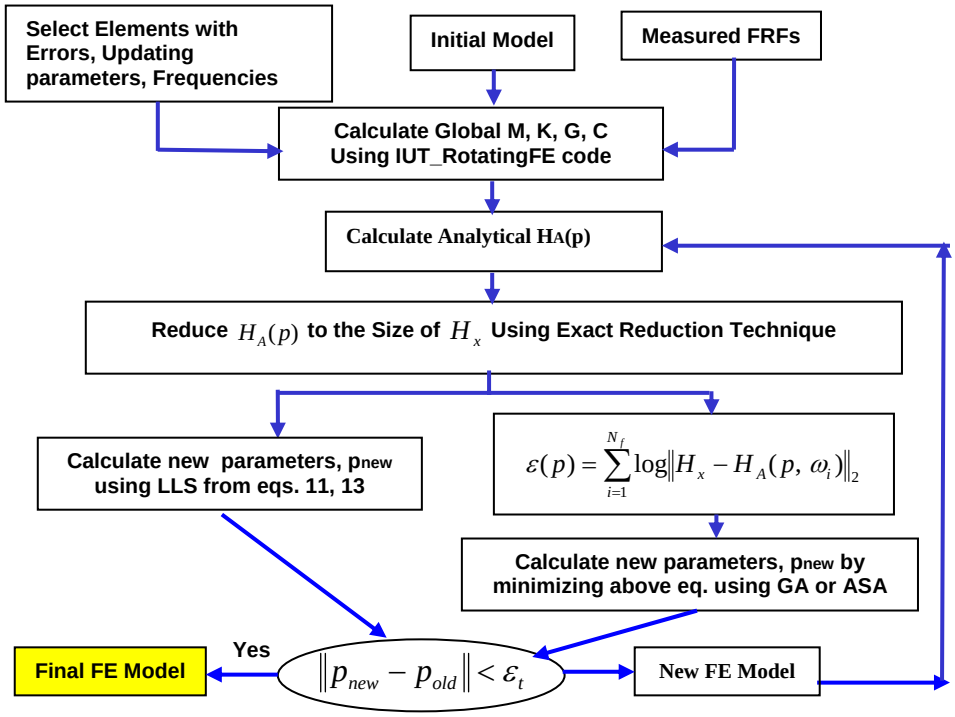

Fig. 1. Overall flowchart for model updating of rotating structures using LLS, GA and ASA

The dash lines in Fig. 2 are additional distributed masses which are modelled in IUT_Rotating FE (A finite element code developed by the author for rotating structures) by use of very low stiff elements. Table 2 tabulated some geometrical properties of the model.

Table 2. Some geometrical properties of the rotor

Variable	RotFE
Total mass	3.41
Centre of mass	0.24
Radial moment of inertia	0.054
Polar moment of inertia	4.31e-004

Next, the natural frequencies of the rotor at zero speed were calculated by IUT_Rotating FE. The results are presented in Table 3 and show a good agreement with the results from other commercial packages (Here MECOS - A commercial European rotating structure code).

