International Journal of Engineering & Applied Sciences (IJEAS)
Vol.5, Issue 1(2013)1-17

STRUCTURAL DAMAGE DETECTION FOR BEAMS SUBJECT TO MOVING LOAD USING PSO ALGORITHMS

Hakan Gökdağ

Bursa Technical University, Mechanical Engineering Dept. Osmangazi Bursa, Turkey
**E-mail address*: *hakan.gokdag@btu.edu.tr*

Received: 25 February 2013

Abstract

In this work, damage detection (DD) method for beam structures subject to moving load is proposed. DD is formulated as an optimization problem and solved for crack locations and depths using three versions of the particle swarm optimization (PSO). It was observed that PSO with constriction factor is superior in the sense of convergence speed and robustness. Also, it was experienced that small cracks with depth ratio of 0.15 can be identified by the present method in spite of 3% noise interference. The proposed method is demonstrated to be better than wavelet transform method at higher moving load speeds.

Keywords: Damage detection; particle swarm optimization; moving load.

1. Introduction

Structures under moving load have many practical applications such as railway tracks, bridges, pipelines, roadways, etc. Basically, a moving load yields larger deflections and higher stresses than equivalent static load conditions. Thus, dynamics of such structures has received considerable attention. Frýba collected a lot of approaches in the well-known textbook [1], investigating mainly beam type structures under moving load with various loading and boundary conditions. If the host structure has crack-like local defects, than the impact of moving load becomes more pronounced [2-6]. On the other hand, various DD methods have been developed for beams subject to moving loads using the continuous wavelet transform (CWT) [7-11]. They are based on the fact that CWT coefficients of beam dynamic response demonstrate local peaks at crack locations, and magnitudes of these peaks are proportional to crack depths.

In structural DD there are other methods based on model updating (MU) [12]. The basic of such a method is to update mathematical or finite element model of the structure to match the calculated response to the one measured from damaged structure. This is achieved through an optimization procedure. To solve the optimization problem the mathematical programming (MP) or metaheuristic techniques are employed. The MP methods can be categorized as gradient techniques such as the generalized reduced gradient technique, successive quadratic programming, Lagrange Newton approaches, successive linear programming and interior-point methods. Their common deficiency is that solution quality relies on the initial point. They may be trapped into local optima easily. Also, computational complexity becomes significant as the problem dimensions rise (see [13] for further disadvantages of these). To overcome the limitations of the MP methods, metaheuristic approaches, one of which is the PSO algorithm, have been introduced. PSO, developed by Kennedy and Eberhart [14], is a stochastic optimization technique inspired by natural flocking and swarm behavior of birds and insects. Unlike genetic algorithms (GAs), PSO has no evolution operators such as crossover and mutation. Also, PSO is easy to implement and there are few parameters to adjust. The information

sharing mechanism in PSO is significantly different than GAs. In GAs, chromosomes share information with each other, thus the whole population moves like a one group towards an optimal area. In PSO, only global and local best points give out the information to others, hence it is a one-way information sharing mechanism. Compared with GAs, all the particles tend to converge to the best solution quickly [15,16]. PSO has been successfully employed in MU based DD applications [17-19]. In MU based DD, time dependent structural response is used, as well. Buezas et al. [20] formulated an optimization problem using time responses from several points on the beam, and determined crack size and depth by solving this problem.

Bilello and Bergman [4] concluded that changes in the time-response of the beam due to damage are more perceptible in comparison to the changes in the natural frequencies. In the present work, motivated by the conclusion of Bilello and Bergman [4] and the method of Buezas et al. [20], a MU based DD approach has been proposed for beam type structures carrying moving load. In this respect, time dependent displacements from several points on a cracked beam were obtained, and an objective function was defined by subtracting these from the ones calculated by the mathematical model of the structure. Then, three variants of PSO were employed to minimize this objective function, and their performances were compared considering several damage scenarios. It was demonstrated that the proposed method is superior to the wavelet coefficients method at higher moving load speeds. To the best of the author's knowledge, this is the first study which deals with formulating the DD in a beam subject to moving load as an inverse problem and solved by the various PSO algorithms.

2. Formulation of The Problem

2.1. Vibration modes of cracked beam

For simplicity, the system including beam and moving load is considered in this work. This is illustrated in Figure 1, where a simply supported beam of length L is carrying moving load P with constant speed V_{p} . E and ρ refer to the modulus of elasticity and density of beam material. I and A are, respectively, the second moment of inertia and cross-sectional area of the beam. The beam contains an open crack at z_1 with depth h_1 . Assuming the beam is composed of two parts joined at the crack location through a rotational spring, we can write the following compatibility equations at the crack location [2]

$$Y_1(z_1)=Y_2(z_1),\ Y'_1(z_1)+\theta Y''_1(z_1)=Y'_2(z_1),\ Y''_1(z_1)=Y''_2(z_1),\ Y'''_1(z_1)=Y'''_2(z_1) \tag{1}$$

where $'=d/dz$, and $Y_i(z)$ is the mode shape of the i$^{\mathrm{th}}$ beam part defined as follows

$$Y_i(z)=C_{i1}\cos(\lambda z)+C_{i2}\sin(\lambda z)+C_{i3}\cosh(\lambda z)+C_{i4}\sinh(\lambda z),\ i=1,2 \tag{2}$$

where $\lambda=(\rho A\omega^2/(EI))^{0.25}$. In Eq.(2) λ and ω are eigenvalue and natural frequency parameters, respectively, and C_{ij} are the constants to be determined by solving the eigenvalue problem. The first, third and fourth terms in Eq.(1) mean deflection, bending moment and shear force at the crack location are continuous while the second term indicates slope discontinuity due to the crack. The geometric factor of the crack, θ, is defined as follows [2,6]

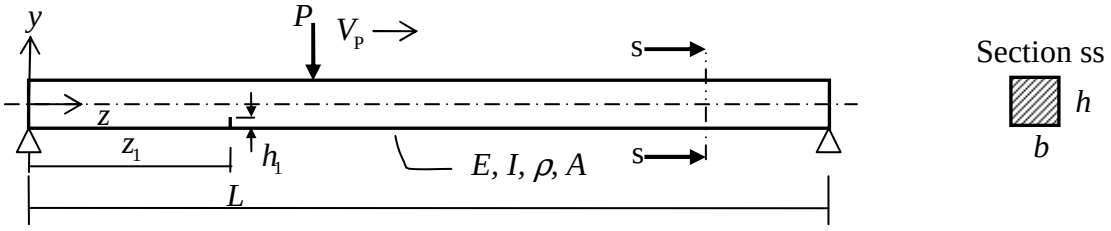


Figure 1: The cracked beam carrying moving load

$$\theta = 2h\left(\frac{\delta_1}{1-\delta_1}\right)^2\left(5.93-19.69\delta_1+37.14\delta_1^2-35.84\delta_1^3+13.12\delta_1^4\right),\ \delta_1 = h_1/h \tag{3}$$

