Advances in Engineering Software 40 (2009) 883–891

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

A hybrid Particle Swarm Optimization – Simplex algorithm (PSOS) for structural damage identification

O. Begambre [a,b,*], J.E. Laier [a]

[a] São Carlos Engineering School, Department of Structures, University of São Paulo, São Carlos-SP, Brazil
[b] Escuela de Ingenieria Civil, Universidad Industrial de Santander, Bucaramanga, Santander, Colombia

ARTICLE INFO

Article history:
Received 7 January 2009
Accepted 22 January 2009
Available online 9 March 2009

Keywords:
Particle Swarm Optimization
Damage identification
Inverse problems
Truss structure
Cracked beam
Non-linear oscillator

ABSTRACT

This study proposes a new PSOS-model based damage identification procedure using frequency domain data. The formulation of the objective function for the minimization problem is based on the Frequency Response Functions (FRFs) of the system. A novel strategy for the control of the Particle Swarm Optimization (PSO) parameters based on the Nelder–Mead algorithm (Simplex method) is presented; consequently, the convergence of the PSOS becomes independent of the heuristic constants and its stability and confidence are enhanced. The formulated hybrid method performs better in different benchmark functions than the Simulated Annealing (SA) and the basic PSO (PSO_b). Two damage identification problems, taking into consideration the effects of noisy and incomplete data, were studied: first, a 10-bar truss and second, a cracked free–free beam, both modeled with finite elements. In these cases, the damage location and extent were successfully determined. Finally, a non-linear oscillator (Duffing oscillator) was identified by PSOS providing good results.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

This work is an updated and revised version of the conference paper [1], which presented a modified PSO using a direct search complex algorithm to control the PSO heuristic parameters. The present article introduces an improved version of that algorithm, called hybrid PSO Simplex algorithm, for damage detection. The main advantage of this new approach is its lower computational cost (number of function evaluations and CPU time) when compared with Ref. [1].

The methods for damage detection (DD) using dynamic responses can be classified into two large groups in accordance with their dependence (or not) on a structural model: the methods based on signals (experimental) and the methods based on models [2,3].

One of the signal-based methods is the technique of Novelty Detection [4], whose principal objective is to extract features from dynamic data that characterize the state of the structure. This approach allows only for the lowest level of damage detection, i.e., deciding whether the damage has occurred or not. The other methods based on signals, i.e., methods of thermal and magnetic field, visual inspection, acoustic emission, X-rays, ultrasound methods and Eddy's currents technique [5], need an *a priori* knowledge of the damage location and the guarantee of access to any part of the structure. Besides these limitations, the signal-based methods can only detect damage in (or) near the structure surface and work relatively well in small-sized structures. The main advantage of these techniques is that they avoid modelling errors and high computational costs involved in numeric simulations. However, these methods are inefficient when applied to large structural systems, where the techniques based on the vibratory response of the structure are more promising due to their global character. On the other hand, the methods based on models allow determining the damage location and severity by the simultaneous use of mathematical models of the structure and experimental vibration data [6].

The DD problem can be defined as a non-linear inverse problem [7]. As the experimental data are usually limited, multiple solutions that satisfy the formulation of the inverse problem can be obtained. To overcome this difficulty, computational techniques, like Artificial Neural Networks (ANNs) [8,9] and Genetic Algorithms (GAs) [10,11], among others, have been used to solve the DD task. The central idea of these techniques is to substitute the inverse problem by a group of direct problems. Such solution allows obtaining a representation (vector) of the system, where the variations in the parameters of the structure, owing to the damage, are heightened, allowing for the identification of possible flaws. The drawback of classical non-linear programming techniques, when applied to DD, is that they are susceptible to converge to local optima and therefore, they cannot be employed successfully in non-linear and multimodal problems, such as the ones studied in this paper.

The main difficulty found in DD based on models is the presence of errors in both the experimental measurements and the models (modelling errors). In the first case, the effect of the errors can be

* Corresponding author. Address: São Carlos Engineering School, Department of Structures, University of São Paulo, São Carlos, SP, Brazil.
E-mail address: ojbegam@uis.edu.co (O. Begambre).

0965-9978/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2009.01.004

reduced by using high-accuracy sensors and reliable signal measuring to obtain the experimental data that will be used either during the training process of the NNs, or in the definition of the objective function employed in the GAs. The error in the Finite Element Model (FEM) is normally reduced by the use of model updating techniques [12]. Another issue that must be addressed in model-based DD is the damage model. In most cases, the damage is simulated by either reducing the stiffness of individual elements of the Finite Element Model (FEM) [13,14] or using simplified crack models [15] and the linear fracture mechanics theory [16]. The problems mentioned above have been the theme of current research and generated the motivation for the present work.

In this study, the DD problem is defined as an unconstrained non-linear mathematical programming problem, in which we specify an objective function in terms of parameters related to the physical properties of the structure. Knowing that the optimum of the objective function is obtained when evaluated with true parameters and using a hybrid PSO Simplex algorithm (PSOS) to estimate them, it is possible to assess the state of the system.