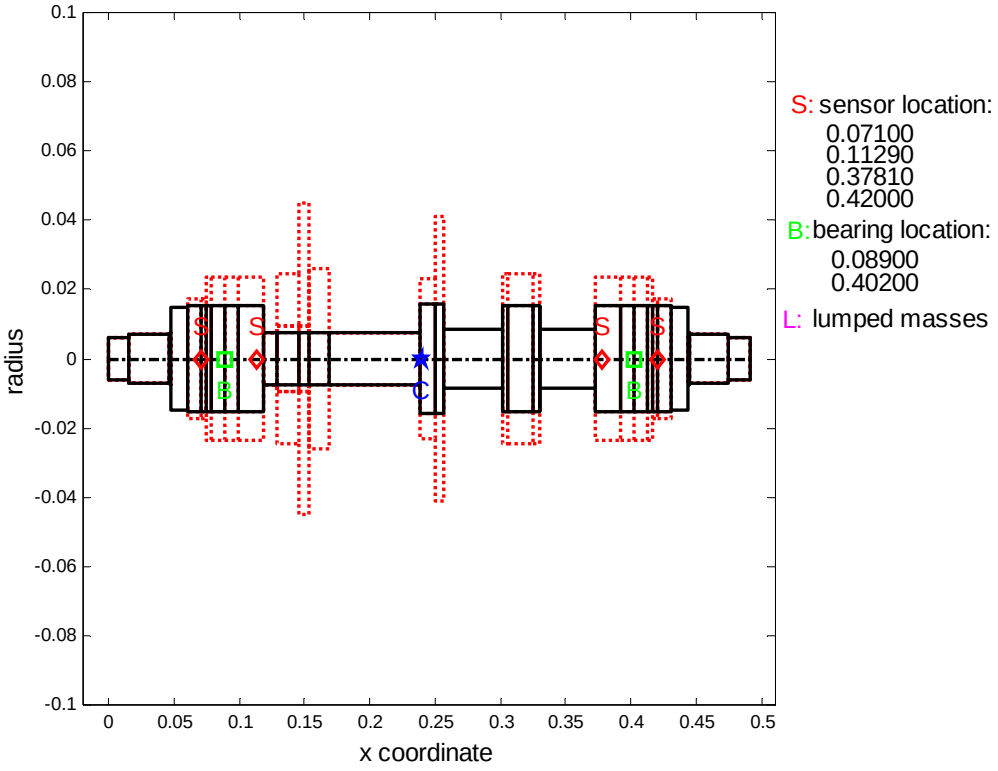

Fig. 2. Test rotor system

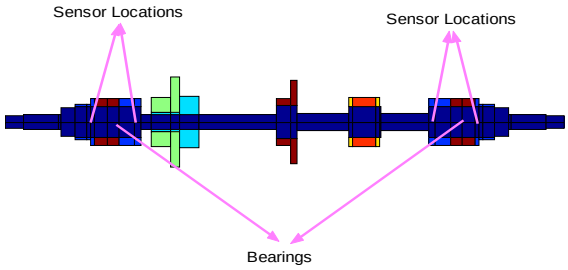

Fig. 3. FE model of the test rotor system

Table 3. Natural frequencies comparison of rotor model by IUT_Rotating FE and MECOS

Natural Frequencies (Hz)	RotFE	MECOS	Difference (%)
1	0	0	0
2	0	2.13e-3	0
3	211.4	210.5	0.43
4	591.19	586.9	0.73
5	1261.29	1246	1.2
6	2165.31	2115	2.3
7	3039.29	2970	2.3

a) FE model updating of the test rotor system using simulated data without noise

First, it was decided to simulate an experimental data set by a numerical model without noise. Therefore, two models were built. The first model, 'experimental model', was used to simulate the experimental data, while some parameters were changed in the second model, hereafter called the analytical model. The stiffness of all bearing in both x and y directions for the updating study were assumed to be 250KN. The aim is to use the developed updating procedure to find out the exact values of the parameters in error (the real values of all parameters are 1.0). The amount of errors introduced in some elements of the analytical models for each case are given in Table 4. In this Table, M7, K50 and G23 are the mass, stiffness and gyroscopic terms of elements 7, 50 and 23 respectively, while K_{xx} and K_{yy} are the stiffness of bearing in x and y direction.

Table 4. Error introduced for model updating

Error (%)	M7	K50	G23	Kxx1	Kyy1	Kxx2	Kyy2
Case 1	0%	0%	0%	8%	-10%	12%	-15%
Case 2	10%	20%	12%	8%	-10%	15%	-15%

For case 1, we assume that the errors is only at the stiffness of the bearings, i.e. K_{xx1}, K_{yy1}, K_{xx2} and K_{yy2}. The introduced error were 8, -10, 12 and -15 percent for K_{xx1}, K_{yy1}, K_{xx2} and K_{yy2}, respectively. For case 2, in addition to the previous errors, errors were introduced in the mass of element 7 (10%), stiffness of element 50 (20%) and the gyroscopic term in element 23 (12%).

Eleven frequency points are selected in the 0-120 Hz frequency range. They are namely; 20, 25, 30, 35, 45, 55, 63, 90, 100, 110 and 120 Hz. The excitation force assumed to be in x direction at sensor location 1 (see Fig. 3), and their responses are measured at x and y directions in all 4 sensor locations. For both cases, the 'measurements' were carried out at two different rotor speeds, namely 5000 and 10000 RPM. Therefore, a total of 16 FRFs were measured at each frequency point.

Figures 4 and 5 show convergence towards the final solution for cases 1 and 2 of the test rig using the linear-least squares (LLS) method. In both cases, the frequency response functions were updated correctly in the sense that the p-values matched those of the experimental model. It is worth mentioning that the total size of the receptance matrix is 144. This means that 8 out of 144 (5.6%) possible measured FRFs are enough for the problem to converge to its correct solution.