$Y_i(z)$ in Eq.(2) and its first three derivatives can be written in matrix form:

$$\mathbf{V}_i = \mathbf{T}\mathbf{C}_i \tag{4}$$

where

$$\mathbf{V}_i=\begin{Bmatrix} Y_i(z) \\ Y_i'(z) \\ Y_i''(z) \\ Y_i'''(z) \end{Bmatrix},\ \mathbf{T}=\begin{bmatrix} \cos(\lambda z) & \sin(\lambda z) & \cosh(\lambda z) & \sinh(\lambda z) \\ -\lambda\sin(\lambda z) & \lambda\cos(\lambda z) & \lambda\sinh(\lambda z) & \lambda\cosh(\lambda z) \\ -\lambda^2\cos(\lambda z) & -\lambda^2\sin(\lambda z) & \lambda^2\cosh(\lambda z) & \lambda^2\sinh(\lambda z) \\ \lambda^3\sin(\lambda z) & -\lambda^3\cos(\lambda z) & \lambda^3\sinh(\lambda z) & \lambda^3\cosh(\lambda z) \end{bmatrix},\ \mathbf{C}_i=\begin{Bmatrix} C_{i1} \\ C_{i2} \\ C_{i3} \\ C_{i4} \end{Bmatrix} \tag{5}$$

Then, Eq.(1) in matrix notation:

$$\mathbf{U}\mathbf{V}_1 = \mathbf{V}_2 \ \Rightarrow\ \left[\mathbf{U}\mathbf{T}(z_1)\ \ -\mathbf{T}(z_1)\right]_{4x8}\begin{Bmatrix}\mathbf{C}_1 \\ \mathbf{C}_2\end{Bmatrix}_{8x1}=\mathbf{0},\quad \mathbf{U}=\begin{bmatrix}1&0&0&0\\0&1&\theta&0\\0&0&1&0\\0&0&0&1\end{bmatrix} \tag{6}$$

In order to determine the elements of the vectors $\mathbf{C}_1$ and $\mathbf{C}_2$ eight equations are required. Eq.(6) provides the four of these. The remaining four come from the boundary conditions, which are given in Eqs.(7-8) for the simply supported beam.

At the left end $Y_1(0)=0$ and $Y_1''(0)=0$, i.e.

$$\mathbf{T}_L\mathbf{C}_1=\mathbf{0},\ \mathbf{T}_L=\begin{bmatrix}1&0&1&0\\0&\lambda&0&\lambda\end{bmatrix} \tag{7}$$

At the right end $Y_2(L)=0$ and $Y_2''(L)=0$, which gives

$$\mathbf{T}_{\mathrm{R}}\mathbf{C}_2=\mathbf{0},\ \mathbf{T}_{\mathrm{R}}=\begin{bmatrix}\cos(\lambda L) & \sin(\lambda L) & \cosh(\lambda L) & \sinh(\lambda L)\\ -\cos(\lambda L) & -\sin(\lambda L) & \cosh(\lambda L) & \sinh(\lambda L)\end{bmatrix} \tag{8}$$

With Eqs.(6-8) the following eigenvalue problem is formulated

$$\mathbf{A}\begin{Bmatrix}\mathbf{C}_1\\ \mathbf{C}_2\end{Bmatrix}_{8\mathrm{x}1}=\mathbf{0},\ \mathbf{A}=\begin{bmatrix}\mathbf{T}_{\mathrm{L}\,2\mathrm{x}4} & \mathbf{0}_{2\mathrm{x}4}\\ \mathbf{T}^{(1)}{}_{4\mathrm{x}8} & \\ \mathbf{0}_{2\mathrm{x}4} & \mathbf{T}_{\mathrm{R}\,2\mathrm{x}4}\end{bmatrix}_{8\mathrm{x}8},\ \mathbf{T}^{(1)}=\left[\mathbf{UT}(z_1)\quad -\mathbf{T}(z_1)\right]_{4\mathrm{x}8} \tag{9}$$

For the nontrivial solution of Eq.(9) the determinant of **A** should be zero. By this way the natural frequencies of the cracked beam, ω_i, and corresponding mode shapes, $Y^{(i)}(z)$ i=1,2,…, are obtained. In the case of multiple cracks one can verify that Eq.(9) becomes as follows

$$\begin{bmatrix}\mathbf{T}_{\mathrm{L}\,2\mathrm{x}4} & & & & \mathbf{0}\\ & \mathbf{T}^{(1)} & & & \\ & & \mathbf{T}^{(2)} & & \\ & & & \cdots & \\ & & & & \mathbf{T}^{(N_{\mathrm{c}})}\\ \mathbf{0} & & & & \mathbf{T}_{\mathrm{R}\,2\mathrm{x}4}\end{bmatrix}_{4(N_{\mathrm{c}}+1)\mathrm{x}4(N_{\mathrm{c}}+1)}\cdot\begin{Bmatrix}\mathbf{C}_1\\ \mathbf{C}_2\\ \cdot\\ \cdot\\ \cdot\\ \mathbf{C}_{N_{\mathrm{c}}+1}\end{Bmatrix}_{4(N_{\mathrm{c}}+1)\mathrm{x}1}=\mathbf{0},\ \mathbf{T}^{(i)}=\left[\mathbf{UT}(z_i)\quad -\mathbf{T}(z_i)\right]_{4\mathrm{x}8} \tag{10}$$

where N_{c} is the number of cracks. Again the same method is applied to solve Eq.(10). By this way, natural frequencies and mode shapes of the multiple-cracked beam are obtained.

2.2 Dynamic response of the beam subject to moving load

Equation of motion of the beam in Figure 1 is [7]

$$\rho A\frac{\partial^2 y(z,t)}{\partial t^2}+C\frac{\partial y(z,t)}{\partial t}+EI\frac{\partial^4 y(z,t)}{\partial x^4}=-P\delta(z-V_{\mathrm{p}}t) \tag{11}$$

where C and δ are, respectively, damping coefficient of the beam and Dirac delta function. Using natural modes of the cracked beam Eq.(11) can be solved by the mode superposition method. Thus, assuming $y(z,t)=\sum_{i=1}^{N_m}q_i(t)Y^{(i)}(z)$, where N_{m} is the number of modes employed, substituting into Eq.(11), multiplying by $Y^{(j)}(z)$ and integrating along the beam lead to

$$\mathbf{M}\ddot{\mathbf{q}}+\mathbf{C}\dot{\mathbf{q}}+\mathbf{K}\mathbf{q}=\mathbf{F} \tag{12}$$