A critical issue in using a heuristic approach to solve an optimization problem is the selection of an extreme excess of parameters that control its performance [17]. In order to manipulate the PSO parameters, Parsopoulos and Vrahatis [17] used the Differential Evolution Algorithm (DEA) [18], but this method contains user's defined heuristics constants and, therefore, the convergence of the combined algorithm is not totally independent of the heuristic parameter selection and its computational cost is high. The present article proposes a novel strategy for the proper choice of the PSO constants based on the Simplex method [19]. Despite all the known failures and inefficiencies of such method [20], it was used here to manipulate the initial selection of the PSOS heuristic parameters, and in this manner, the PSOS becomes independent of these values. In addition, the combination produces improvement with only few iterations of the hybrid algorithm (PSOS). The use of the Simplex method has another advantage: it works only with function values (it is a derivative free algorithm).

The proposed procedure aims to identify if single or multiple damage has occurred, to determine its position inside the structure and to estimate its severity (extension) using Frequency Response Functions (FRFs) and PSOS.

This study is structured as follows: first, the hybrid algorithm used in this work was introduced; next, three benchmark functions were studied to assess the performance of the PSOS algorithm; then, the damage identification problem, its formulation as a non-linear optimization problem and its solution through PSOS were discussed; subsequently, two DD numerical problems in which we supposed that the initial model is a precise representation of the intact structure were analyzed; a non-linear system was identified (Duffing oscillator) and finally our conclusions and recommendations were presented.

2. The PSO

The Particle Swarm Optimization Algorithm is a member of the wide category of swarm intelligence methods to solve non-linear programming problems. It was recently proposed by Kennedy and Eberhart [21] and Kennedy [22]. Since its introduction, many applications to structural and multidisciplinary optimization have been published. A review of PSO applications in solving optimization problems in the area of electric power systems is presented in [23]. Other applications to optimal building and design problem can be found in [24]. Additional applications to structural shape and size optimization are presented in [25–27]. The PSO algorithm has recently been used to train ANNs in order to predict real-time water levels and algal bloom in aquatic systems [28,29].

In the PSO there are several explicit parameters whose values affect the manner in which the algorithm searches the problem space [21–27]. These heuristic parameters impact the convergence properties of the PSO, i.e., they may cause premature convergence of the search. To overcome these problems, the performance of PSO can be improved by controlling these constants, as explained in the next sections.

2.1. PSO: basic parameters

In this work, the global asynchronous [26] version (with linear inertia reduction) of the PSO algorithm was implemented to solve our DD problem. The basic algorithm assumes that each particle (candidate solution) in the population (swarm or set of N particles) flies over the search space looking for promising regions of the landscape, i.e., in a maximization problem, regions that possess higher function values than others previously discovered. In this context, the position of each particle is updated based on the social information shared by the members of the swarm and each particle attempts to change its position to a point where it has a higher cost function value at previous iterations. The particles are manipulated according to the following vectorial equations [27]:

$$p^i_{k+1} = p^i_k + v^i_{k+1}, \tag{1}$$

$$v^i_{k+1} = \omega v^i_k + C_1 rand_1(b^i_k - p^i_k) + C_2 rand_2(b^g_k - p^i_k), \tag{2}$$

where k indicates a unit pseudo-time increment, p^i_k represents the position of each particle i (candidate solutions), p^i_{k+1} is the position of particle i at time $k+1$, b^i_k represents the best ever position of particle i at time k (best individual position), b^g_k is the best position in the swarm at time k (global best), v^i_k is the velocity of particle i at time k and v^i_{k+1} is the updated velocity of particle i at time $k+1$. All vectors in Eqs. (1) and (2) are of dimensions $m \times 1$, where m is the number of optimized parameters, $rand_1$ and $rand_2$ are independent random numbers (with uniform probability) between 0 and 1. Parameters C_1 and C_2 control the flow of information between the current swarm. If $C_2 > C_1$, then the particle puts more trust in the swarm, otherwise, it puts more confidence in itself. C_1 and C_2 are known as the cognitive and social parameters, respectively. ω is the inertia factor (or inertia weight) that controls the impact of the previous particle velocity on the current particle velocity [30].

The PSO algorithm proceeds by modifying the distance each particle moves in each direction per iteration. In order to control the step length of the algorithm (velocity) and to prevent the explosion phenomenon [31], the value $v^i_{max} = (pUB - pLB)/H$ was used. In the above expression, v^i_{max} is the particle's maximum step length, pUB and pLB are the upper and lower bounds of each particle and H is a parameter that controls the size of the step length. It is worth noting that all the particle positions p^i_k must be limited by their lower and upper bounds (pLB, pUP).

The literature [17,21–23,30] proposes using $0 < \omega < 1.4$, $C_1 = C_2 = 2$ with $C_1 + C_2 \leqslant 4$ and $5 < H < 10$ to maintain a balance between the global and local search capabilities of the algorithm. In our implementation of the basic PSO (PSO_b), a value of $H = 5$ was selected. It was observed that this value worked satisfactorily in all examples presented in this work. As it is known, the three parameters ω, C_1 and C_2 are problem-dependent [25,26]. Due to this fact, even experienced PSO users have to perform exhaustive testing to find the best set of PSO constants and less skilled users may provide the algorithm with an improper set of parameters causing its failure.

3. The Nelder–Mead algorithm

In this work, the original version of the Nelder and Mead Simplex algorithm [19] was utilized, where $n+1$ points (these points

Download English Version:

https://daneshyari.com/en/article/568431

Download Persian Version:

https://daneshyari.com/article/568431

Daneshyari.com