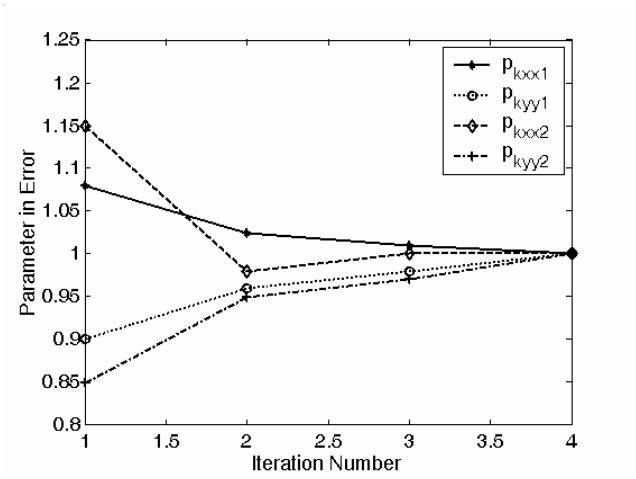

Fig. 4. Convergence of parameters for the test rig using LLS -Case 1 (p_{kxx1} : correction factor for stiffness in x direction at bearing 1, ….)

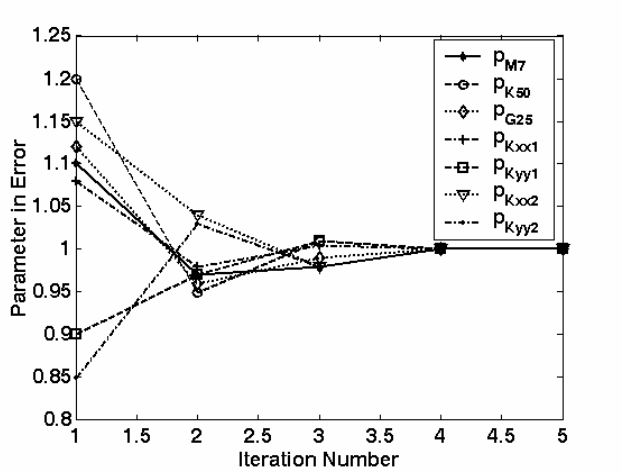

Fig. 5. Convergence of parameters for the test rig using LLS -Case 2 (p_{M7} : correction factor for the mass matrix of element number 7, ….)

It can be concluded that the procedure for LLS is working satisfactorily for simulated data and in absence of any noise. As the main idea for model updating of the test rig was to adjust the stiffness of the bearings, only case 1 will be used for further investigation.

Next, the genetic algorithm is used to capture the error location and magnitude using the same data as used for case 1. The optimisation problem is similar to the one defined in Eq. (7), i.e.

$$Min \quad \{\varepsilon(\{p\},\omega)\} = -\sum_{i=1}^{N_f} log\left\|\{H_x\} - \{H_A(p,\omega_i)\}\right\|_2 , \qquad L_j \le p_j \le U_j \qquad 1 \le j \le N \tag{18}$$

The negative sign in Eq. (18) is due to the fact that genetic algorithm always search for the maximum (here we assume that the value of ε is less than unity). Norm geometry selection was used as selection function, while heuristic crossover and uniform mutation options were used as genetic operators. An initial population of 10 was created for this case, and after some tuning the algorithm was run and the results are tabulated in Table 5 and Fig. 6.

Table 5. The p-values obtained for the test rotor – Case 1 (no noise)

P-values	Kxx1	Kyy1	Kxx2	Kyy2
Initial	1.0800	0.9000	1.1500	0.8500
Target	1.0	1.0	1.0	1.0
LLS (% Error)	1.0 (0.0 %)	0.9999 (-0.01%)	1.0001 (0.01%)	1.0 (0.0%)
GA (% Error)	1.0000 (0.0%)	0.9993 (-0.07%)	1.0000 (0.0%)	0.9965 (-0.35%)
ASA (% Error)	1.0000 (0.0%)	0.9926 (-0.74%)	0.9998 (-0.02%)	0.9972 (-0.28%)

The same problem was then solved using adaptive simulated annealing. The lower and upper bonds of the parameters in Eq. (7) were set to be -0.9 and 2, respectively. Note that the exact value for all parameters is one. After tuning the initial parameters of the algorithm, the program runs. Figure 7 shows the results for the test rig using simulated data and no noise. In this figure, the objective function is depicted against the number of iterations. The final p values for this case are tabulated in Table 5.

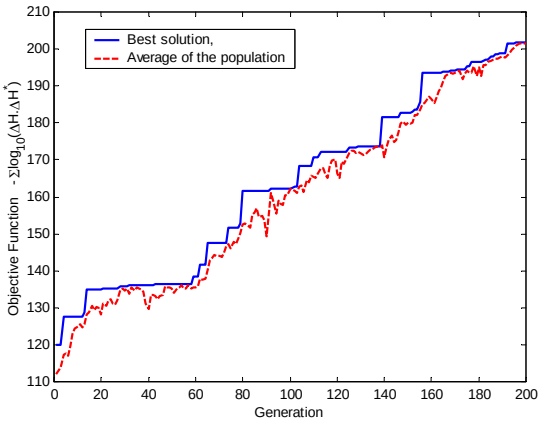

Fig. 6. The track of the best solution and the average of the population for the test rig using GA-Case 1