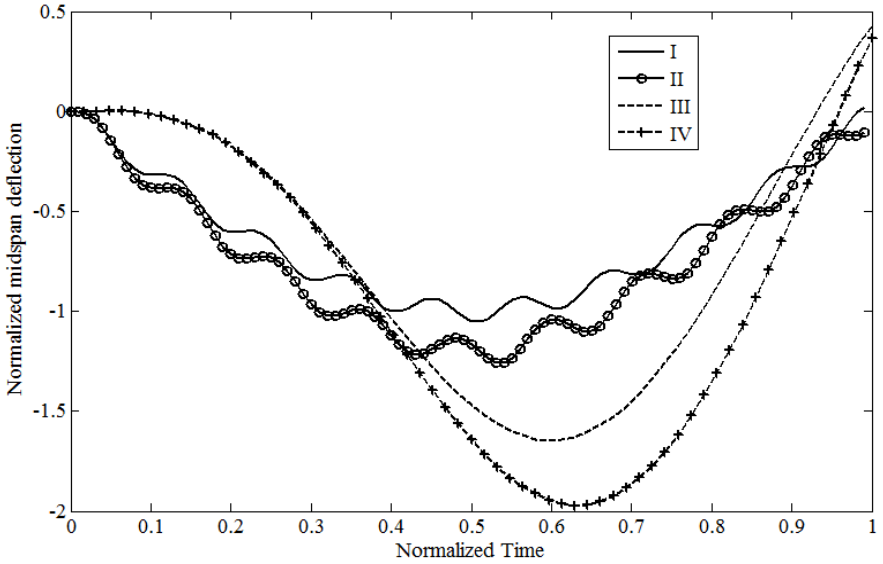
where **M**, **C** and **K** are mass, damping, and stiffness matrixes, respectively, with the following entities

$$M_{ij}=\int_0^L\rho AY^{(i)}(z)Y^{(j)}(z)\mathrm{d}z,\ C_{ij}=\int_0^L CY^{(i)}(z)Y^{(j)}(z)\mathrm{d}z, \tag{13.1}$$

$$K_{ij} = \int_0^L EIY'''^{(i)}(z)Y^{(j)}(z)\mathrm{d}z\,,\ F_i = \int_0^L -P\delta(z - V_P t)Y^{(j)}(z)\mathrm{d}z \tag{13.2}$$

$$\mathbf{q} = [q_1, q_2, ..., q_{N_\mathrm{m}}]^\mathrm{T},\quad i,j=1,2,\ldots,N_\mathrm{m} \tag{13.3}$$

In this work Eq.(12) is solved by the Newmark Beta method (with β=1/6 and γ=1/2 [21]). To verify the codes written for computations, midspan deflections of the beam were obtained using the following data [7]: E=210GPa, ρ=7860 kg/m^3, L=50m, b=0.5m, h=1m, P=10 kN, sampling frequency 100 Hz, number of modes $N_\mathrm{m}=5$, 2% modal damping for each mode. Figure 2 illustrates the normalized midpoint deflections for different speed and damage combinations. Normalization is performed dividing by the static deflection, i.e. $PL^3/(48EI)$, of the midpoint when P acts on this point. One can verify the agreement comparing the curves in Figure 2 with those in [2,6,7].



$,V_\mathrm{P}=5\mathrm{m/s},\ \delta_1=0$ **Figure 2**: Normalized midspan deflection of the beam. I: II: $V_\mathrm{P}=5\mathrm{m/s},\ \delta_1=0.5$, III: $V_\mathrm{P}=40\mathrm{m/s},\ \delta_1=0$, IV: $V_\mathrm{P}=40\mathrm{m/s},\ \delta_1=0.5$

2.3 The objective function and the constrains

The aim is to correlate the response of the damaged structure to the one calculated by the mathematical model of the structure. To achieve this, it is proposed to adjust crack sizes and locations by solving an optimization problem. The objective function of the problem is introduced as follows.

$$f(\mathbf{z}) = \sum_{n=1}^{N_\mathrm{mp}} \int_0^T \frac{\left|y(z_n,t) - \bar{y}(z_n,t)\right|}{\max\left(\left|\bar{y}(z_n,t)\right|\right)}\mathrm{d}t \tag{14}$$

where N_mp is the number of measurement points on the beam, z_n is the location of the nth measurement point on the beam, $\bar{y}$ denotes the reference displacements measured from damaged beam whereas y stands for the corresponding displacements computed by the mathematical model

of the structure. T is the total time for the load to move across the beam, and $\mathbf{z}$ is the vector containing crack location and size parameters, i.e.

$$\mathbf{z} = \left\{ \delta_1, \delta_2, ... \delta_{N_c}, \overline{z}_1, \overline{z}_2, ..., \overline{z}_{N_c} \right\}, \quad \delta_i = \frac{h_i}{h}, \ \overline{z}_i = \frac{z_i}{L} \tag{15}$$

The ith element of $\mathbf{z}$ denotes the depth ratio of the ith crack while its $(N_c + i)$th element shows the location of the ith crack. As to the measurement points, it is better to choose them close to the midpoint, since maximum deflection occurs at the beam midspan. Thus, four points [20] on the beam were determined as $\{0.3\ 0.5\ 0.6\ 0.7\}L$, i.e. $N_{\text{mp}} = 4$ in Eq. (14). If the number of cracks (N_c) is more than one, then extra constraints should be introduced for the optimization algorithm to make search in the feasible region. With these explanations in mind, the optimization problem can be formulated as follows

$$\text{min } f(\mathbf{z}) \text{ subject to: } \overline{z}_i - \overline{z}_{i+1} < 0 \text{, i=1,2,…, } N_c - 1 \text{ and } 0 < \overline{z}_j < 1, \ 0 \le \delta_j < 1 \text{, j=1,2,…, } N_c \tag{16}$$

Crack locations and depths can be determined by solving Eq.(16). In this work the PSO is employed for this purpose, and its details are given in the next section. At each iteration, the candidate solutions are checked if crack locations satisfy the relevant constrains in Eq.(16). Those not satisfying are penalized by adopting extremely high objective function value such as 10^{10}.

3. The Particle Swarm Optimization (PSO) Algorithm

PSO algorithm is initialized with a "swarm" composed of N particles. Particles refer to the candidate points in the search space of the optimization problem. To obtain the best solution each particle adjusts its trajectory toward its own previous best position and toward the previous best position of the swarm. By this way, each particle moves in the search space with an adaptive velocity, and stores the best position of the search space. Location (x) and velocity (v) of a particle are updated with the following equations [14,15].

