Hierarchical Damage Detection Using Power Spectral Density Methods

R. K. GILES[1] AND B. F. SPENCER, JR.[2]

ABSTRACT

Structural health monitoring offers the opportunity to revolutionize the maintenance and management of civil infrastructure. To achieve the full potential of this technology, dense arrays of multi-scale sensors are expected to be required. However, a densely distributed sensor network deployed on a large-scale civil engineering structure will produce excessive amounts of data that need to be processed. Using a hierarchical computing framework that organizes the sensors into groups and processes the data locally, will greatly reduce the data that needs to be broadcast, as well as the associated power consumed, and the time it takes to get results. This paper proposes damage detection algorithms based on changes in the structure's Power Spectral Density (PSD) using a hierarchical, distributed computing approach. The proposed algorithms are model-independent and require only output measurement, making them appropriate for use in full-scale applications employing ambient vibration. Numerical studies are presented for several cases for multiple damage sites with varying hierarchical organization. The results show that the proposed hierarchical computing algorithm is effective for detecting damage.

INTRODUCTION

Structural Health Monitoring (SHM) allows for continuous monitoring of the condition and performance of civil infrastructure. This ability is increasingly important as natural and engineering elements continue to adversely affect the performance and capacity of ageing civil infrastructure. The continual ageing and decay processes of the infrastructure requires heavy investments in inspections and repairs to ensure the public's safety in using the infrastructure. An installed SHM system offers the possibility for increased confidence in the safety of the structure, as well as reduced inspection and repair costs [1, 2]. Of course, the effectiveness of a SHM system will depend on the type of sensor network installed and the damage detection algorithm employed.

Two types of sensor networks are available or are being developed: wired and wireless networks. With a traditional wired sensor network, a given bridge would typically require wire many times its length to densely instrument it for SHM. These wires would all lead to a base station where the system is controlled, the damage detection algorithm is implemented, and the data is stored. Such a centralized approach has little redundancy; damage to the central computing base station or the wires from the sensors could easily

[1] Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 N. Mathews Ave., Urbana, IL 61801, U.S.A.
E-mail: rgiles2@uiuc.edu

[2] Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 N. Mathews Ave., Urbana, IL 61801, U.S.A.
E-mail: bfs@uiuc.edu

render the system ineffective. Simple wireless sensor networks have similar problems. Alternatively, a smart sensor network where each sensor node has an on-board microprocessor in addition to being wireless can provide a more fault-tolerant approach to SHM, among other advantages. However, such a network also presents its own unique challenges. A principal task is to develop effective algorithms that can be deployed in the distributed computing environment intrinsic to smart sensor networks. Such algorithms require that damage detection be implemented in a hierarchical manner where information is only shared locally, and computation occurs on the sensors themselves [3, 4].

This paper proposes a model-independent, hierarchical, damage detection algorithm that can be implemented in a distributed computing environment. Several numerical examples are presented to explore the effects of group size, multiple damage locations, and group overlap on the algorithm.

POWER SPECTRAL DENSITY METHODS

The proposed approach is an extension of the work of Beskhyroun et al. (2004, 2005) which examined the changes in the Power Spectral Density (PSD) of the structure induced by damage [5, 6, 7]. For completeness, this work is summarized in this section.

The theory behind the PSD damage detection algorithms holds that changes in the operational mode shapes, or changes in their curvatures, can be attributed to changes in the structure due to damage. Beskhyroun et al. (2005, 2006) presented two variations on the PSD methods. Both the Absolute Difference PSD (ADPSD) Method and the Curvature Difference PSD (CDPSD) Method have the same initial calculations. The first step is to calculate the PSD, $G_i(f)$, via

$$G_i(f) = \frac{2}{T} E\left[\left|X(f)^2\right|\right] \tag{1}$$

where $X(f)$ is the Fourier transform of the measured acceleration, f is frequency in Hertz, $E[\cdot]$ is the expected value operator, T is the measured record length, and i is the sensor node number. Subsequently, the two algorithms normalize the PSD using the sum of the squares of the PSD values evaluated at each frequency

$$P(f) = G(f) \Big/ \sqrt{\sum_{i=1}^{n} G_i^2(f)} \tag{2}$$

where $P(f)$ is the normalized PSD and n is the number of nodes. Normalizing the PSD serves a dual purpose. First, it allows for the use of different excitation events in the analysis. Second, it equalizes the importance of each mode shape by effectively eliminating the modal contribution factors. Thus, a change in the first and subsequent mode shapes has an equal influence in determining where damage has occurred. This normalization of the modal contributions suggests that the more modes shapes included in the analysis, the more accurate the algorithm will become.