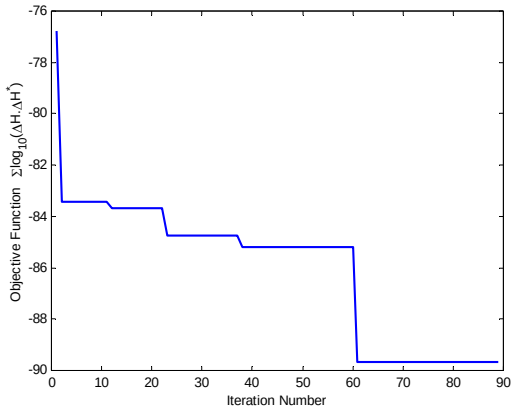

Fig. 7. Convergence for model updating of the test Rig using ASA algorithm-Case 1

b) FE model updating of the test rig using simulated data with noise

The test rotor was then studied for updating with noisy data. The stiffness of the bearings, i.e. Kxx1, K_{yy1}, K_{xx2}, K_{yy2} were considered as parameters in error. Random noise with zero mean value and variance one was created and added to the simulated FRFs. Two different cases with different noise levels were investigated. Table 6 shows each case study in more detail.

Table 6. Case studies for the test rotor

Case	% of noise in FRF	Elements in error	Optimisation method
3	1	K_{xx1}, K_{yy1}, K_{xx2}, K_{yy2}	LLS, GA, ASA
4	10	K_{xx1}, K_{yy1}, K_{xx2}, K_{yy2}	LLS, GA, ASA

Case 3: The parameters in error are the same as case 1. Figure 8 plots the convergence of the parameters for case 3, i.e. FRFs polluted with 1 percent of noise and LLS technique.

For GA, the initial population was set to 100 for the current cases. This increased the CPU time required for a converged solution. The heuristic crossover and non-uniform mutation were selected as GA operators for cases 3 and 4. The results after 100 generations for this are reported in Table 7. Figure 9 shows the trace of convergence for this case.

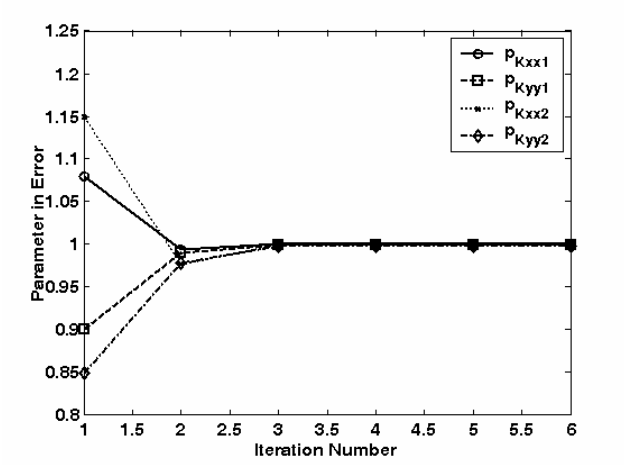

Fig. 8. The convergence of parameters in error for case 3 using LLS (1% noise)

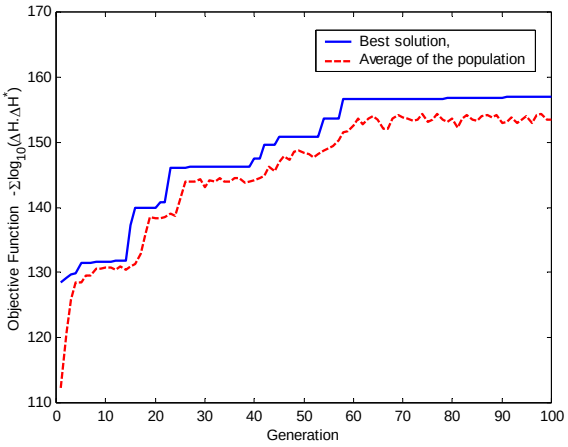

Fig. 9. The track of the best solution and the average of the population for the test rotor with 1% noise (case 3) using GA

Table 7. The p-values obtained for the test rotor – 1% noise

P-values	K_{xx1}	K_{yy1}	K_{xx2}	K_{yy2}
Initial	1.0800	0.9000	1.1500	0.8500
Target	1.0	1.0	1.0	1.0
LLS (% Error)	1.0008 (0.08%)	0.9992 (-0.08%)	0.9998 (-0.02%)	0.9978 (-0.22%)
GA (% Error)	1.0013 (0.13%)	1.0093 (0.93%)	0.9999 (-0.01%)	1.0091 (0.91%)
ASA (% Error)	1.0007 (0.07%)	1.0003 (0.03%)	1.0000 (0.0%)	1.0015 (0.15%)

Figure 10 depicts the objective against the number of iterations for case 3 using ASA. Table 7 summarises the final p values for this case using different optimisation techniques. It can be concluded that for this case, all three methods can correctly capture the errors in the model.