$$\begin{aligned} v_{ij}^{k+1} &= v_{ij}^k + c_1 R_1 (p_{ij}^k - x_{ij}^k) + c_2 R_2 (p_{gj}^k - x_{ij}^k) \\ x_{ij}^{k+1} &= x_{ij}^k + v_{ij}^{k+1}, \quad i = 1, 2, .., N, \ j = 1, 2, ..., m, \ k = 1, 2, ..., K_{\max} \end{aligned} \tag{17}$$

where k is the iteration counter, $K_{\max}$ denotes the maximum number of iterations, m is the problem dimension, p_{ij}^k and p_{gj}^k are, respectively, the best positions of the ith particle and the swarm found until the kth iteration, R_1 and $R_2 \in U(0,1)$, where U means the uniform random distribution, c_1 and c_2 are positive weighting constants called cognitive and social coefficients, respectively. These two constants regulate the relative velocity toward global and local best points. The algorithm using Eq.(17) is called standard PSO, the pseudo code of which is illustrated in Figure 3. Shi and Eberhart [22] further studied the performance of PSO, and modified it by introducing a constant called "weighting coefficient", w, as follows

$$v_{ij}^{k+1} = w v_{ij}^k + c_1 R_1 (p_{ij}^k - x_{ij}^k) + c_2 R_2 (p_{gj}^k - x_{ij}^k) \tag{18}$$

This w plays the role of balancing the global and local searches. The authors, assuming $c_1 = c_2 = 2$, concluded that the PSO with inertia weight in the range [0.9,1.2] on average has a bigger chance to

find the global optimum within a reasonable number of iterations. Later studies revealed that PSO parameters cannot be selected arbitrarily, i.e. they must satisfy certain mathematical relationships to guarantee convergence [15,23]. One way to improve the convergence speed of PSO is to decrease *w* during the optimization process. In this case the algorithm is initialized with a bigger value (w_{up}) to promote exploration in the early stages of optimization. Then it is decreased linearly, as in Eq.(19), to a smaller value (w_{low}) to eliminate oscillatory behaviors in later stages.

```
Step 1.   Set k ← 0
Step 2.   Initialize S and Set P≡S
Step 3.   Evaluate S and P, and define index g of the best position
Step 4.   While (termination criterion not met)
Step 5.         Update S using Eq. (1)
Step 6.         Evaluate S
Step 7.         Update P and redefine index g
Step 8.   Set k ← k+1
Step 9.   End While

S: swarm, P: best values of particles
```

Figure 3: Pseudo-code of PSO [15].

$$w^k = w_{up} - (w_{up} - w_{low})\frac{k}{K_{max}} \tag{19}$$

This version of PSO is called linearly decreased inertia weight (LDIW) PSO [15,22,24]. On the other hand, Zheng et al. [25] proposed an opposite approach in which inertia weight is increased from a lower value to an upper one iteratively. This is called as linearly increased inertia weight (LIIW) PSO. In their opinion, either global or local search ability associates with a small inertia weight, which possesses the capacity of exploring new space. Besides, a large inertia weight provides the algorithm more chances to be stabilized. According to the PSO algorithm with LIIW, velocity of each particle is updated as follows.

$$v_{ij}^{k+1} = w^k v_{ij}^k + \overline{c}_1(p_{ij}^k - x_{ij}^k) + \overline{c}_2(p_{gj}^k - x_{ij}^k)\ ,\ \overline{c}_n = b_n R_n + d_n\ ,\ (n{=}1{,}2) \tag{20}$$

where $b_1 = b_2 = 1.5$, $d_1 = d_2 = 0.5$, and w^k is the inertia weight linearly increasing from 0.4 to 0.9, i.e.

$$w^k = 0.4 + (0.9 - 0.4)\frac{k}{K_{max}} \tag{21}$$

Clerc and Kennedy [26], replacing inertia weight with constriction factor (χ), introduced another version known as contemporary PSO [15], in which the velocity of each particle is updated as follows.

where $\chi = 2\left(\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|\right)^{-1}$, $\varphi = c_1 + c_2$, and $\varphi > 4$. This version of the PSO is denoted in this work by the abbreviation CF (constriction factor). The CF not only induces particles to converge on local optima, but also prevents swarm explosion, which is a common problem with traditional PSO algorithms [26]. In this work, the three PSO variants given above, also summarized in Table 1, are employed. Algorithms were executed in MATLAB environment. To test the written codes two

common benchmark functions, i.e. Rosenbrock (f_1) and Sphere (f_2) [15,22,27], are utilized with the relevant parameters in Table 2.

Table 1. PSO variants used in this work.

Method	Relevant Equations	Parameters	References
LDIW	(18), (19)	, $w_{\text{low}} = 0.4$, $c_1 = c_2 = 2$ $w_{\text{up}} = 0.9$	[15,27]
LIIW	(20), (21)	, $(n=1,2)\,\overline{c}_n = b_n R_n + d_n$, $b_1 = b_2 = 1.5$ $d_1 = d_2 = 0.5$	[25]
CF	(22)	, $c_1 = c_2 = 2.05$ $\chi = 0.729$	[15,26]

Initial values of particles and their velocities were obtained drawing random numbers within the range of each dimension [28], i.e.

$$x_{ij}^1 = x_{\text{LB},j} + R.(x_{\text{UB},j} - x_{\text{LB},j}),\ v_{ij}^1 = x_{\text{LB},j} + R.(x_{\text{UB},j} - x_{\text{LB},j}) \tag{23}$$

where $x_{\text{LB},j}$ and $x_{\text{UB},j}$ are, respectively, lower and upper boundary values of the j^{th} dimension, and $R \in U(0,1)$. Besides, if a particle moves beyond ranges, then it is bounced back to the search space by the following equations:

$$\begin{aligned} x_{ij}^k &= x_{\text{UB},j} - R.(x_{ij}^k - x_{\text{UB},j}), \quad \text{if} \quad x_{ij}^k > x_{\text{UB},j} \\ x_{ij}^k &= x_{\text{LB},j} + R.(x_{\text{LB},j} - x_{ij}^k), \quad \text{if} \quad x_{ij}^k < x_{\text{LB},j} \end{aligned} \tag{24}$$

Analysis results are shown in Table 3. It is obvious that all three algorithms produce accurate results in comparison with the reference values. CF and LIIW have close best values, and they are better than LDIW. Considering the runs in which the best results are obtained, visual comparison is given in Figure 4 for the Rosenbrock function. It is clear that convergence speeds of LIIW and CF are similar, and better than that of LDIW.

Table 2. Benchmark functions and running parameters of the algorithms.

Function	Dimension (m)	Initial Range	Objective
$f_1 = \sum_{i=1}^{m-1}\left(100(x_i^2 - x_{i+1})^2 + (1-x_i)^2\right)$	30	±10	0
$f_2 = \sum_{i=1}^{m} x_i^2$	30	±20	0
N(swarm size) = 20, $K_{\max}$ (max. number of iterations) = 2000 Number of runs for each algorithm: 20			

Table 3. Optimization results for the two benchmark functions.

Method	Function	Worst	Mean	Best	σ
LDIW	f_1	309.4288	73.4422	1.9396	71.7047
	f_2	1.0455E-7	1.2658E-8	6.6664E-11	3.2352E-8
LIIW	f_1	149.2786	49.4933	**0.2627**	**39.0609**
	f_2	3.7972E-11	1.0859E-11	4.1E-14	1.4205E-11
CF	f_1	153.0790	44.2481	0.2733	39.4564
	f_2	0	0	**0**	**0**

[26]. σ: standard deviation. $f_2 = 0$, $f_1 = 39.1185$ Reference results: Best results are typed in bold.