Absolute Difference PSD Method

After normalization, the ADPSD and the CDPSD have divergent calculation paths. The ADPSD immediately determines the change in the normalized PSD, $\Delta P_i(f)$, using the undamaged normalized PSD, $P_i^u(f)$, and the damaged normalized PSD, $P_i^d(f)$ via

$$\Delta P_i(f) = \left|P_i^u(f) - P_i^d(f)\right| \tag{3}$$

Thresholding then eliminates all but the maximum change in the normalized PSD value for each frequency. Thus, the matrix ΔP^{MAX}, representing the maximum change in the normalized PSDs, has a single non-zero value for each frequency but may have many non-zero values for each node, to wit

$$\Delta P^{MAX} = \begin{bmatrix} 0 & \Delta P^{MAX}(f_2) & \cdots & 0 \\ \Delta P^{MAX}(f_1) & 0 & \cdots & \Delta P^{MAX}(f_m) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \qquad C^{\Delta P} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 1 & 0 & \vdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{4}$$

where *m* is the frequency index. At the same time as the thresholding, a counting matrix, $C^{\Delta P}$, is established by replacing each non-zero value in ΔP^{MAX} with a 1. These matrices are then summed for each node to create a vector with length equal to the group size. After computing the standard deviation for each vector, twice the respective standard deviation is subtracted from each value to form a normalized nodal damage quality vector, $\hat{P}$, and a normalized nodal damage quantity vector, $\hat{C}$, where all negative values are eliminated via

$$\hat{P}_i = \sum_{f=f_1}^{f_m} \Delta P_i^{MAX}(f) - 2\sigma_P \qquad \hat{C}_i = \sum_{f=f_1}^{f_m} C_i^{\Delta P}(f) - 2\sigma_C \tag{5}$$

where f_m is the last frequency in the PSD, σ_P is the standard deviation of ΔP^{MAX}, and σ_C is the standard deviation of $C^{\Delta P}$. The Accumulated Damage Index (ADI), D, is then computed by multiplying the nodal values of the $\hat{P}$ and $\hat{C}$ values together.

$$D_i = \hat{P}_i \hat{C}_i \tag{6}$$

Curvature Difference Method

After the normalization in (2), the CDPSD fits the normalized PSD, $P(f)$, to a series of cubic splines with periodic end conditions for each frequency. This step allows the second derivative of the PSD – the curvature of the operational mode shapes – to be determined. After computing the undamaged and damaged curvatures, they are used to compute $\Delta P''(f)$, the change in the curvature of the normalized PSD via

$$\Delta P_i''(f) = \left| P_i^{u''}(f) - P_i^{d''}(f) \right| \tag{7}$$

where $P^{u''}$ is the undamaged normalized PSD curvature and $P^{d''}$ is the damaged normalized PSD curvature. Before thresholding, $\Delta P''(f)$ is normalized as a sample population using the mean and standard deviation of the sample, to wit

$$\tilde{P}_i(f) = \frac{\Delta P_i(f) - \overline{\Delta P}(f)}{\sigma_P(f)} \tag{8}$$

where $\tilde{P}(f)$ is the renormalized PSD curvature difference matrix, $\overline{\Delta P}(f)$ is the mean of nodal values for each frequency, and $\sigma_P(f)$ is the standard deviation for each frequency. $\tilde{P}_i(f)$ is then thresholded and counted by eliminating all values that are less than a percentage, α, of the maximum $\tilde{P}_i(f)$ value, $\tilde{P}_{\max}(f)$. A counting matrix, $C(f)$, is also formed at this point as in (4).