Case 4: Figure 11 shows a plot of parameters for case 4 when the FRFs are polluted by 10% random noise using the LLS technique. Although the convergence to an acceptable level of the parameters in error was achieved, the parameters did not converge to the exact solution. The offset is mainly due to the presence of noise and is increased by increasing the noise level. By increasing the noise level even more, the solution diverges. This clearly shows the reduction in the sensitivity of the method to the level of the noise as described in Section 2.5. The standard method (FRF without noise sensitivity reduction) cannot tolerate 10% of noise and diverges.

Next, the GA algorithm was employed while the initial population was set to 100. Again, the results after 100 generations were shown in Fig. 12 and the value of the parameters are presented in Table 8. It can be seen from the figure that the final value of the objective function for this case is less than the

previous two cases, i.e. 134.84, 156.96, 202.45 for 0, 1 and 10 percent noise respectively. This is mainly due to the presence of noise in the FRFs which makes the convergence slow.

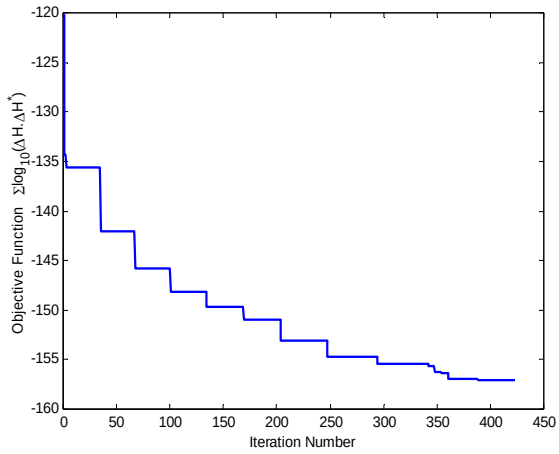

Fig. 10. Convergence for model updating of the test rotor with 1% noise (case 3) using ASA algorithm

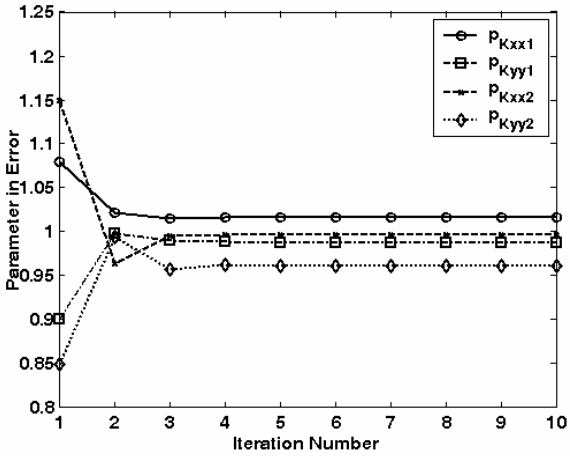

Fig. 11. The convergence of parameters in error for case 4 using LLS (10% noise)

The ASA algorithm was then employed. The performance of the ASA with GA and LLS are compared in Table 8. Again, the number of iterations required to obtain a converged solution increased as the noise level was increased (Fig. 13). Morever, an offset similar to LLS can be observed from both the GA and ASA results.

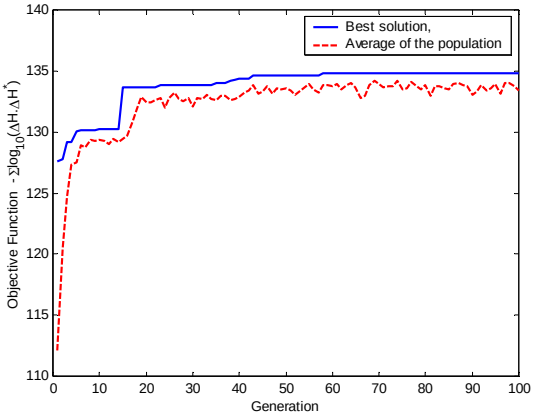

Fig. 12. The track of the best solution and the average of the population for the test rotor with 10% noise (case 4) using GA

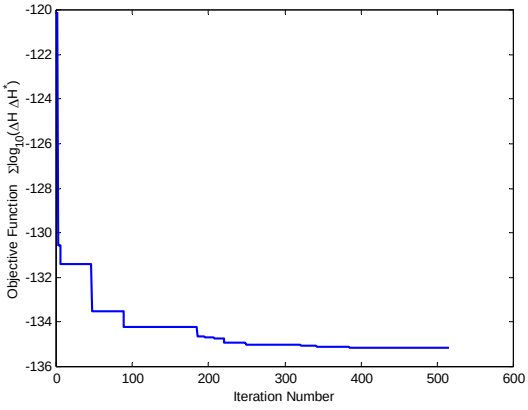

Fig. 13. Convergence for model updating of the test rotor with 10% noise (case 4) using ASA algorithm