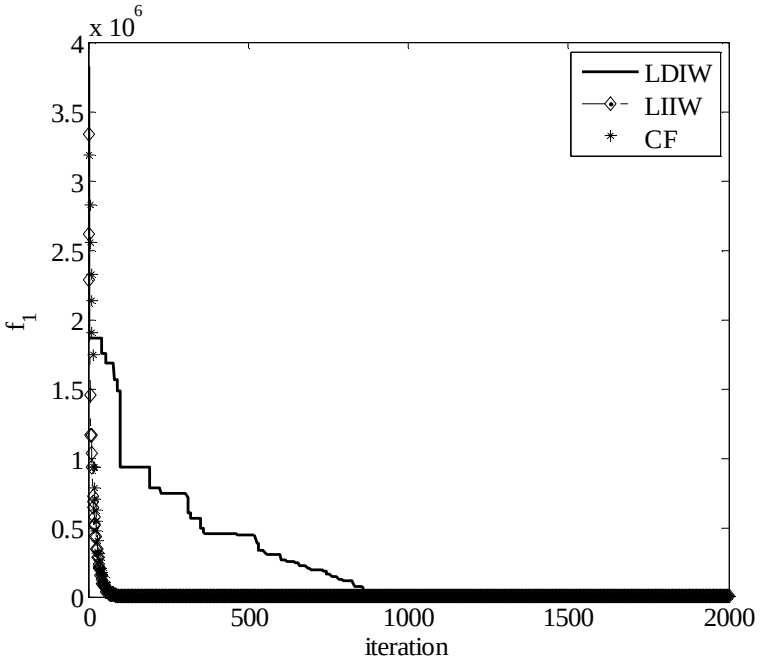


Figure 4: Minimization of f_1 by the three PSO variants.

4. Damage Identification Performance of the PSO Algorithms

To evaluate damage identification performance of the algorithms several damage scenarios in Table 4 were considered.

Table 4. Damage scenarios to evaluate the PSO algorithms.

Case	Damage Parameters	(m/s) V_p	(%) N_p	N	(%) e_p
1	$\bar{z}_1 = 0.5,\ \delta_1$=0.5	1	1	20	2
2	$\bar{z}_1 = 0.33,\ \delta_1$=0.3	3	2	20	2
	$\bar{z}_2 = 0.66,\ \delta_2$=0.2				
3	$\bar{z}_1 = 0.2,\ \delta_1$=0.15	5	3	20, 30	10
	$\bar{z}_2 = 0.4,\ \delta_2$=0.15				
	$\bar{z}_3 = 0.6,\ \delta_3$=0.15				
	$\bar{z}_4 = 0.8,\ \delta_4$=0.15				

Damage cases were arranged in increasing difficulty, i.e. as the damage case index rises, number of cracks, moving load speed, and noise amount increase whereas crack depths decrease. Noise is added to the calculated data as follows [7,9]:

$$\bar{y}(z,t)_{\text{noisy}} = \bar{y}(z,t)_{\text{calc}} + N_p.G.\sigma \tag{25}$$

where $\bar{y}(z,t)_{\text{calc}}$ is the calculated response of point *z* of the damaged beam, i.e. $\bar{y}$ in Eq.(14), N_p is the noise percentage, *G* is Gaussian distribution with zero mean and unit standard deviation, σ is the standard deviation of $y(z,t)_{\text{calc}}$. The data to obtain Figure 2 is used again. The sampling frequency is the same as in [7], i.e. 100 Hz, for all cases. Thus, the more the moving load speed is, the shorter the length of recorded data. Empirically, swarm size between 20 and 30 is sufficient for most benchmark problems [29]. Thus, considering the dimensions of the present problem, the swarm size is kept within this interval. To evaluate the success of the algorithms, the success index (SI), defined in Eq.(26), is employed.

$$\text{SI}=\frac{\text{NSR}}{\text{TNR}} \tag{26}$$

where NSR means number of successful runs, while TNR is the total number of runs of an algorithm. Since each algorithm is run ten times [18] TNR = 10. Successfulness of single run of algorithm is determined in the following way: Suppose that an optimization problem has *m* variables, and their optimum values are α_i, i=1,2,…,*m*. If predicted value $\bar{\alpha}_i$, which is found by solving the optimization problem, satisfies

$$100\left|\frac{(\alpha_i - \bar{\alpha}_i)}{\alpha_i}\right| \le e_p \tag{27}$$

for all $i \in [1,m]$, where e_p denotes the error percentage, then we can conclude that this run of the algorithm is successful. SI indicates the robustness of algorithm. If SI=1 the algorithm is independent of run times, so that it can be called "robust". If SI=0 the algorithm fails to give acceptable results, thus it is unsuccessful. Here $e_p = 2$ is assumed for noise up to 2%, which means a predicted damage parameter is allowable if it has at most 2% relative error. In the case of more noise and less crack size, a bigger value such as $e_p = 10$ is adopted by experience, since noise is much more dominant in this case. With these explanations in mind looking at Table 5 we see that each algorithm

demonstrates excellent performance for K_{max}=30. However, when we reduce K_{max} to 20, a significant performance loss in LDIW is observed in the sense of SI. The other two still have sufficient robustness. Thus, for this simple case either CF or LIIW (preferably LIIW) should be employed to achieve less number of function evaluations, which is equal to the product of swarm size and K_{max}, and faster convergence. Figure 5 verifies this conclusion, as well. Similar observations are valid for Case 2 (see Table 6 and Figure 6) when $K_{max} = 100$. However, when it is reduced to 50, LDIW fails to give results within error limits, thus SI=0. Again there is no remarkable difference between LIIW and CF. Table 7 demonstrates the results for Case 3. It is observed that the swarm size equal to 20 seems insufficient to determine both damage locations and extents. The reason may be due to the number of constrain equations. Because, as the crack number increases so does the number of constrains for crack locations. This may reduce the number of particles inside the feasible region. As a remedy a bigger swarm is tried, since the search ability of algorithm in the earlier stages of optimization is crucial to converge to desired optimum. Besides, number of iterations is increased from 100 to 150 to give the algorithms more chance to exploit local search. In this case, it is observed that only CF, though its SI value is not close to 1, achieved to find desired local optimum within error limits.

Table 5. Results for Case 1 in Table 4.

K_{max}=30							
		LDIW		LIIW		CF	
	E	Pr	|ε|	Pr	|ε|	Pr	|ε|
δ_1	0.5	0.5002	0.03	0.5001	0.02	**0.5001**	0.02
$\overline{z}_1$	0.5	0.5004	0.07	0.5003	0.05	**0.5000**	0.01
f	0.4805	0.4786	0.40	0.4776	0.60	**0.4774**	0.64
SI		1		1		1	

K_{max}=20							
	E	Pr	|ε|	Pr	|ε|	Pr	|ε|
δ_1	0.5	0.5018	0.36	0.4994	0.14	**0.4995**	0.10
$\overline{z}_1$	0.5	0.5047	0.94	**0.4988**	0.26	0.5018	0.36
f	0.4805	0.6495	35	**0.5080**	5.72	0.5374	11.84
SI		0.5		**1**		0.7	

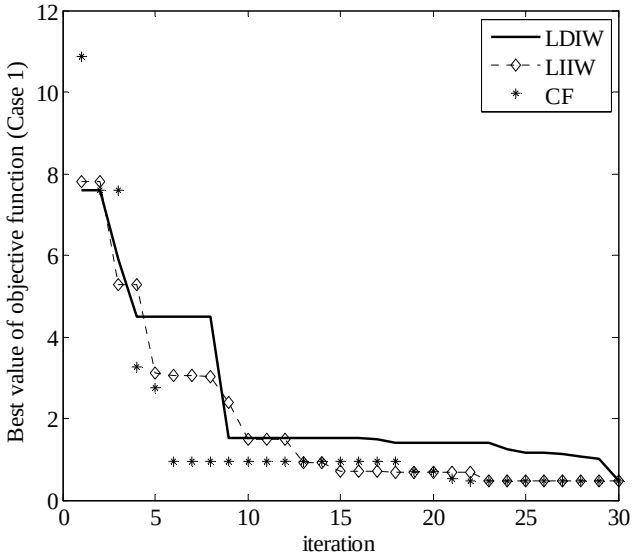


Figure 5: Comparison of the PSO algorithms for Case 1 in Table 4.

With CF both damage sizes and locations are determined with sufficient accuracy. Hence, considering the results in Table 5-7 we can conclude that CF is suitable for the optimization problem considered in this study.

Table 6. Results for Case 2 in Table 4

		K_{max}=100					
		LDIW		LIIW		CF	
	E	Pr	|ε|	Pr	|ε|	Pr	|ε|
δ_1	0.3	**0.3005**	0.18	0.3006	0.20	0.3006	0.20
δ_2	0.2	**0.2000**	0.02	**0.2000**	0.02	**0.2000**	0.02
$\overline{z}_1$	0.33	0.3294	0.17	0.3294	0.18	**0.3293**	0.21
$\overline{z}_2$	0.66	**0.6618**	0.27	0.6619	0.29	0.6619	0.29
f	0.3199	**0.3191**	0.25	**0.3191**	0.25	**0.3191**	0.25
SI		0.8		0.7		**0.9**	

		K_{max}=50					
		LDIW		LIIW		CF	
	E	Pr	|ε|	Pr	|ε|	Pr	|ε|
δ_1	0.3	0.2972	0.9343	0.3006	0.20	**0.3003**	0.11
δ_2	0.2	0.1967	1.6300	0.1990	0.50	**0.1996**	0.21
$\overline{z}_1$	0.33	0.3289	0.3321	0.3295	0.15	**0.3301**	0.04
$\overline{z}_2$	0.66	0.6406	2.9412	**0.6604**	0.05	**0.6604**	0.06
f	0.3199	0.3214	0.47	**0.3191**	0.25	**0.3191**	0.25
SI		0		**0.6**		**0.6**	

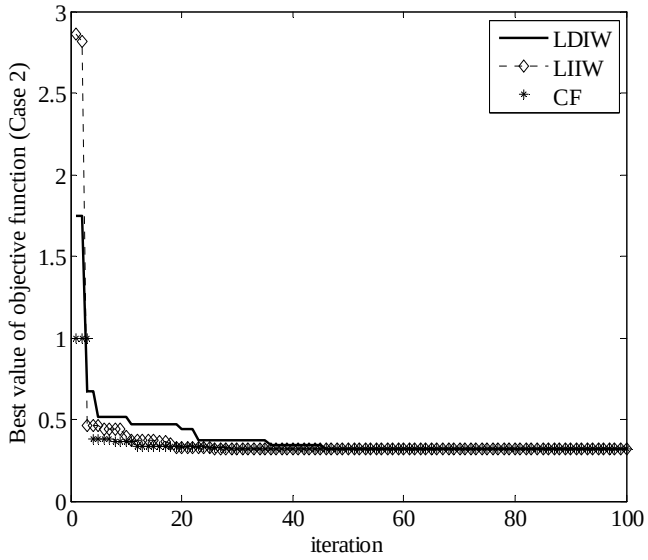


Figure 6: Comparison of the PSO algorithms for Case 2 in Table 4.

Table 7. Results for Case 3 in Table 4.

$e_p = 10$ Swarm size N=20, K_{max}=100,							
		LDIW		LIIW		CF	
	E	Pr	|ε|	Pr	|ε|	Pr	|ε|
δ_1	0.15	**0.0821**	45.27	0.0244	83.75	0.0670	55.36
δ_2	0.15	**0.1824**	21.59	0.1842	22.78	0.1827	21.83
δ_3	0.15	**0.1506**	0.39	0.1367	8.86	0.1329	11.42
δ_4	0.15	**0.1809**	20.57	0.2163	44.19	0.2022	34.77
$\bar{z}_1$	0.2	0.1731	13.43	0.2262	13.08	**0.2022**	1.09
$\bar{z}_2$	0.4	0.3812	4.70	0.3867	3.33	**0.3877**	3.07
$\bar{z}_3$	0.6	0.6289	4.82	0.6078	1.30	**0.6069**	1.15
$\bar{z}_4$	0.8	0.8616	7.70	0.8612	7.65	**0.8494**	6.17
f	0.2894	0.1731	40	0.2760	4.63	**0.2758**	4.70
SI		0		0		0	
$e_p = 10$ Swarm size 30, K_{max}=150,							
		LDIW		LIIW		CF	
	E	Pr	|ε|	Pr	|ε|	Pr	|ε|
δ_1	0.15	0.1656	10.42	**0.1599**	6.61	0.1632	8.83
δ_2	0.15	0.1329	11.39	0.1613	7.55	**0.1423**	5.15
δ_3	0.15	0.1683	12.22	0.1289	14.09	**0.1502**	0.11
δ_4	0.15	0.1890	25.98	0.1666	11.08	**0.1539**	2.59
$\bar{z}_1$	0.2	**0.1977**	1.13	0.1750	12.52	0.1960	2.00
$\bar{z}_2$	0.4	**0.3974**	0.64	0.4103	2.55	0.4056	1.41
$\bar{z}_3$	0.6	**0.5999**	0.02	0.5998	0.04	0.5947	0.89
$\bar{z}_4$	0.8	0.8792	9.90	0.7936	0.80	**0.7995**	0.07
f	0.2894	0.2795	3.42	**0.2796**	3.39	0.2794	3.46
SI		0		0		**0.2**	

5. Comparison With The Wavelet Transform Method

Common deficiency of the wavelet transform methods in [7-11] is that CWT coefficients lose sensitivity to damage with increasing moving load speed. To indicate this the cases in Table 8 are considered. Figure 7 demonstrates the wavelet coefficients of the midpoint deflection corresponding to Case 1 in Table 8. As is obviously seen, CWT coefficients are sufficiently sensitive to damage locations, i.e. the CWT coefficients at the crack points have large magnitudes and these magnitudes increase along with the scale. Therefore, it is possible in this case to determine crack locations and sizes with sufficient accuracy using CWT coefficients.