Table 8. The p-values obtained for the test rotor – 10% noise

P-values	K_{xx1}	K_{yy1}	K_{xx2}	K_{yy2}
Initial	1.0800	0.9000	1.1500	0.8500
Target	1.0	1.0	1.0	1.0
LLS (% Error)	1.0078 (0.78%)	0.9932 (-0.68%)	0.9983 (-0.17%)	0.9793 (2.07%)
GA (% Error)	1.0181 (1.81%)	1.0093 (0.93%)	0.9949 (-0.51%)	1.0091 (0.91%)
ASA (% Error)	1.0068 (0.68%)	0.9916 (0.84%)	1.0008 (0.08%)	1.0175 (1.75%)

Regarding performance comparison, the LLS method is faster than both GA and ASA algorithms. However, in many of the cases it diverged. The GA and ASA have slower convergence rates, but can provide some improved results with respect to the initial guess. The GA is likely to be trapped in a local minimum if the size of the initial population is not large enough. On the other hand, the performance of

the ASA is based on the annealing rate. A large annealing rate makes the convergence faster. However, it may not give enough time to the liquid to reach to the minimum state energy, i.e. it becomes trapped in a local minimum. On the other hand, a small annealing ratio needs greater iteration and will increase the CPU time substantially.

c) Convergence and CPU time study of LLS, GA and ASA algorithms

Equations (3) and (6) are mainly based on expansion by the Taylor series and are somehow the linearization of the main residue problems, i.e Eqs. (1) and (5). The model updating problems are generally a function of updating parameters {p} and frequency ω, i.e.

$$\{\varepsilon(\{p\},\omega)\} = \{I\}_j - [Z_A(\{p\},\omega)]^{Red}\{H_X\}_j \tag{19}$$

Note that the elements of the dynamic stiffness matrix [Z] (not $[Z]^{Red}$) are smooth functions of frequency and the vector of updating variables, i.e. {p}. As long as the changes in these parameters remain physically acceptable, the elements will keep their smooth behaviour. However such smoothness does not exist for the reduced dynamic stiffness matrix $[Z]^{Red}$.

Figures 14 and 15 show the variation of element $Z_{6x,6x}$ and $Z^{Red}_{6x,6x}$ as a function of frequency at a constant {p} for our test rotor structure. Actually, the singularities in the reduced dynamic stiffness matrix results from incomplete measurements. It can be seen that $Z^{Red}_{6x,6x}$ is a highly nonlinear function, and therefore the linearization of Eq. (5) cannot be justified at all frequencies. In general, Z^{Red} is a very complicated function of updating variables and frequency. This is the main reason that all the FRF-based algorithms do not work properly in practice and the reported results are mostly case dependent. The situation becomes more complicated when random noises due to experiments were added to the analytical data or when the data comes directly from the measurements. Therefore, we have to search for methods that directly use the main residue problem and not the linearised version of it. However, such algorithms are not as fast as the LLS algorithm and usually need a vast bulk of evaluation of the objective function.

For a test rotor structure, $[Z]^{Red}$ is a smooth function for frequencies up to a frequency limit, here 180 Hz (see Fig. 14). This is why we selected eleven frequency points in the frequency range of 0-120 Hz for our case studies (Cases 1 to 4). If frequency points were selected beyond 180 Hz, one expects to have difficulties in convergence of the LLS algorithm. Another issue is noise. By increasing the noise level, the smoothness of $[Z]^{Red}$ deteriorates at every frequency point, which may also cause instability in LLS numerical algorithm.

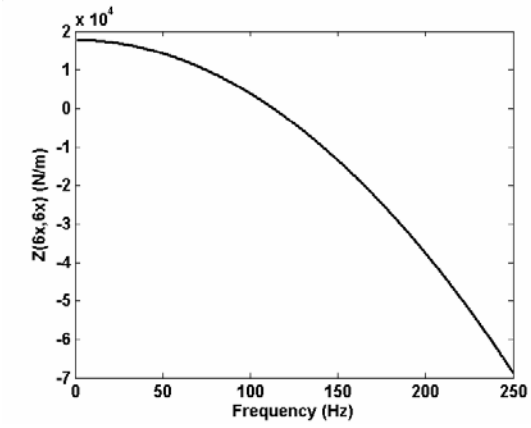

Fig. 14 . Element $Z_{6x,6x}$ of test rotor structure as a function of frequency ({p}=constant)