Table 8. Damage cases for comparison with the wavelet transform method.

Case	(m/s) V_{p}	(Hz) ω_s	Data length	(%) N_{p}
1	1	100	5000	3
2	5	100	1000	3
3	5	500	5000	3
$\bar{z}_1 = 0.33,\ \bar{z}_2 = 0.67,\ \delta_1 = \delta_2 = 0.5$ Damage parameters for each case:				
$\left(L/V_p\right)\omega_s$: sampling frequency, Data length $\cong\ \omega_s$				

However, when the moving load speed is increased to 5 m/s, which corresponds to Case 2 in Table 8, CWT coefficients fail to reveal damage effects, as seen in Figure 8. One can verify that a similar variation in Figure 8 is obtained for the Case 3 in Table 8. Thus, a bigger sampling frequency is not helpful in extracting damage locations. On the other hand, Table 9 demonstrates that both damage locations and crack sizes can be determined by the proposed approach with sufficient accuracy. With increasing moving load speed, the displacement curves become smoother, thus singularities in the response curve disappear. This reduces the sensitivity of CWT coefficients to damage. However, the proposed approach is based on the correlation of reference displacement curves, which are measured in the damaged case, and those calculated by the mathematical model of the structure. Thus, disappearing of singularities in the response curve does not affect the performance of the proposed method.

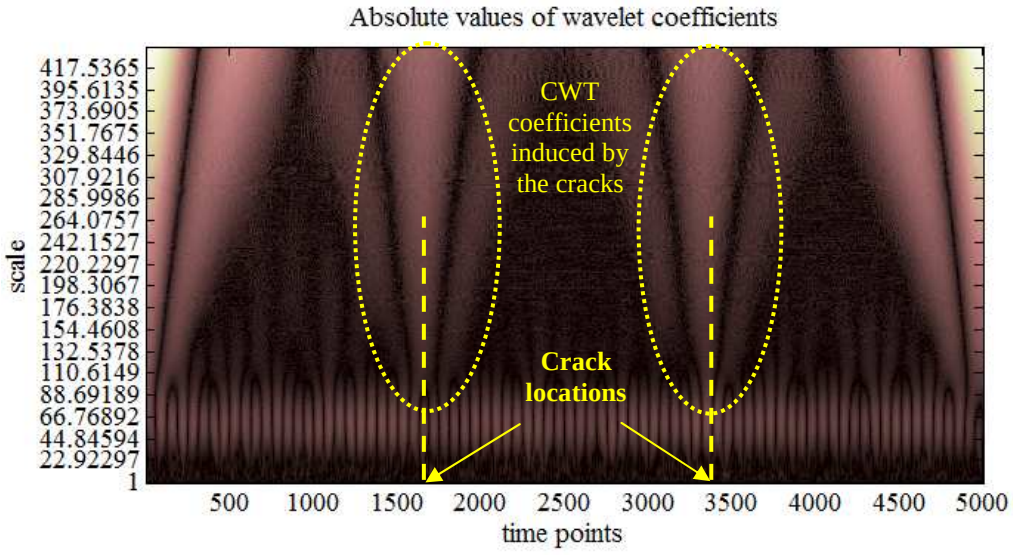


Figure 7: CWT coefficients of the midpoint displacement (Case 1 in Table 8). Wavelet: Gaus4

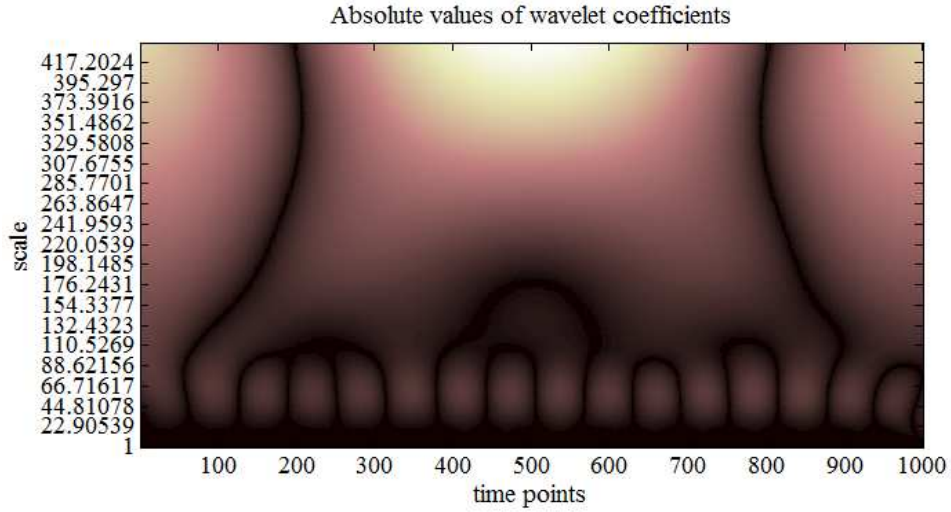


Figure 8: CWT coefficients of the midpoint displacement (Case 2 in Table 8).

6. Conclusions

In this work, DD in a beam subject to moving load is formulated as an inverse problem, and damage parameters are determined by solving this problem using PSO variants. The applicability of the proposed method to real-life problems depends on three issues: 1) Well-defined mathematical model of the system, 2) Suitable crack or damage model for the carrying structure, 3) Measuring reference data (i.e. $\bar{y}$ in Eq.(14)) with sufficient quality. These three requirements are characteristics of the general MU based DD methods, and they are related to different disciplines such as dynamics of structures and systems, fracture mechanics, data acquisition and signal processing. On the other hand, considering the above studies the following conclusions can be drawn:

Table 9. Optimization results for the cases in Table 8.

		f	δ_1	δ_2	$\bar{z}_1$	$\bar{z}_2$
Case 1	E	1.4579	0.5	0.5	0.33	0.67
	Pr	1.5034	0.4988	0.5022	0.3314	0.6628
Case 2	E	0.2860	0.5	0.5	0.33	0.67
	Pr	0.3443	0.4954	0.5007	0.3270	0.6452
Case 3	E	0.2808	0.5	0.5	0.33	0.67
	Pr	0.3191	0.5047	0.5001	0.3251	0.6670
Swarm size 20, K_{max}=30, E: Exact, Pr: Predicted, Optimization algorithm: CF						

1) Performances of the three PSO algorithms were compared. It was observed that LIIW and CF methods are superior to LDIW in the sense of convergence speed and robustness. LIIW and CF were nearly the same for most of the cases considered. However, CF was observed to be slightly better than LIIW from the point of robustness. Also, swarm size of 30 and the maximum number of iterations equal to 150 were experienced to be sufficient for the considered cases. On the other hand, for the inertia weight, cognitive and acceleration constants there are other combinations in the

relevant literature, though the ones employed in this study are widely used. Trying these combinations may be worth further studying in another study.