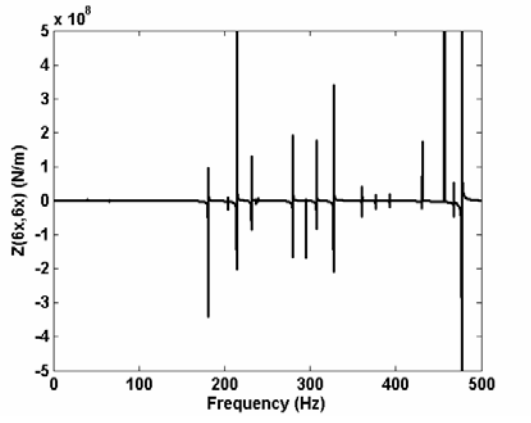

Fig. 15. Element $Z^{Red}_{6x,6x}$ of test rotor structure as a function of frequency ({p}=constant)

Table 9 compares the convergence of different algorithms for various selections of frequency points and noise level. It can be seen from the table that LLS diverges if some of the frequency points were selected beyond the frequency limit (180 Hz), while GA and ASA both converge to their correct values at all sets of frequency points (exact value is one for all parameters in error). It is worth mentioning that the frequency limit relates to the ratio of measured DOFs to the total DOFs in the FE model. By decreasing the ratio, the limit frequency decreases and so the range of validity of linearization for the LLS algorithm also decreases. However, this is not the case for GA and ASA as they use the main residue equation directly. Both GA and ASA can also tolerate a high level of noise. For very high noise levels, these two algorithms converge, but not to their correct values. This means that they are trapped in some local minima.

Table 9. Convergence comparison of different algorithms for various selection of frequency points and noise level

Frequency points (Hz) and Noise level	LLS (K_{xx1}, K_{yy1}, K_{xx2},K_{yy2})	GA (K_{xx1}, K_{yy1}, K_{xx2},K_{yy2})	ASA (K_{xx1}, K_{yy1}, K_{xx2},K_{yy2})
20, 25, 30, 35, 45, 55, 63, 90, 100, 110, 120 (10% Noise)	1.0078, .9932, .9983, .9793	1.0181, 1.0093, 0.9949, 1.0091	1.0068, 0.9916, 1.0008, 1.0175
100, 120, 170, 220, 230, 250, 300, 330 (1% Noise)	Diverged	1.0071, 1.0053, 0.9973, 0.9986	1.0055, 1.0043, 0.9977, 1.0089
100, 120, 170, 220, 230, 250, 300, 330 (10% Noise)	Diverged	1.0125, 1.0156, 0.9885, 0.9863	1.0219, 1.0155, 1.0179, 0.9842
260, 290, 300,310, 330, 400,420, 440 (1% Noise)	Diverged	1.0132, 0.9867, 1.0145, 0.9944	1.0098, 1.0122, 0.9876, 0.9983
260, 290, 300,310, 330, 400,420, 440 (10% Noise)	Diverged	1.0247, 0.9865, 1.0357, 0.9645	1.0198, 1.0147, 1.0289, 0.9844

Table 10 tabulated the CPU time required for the LLS, GA and ASA algorithms to converge. The reported values are based on the calculation using a Pentium 4 (2.4 GHz). It is notable that LLS is very fast in comparison with GA and ASA. SA algorithms are normally slower than GA, however, the automatic step selection used here causes the ASA to be faster than GA.

Table 10. CPU time required for calculation of case studies 1 to 4

Case Studies	CPU time (sec) – Pentium 4 (4.2 GHz)		
	LLS	GA	ASA
Case 1: 4 parameters - 0% noise	25.3	1534.3	1254.9
Case 2: 7 parameters – 0% noise	47.9	3768.9	2876.4
Case 3: 4 parameters - 1% noise	44.8	2544.7	2256.8
Case 4: 4 parameters – 10% noise	77.6	3199.7	2896.7

6. CONCLUSIONS

A treatment to avoid ill-conditioning in the minimisation of force balance residue was made by pre-multiplying the residue vector by the dynamic flexibility matrix of the analytical model. It was shown that this could partly reduce the sensitivity of the method to measurement errors.

It was also shown that the linearisation of the highly nonlinear and nonsmooth objective function which appears in most model updating problems is not always a good option and may cause instability and convergence towards the local minima. Gradient methods such as LLS are ineffective in such nonlinear and nonsmooth cost functions. Therefore, optimisation methods that require no gradient and can achieve a global optimal solution offer considerable advantages. Two well-known classes of such global optimisation methods are introduced, namely the genetic algorithm (GA) and the adaptive simulated

annealing (ASA), both of which were implemented in a program specifically written for the model updating of rotating structures.

Some case studies were carried out for simulated data with and without noise. It was concluded that the generic algorithm (GA) might become trapped in local minimum if the size of the initial population is not large enough. However, a large initial population may need huge computational effort and is time consuming.

A new simulated annealing method called "adaptive simulated annealing", which is fast and does not need initial tuning, was adopted and implemented in the program. It was found that this method could detect the locations and the magnitude of the elements in error. It was concluded that the linearisation of the analytical receptance matrix is faster than any other method investigated here. ASA and the fine tuned GA have the following two ranks respectively. However, due to inherent nonlinearity of the objective function, it may be prone to divergence, instability or converging to incorrect parameter values, particularly in the presence of high levels of experimental noise.

REFERENCES

1. Mottershead, J. E. & Friswell, M. I. (1993). Model updating in structural dynamics: A survey. *J. of Sound and Vibration, 167*, 347-375.
2. Natke, H. G. (1988). *Updating* computational models in the frequency domain based on measured data: A survey. *Probabilistic Engineering Mechanics, 3*, 8-35.
3. Visser, W. J. & Imregun, M. (1991). A technique to update finite element models using frequency response data. *Proc. of IMAC 9.*
4. Bucher, I. & Ewins, D. J. (1996). Modal testing of rotating structures: difficulties, assumptions and practical approach. *ImechE Conference Transaction of the Sixth International Conference on Vibration in Rotating Machinery*, Oxford.
5. Bucher, I. & Ewins, D. J.(1996). *A method for model updating undamped gyroscopic structures. Conference Transaction of Model Updating, Lake Districts*.
6. Ziaei Rad, S. (2002). *IUT_Rotating FE Manual.* Isfahan University of Technology.
7. Lammens, S. (1993). Model updating using experimental frequency response functions: case studies. *Proc. of IMAC11*, 449-455, Kissimmee, Florida.
8. Larsson, P. O. & Sas, P. (1992). Modal updating based on force vibration testing using numerically stable formulations. *Proc. of IMAC10*, 966-974, Sandiego, California.
9. Levin, R. I. & Lieven, N. A. J. (1998). Dynamic finite element model updating genetic algorithms. *Mechanical Systems and Signal Processing, 12*(1), 91-120.
10. Lin, R. M. & Ewins, D. J. (1990). Modal updating using FRF data. *Proc. of the 15th International Seminar on Modal Analysis*, 141-162, Leuven, Belgium.
11. Nalitolela, N. G. (1993). A frequency domain technique for structural parameter updating. *Proc. of IMAC11*, 676-682, Kissimmee, Florida.
12. Ziaei Rad, S. & Imregun, M. (1996). On the accuracy required of experimental data for finite element model updating. *Journal of Sound and Vibration, 196*(3), 323-336.
13. Ziaei-Rad, S. (1997). Methods for updating numerical models in structural dynamics. PhD Thesis, Imperial College, London.
14. Meirovitch., L. (1980). *Computational methods in structural dynamics*. Sijthoff and Noordhoff.
15. O'Callahan, J.C. & Li, P. (1995). SEREP expansion. *Proc. of IMAC13, 1258-1264*, Nashville, Tennessee.
16. Davis, L. (1991). *Handbook of genetic algorithms*. Van Nostrand Reinhold.

17. Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA.
18. Holland, J. H. (1975). *Adaptation in natural and artificial systems.* University of Michigan Press, Ann Arbor, Michigan.
19. Corana, A., Marchesi, M., Martini, C. & Ridella, S. (1987). Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. *ACM Trans. Mathematical Software, 3*(13), 262-280.
20. Kirkpatrick, S., Gelatt Jr., C. D. & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220, 671-680.
21. Van Laarhoven, P. J. M. & Aarts, E. H. L. (1989). *Simulated annealing: theory and applications*. Dordrecht, The Netherlands, Kluwer Academic Press.
22. Michalewicz, S. (1994). *Genetic algorithm+data structures=Evolution programs.* AI Series, Springer-Verlag, New York.
23. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics, 2*, 1087-1090.
24. Chen, S., Luk, B. L. & Liu, Y. (1998). Application of adaptive simulated annealing to blind channel identification with HOC fitting. *Electronics Letters, 3*(34), 234-235.
25. Ingber, L. (1993). Simulated annealing: practice versus theory. *Mathematical and Computer Modelling, 11*(18), 29-57.
26. Ingber, L. & Rosen, B. (1992). Genetic algorithms and very fast simulated reannealing: a comparison. *Mathematical and Computer Modelling, 11*(16), 87-100.
27. Rosen, B. E. (1997). Rotationally parameterized very fast simulated reannealing. *IEEE Trans. Neural Networks*.
28. Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distribution and the Bayesian restoration in images. *IEEE Trans. Pattern Analysis and Machine Intelligence*(6), 721-741.
29. Szu, H. & Hartley, R. (1987). Fast simulated annealing. *Physics Letters A*, (122), 157-162.