2) It was observed that small damages, i.e. crack depth ratio equal to 0.15, can be determined by the proposed method in spite of three percent noise interference. Also, this method can be applied for higher moving load velocities which the CWT coefficients fail to extract damage info. However, the proposed method requires an accurate mathematical or finite element model of the structure, which is a common disadvantage of the MU based DD methods. In this respect, the wavelet transform methods are superior as they generally do not require this. However, their sensitivity to damage depends on various parameters, as is well known from the relevant literature.

3) In this work, the inertia effects of the moving load and load-beam interaction are omitted. Indeed, this does not bring a serious restriction on the application of the proposed method, since the method is based on the correlation between reference and calculated responses. Including these effects may enhance the performance of the proposed method, since their impact on the response curve is considerable especially at higher velocities.

4) Denoising the reference displacements in Eq.(14) may enhance the accuracy of crack identification. In this regard suitable filtering methods can be applied before solving the optimization problem.

Future works are planned to cover the omitted issues above and verification with real data.

References

[1] Frýba, L., *Vibration of Solids and Structures Under Moving Loads*, Telford, 1999.
[2] Mahmoud, M.A., Effect of cracks on the dynamic response of a simple beam subject to a moving load. *Proc. of the Inst. of Mech. Eng*., 215, 207-215, 2001.
[3] Mahmoud, M.A., Abou Zaid, M.A., Dynamic response of a beam with a crack subject to a moving load. *J. of Sound and Vib.*, 256(4), 591-603, 2002.
[4] Bilello, C., Bergman, L.A., Vibration of damaged beam under a moving mass: theory and experimental validation. *J. of Sound and Vib.*, 274, 567-582, 2004.
[5] Lin, H.P., Chang, S.C., Forced responses of cracked cantilever beams subjected to a concentrated moving load. *Int. J. of Mech. Sci.*, 48, 1456-1463, 2006.
[6] Ariaei, A., Ziaei-Rad, S., Ghayour, M., Vibration analysis of beams with open and breathing cracks subjected to moving masses. *J. of Sound and Vib*., 326, 709-724, 2009.
[7] Zhu, X.Q., Law, S.S., Wavelet-based crack identification of bridge beam from operational deflection time history. *Int. J. of Solids and Struct.*, 43, 2299-2317, 2006.
[8] Hester, D., Gonzalez, A., A wavelet-based damage detection algorithm based on bridge acceleration response to a vehicle.*Mech. Syst. and Signal Process*. doi:10.1016/j.ymssp.2011.06.007.
[9] Nguyen, K.V., Tran, H.T., Multi-cracks detection of a beam-like structure based on the on-vehicle vibration signal and wavelet analysis. *J. of Sound and Vib*., 329, 4455-4465, 2010.
[10] Khorram, A., Bakhtiari-Nejad, F., Rezaeian, M., Comparison studies between two wavelet based crack detection methods of a beam subjected to a moving load. *Int. J. of Eng. Sci*., 51, 204-215, 2012.
[11] Gökdağ, H., Wavelet-based damage detection method for a beam-type structure carrying moving mass. *Struct. Eng. and Mech*. 38(1), 81-97, 2011.
[12] Doebling, S.W., Farrar, C.R., Prime, M.B., Shevitz, D.W., Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review. Rep. LA-13070- MS, UC-900, Los Alamos National Laboratory, USA, (1996).

[13] Biskas, P.N., Ziogos, N.P., Tellidou, A., Zoumas, C.E., Bakirtzis, A.G., Petridis, V., Comparison of two metaheuristics with mathematical programming methods for the solution of OPF. *Proc. of the 13th IEEE Conf. on Intell. Syst. Appl. to Power Syst*., doi:10.1109/ISAP.2005.1599316.
[14] Kennedy, J., Eberhart, R., Particle swarm optimization. *Proc. of the 4th IEEE Int. Conf. on Neural Netw*. 4, 1942-1948, 1995.
[15] Parsopoulos, K.E., Vrahatis, M.N., *Particle swarm optimization and intelligence: Advances and Applications*. IGI Global, 2010.
[16] http://www.swarmintelligence.org/tutorials.php (the last access date: 30.08.2012)
[17] Begambre, O., Laier, J.E., A hybrid particle swarm optimization–simplex algorithm (PSOS) for structural damage identification. *Adv. in Eng. Softw*. 40, 883-891, 2009.
[18] Moradi, S., Razi, P., Fatahi, L., On the application of bees algorithm to the problem of crack detection. *Comput. and Struct*. 89, 2169-2175, 2011.
[19] Seyedpoor, S.M., A two stage method for structural damage detection using a modal strain energy based index and particle swarm optimization. *Int. J. of Non-Linear Mech*. 47, 1-8, 2012.
[20] Buezas, F.S., Rosales, M.B., Filipich, C.P., Damage detection with genetic algorithms taking into account a crack contact model. Eng. Fract. Mech., doi:10.1016/j.engfracmech.2010.11.008
[21] Clough, R.W., Penzien, J., *Dynamics of Structures*. Comput. & Struct., Inc., 1995.
[22] Shi, Y., Eberhart, R. C., A modified particle swarm optimizer. *IEEE World Congr. on Comput. Intell*., doi:10.1109/ICEC.1998.699146.
[23] Chuan, L., Quanyuan, F., The standard particle swarm optimization algorithm convergence analysis and parameter selection. *3rd IEEE Int. Conf. on Nat. Comput*. 3, 823-826, 2007.
[24] Shi, Y., Eberhart, R. C., Empirical study of particle swarm optimization. *Proc. of Congr. on Evol. Comput*. 3, 1945-1949, 1999.
[25] Zheng, Y-L., Ma, L-H., Zhang, L-Y., Qian, J-X., On the convergence analysis and parameter selection in particle swarm optimization. *Proc. of the 2nd Int. Conf. on Mach. Learn. and Cybern*. 3, 1802-1807, 2003.
[26] Clerc, M., Kennedy, J., The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. on Evol. Comput*. 6(1), 58-73, 2002.
[27] Yamaguchi, T., Yasuda, K., Adaptive particle swarm optimization; self-coordinating mechanism with updating information. *IEEE Int. Conf. on Syst., Man, and Cybern*., 3, 2303-2308, 2006.
[28] Trelea, I. C., The particle swarm optimization algorithm: convergence analysis and parameter selection. *Inf. Process. Lett*., 85, 317-325, 2003.
[29] Clerc, M., *Particle swarm optimization*. ISTE Ltd, 2006.