$$\begin{aligned} &\text{if } \left|\tilde{P}_i(f)\right| < \left|(\alpha * \tilde{P}_{\max})\right| \text{ then let } \tilde{P}_i(f) = 0 \text{ and } \mathrm{C}_i(f) = 0 \\ &\text{if } \left|\tilde{P}_i(f)\right| \geq \left|(\alpha * \tilde{P}_{\max})\right| \text{ then let } \tilde{P}_i(f) = \tilde{P}_i(f) \text{ and } \mathrm{C}_i(f) = 1 \end{aligned} \tag{9}$$

In this paper, α was set to 57.5% to achieve suitable results. The $\tilde{P}(f)$ and $C(f)$ matrices are then summed for each nodal value to create vectors with length equal to the group size.

$$A_i = \sum_{f=f_1}^{f_m} \left|\tilde{P}_i(f)\right| \qquad O_i = \sum_{f=f_1}^{f_m} C_i(f)$$

(10)

Finally, these values are multiplied together at each node to produce the ADI, D_i.

$$D_i = A_i O_i \qquad (11)$$

Proposed Algorithm

The proposed algorithm is an extension of the ADPSD and CDPSD to accommodate implementation on a network of smart sensors. When the unaltered methods are applied to the structure in the clusters that are the basis of hierarchical computing, the magnitudes of the ADI vary significantly making it difficult to determine whether damage or noise is responsible for the given results. To rectify this difficulty, the ADIs for both the ADPSD and CDPSD are normalized by dividing each D_i from (6) and (11) by the respective sum of D_i for the group. This normalization eliminates the variation in magnitude and allows a single set of criteria to be programmed into each cluster head to detect damage regardless of group size. A single set of damage criteria allows the clusters in the network to be assigned dynamically which increases the robustness and improves the fault tolerance of the smart sensor network. For damage to be indicated in the ADPSD, one condition must be met:

1. $D_{\max} > 0.02$ where $D_{\max}$ is the maximum value excluding the D_i at the end nodes

For the CDPSD, damage is indicated when the following two conditions are satisfied:

1. $D_{\max} > 1.5/n$ where *n* is the length of the group
2. $D_{end} < 1.5/n$ where *n* is the length of the group and D_{end} is either end node.

Note that the end nodes in the cluster are necessarily excluded from the calculations because the end nodes return false positives for damage. The false positives on the end nodes are remnants of the clustering process and its affect on the normalization constants and different virtual boundary conditions for the group.

Furthermore, the proposed algorithm uses the ADPSD and CDPSD in sequence to locate damage in the structure. For the algorithm to give a positive damaged result, both the APSD and CPSD must indicate damage in turn. Doing so takes advantage of their unique characteristics and abilities. The ADPSD requires less calculation as it does not require spline fitting and differentiation. The less intensive damage detection calculations allow the node to return to a less power intensive state. However, the ADPSD often gives a singular strong damage indication on one sensor even when damage is located between two sensors. Therefore, once damage is indicated using the ADPSD, the more calculation intensive CDPSD is used to both confirm the damage and better locate it between two sensors. As a final step, the two normalized ADIs from both algorithms are incorporated into a Combined ADI by averaging the values to give the final indication of damage location. All the calculations are performed locally on the cluster head without need for communication with nodes outside the cluster. However, once damage is detected, confirmed, and located, the message broadcast could consist only of a bit indicating damage and the sensor numbers of the affected nodes. By transmitting information only when damage has been confirmed by both sensors improves the power efficiency of the network and the confidence in the reported results.

NUMERICAL EXAMPLES

To investigate the proposed method, a MATLAB finite element model of a planar truss was used. The truss, illustrated in Figure 1, consists of 53 frame elements arranged in fourteen bays with simple supports. The structure was excited using a band-limited white noise applied in the vertical direction at node 7, and the vertical acceleration at the 13 numbered sensor locations given in Figure 3 was recorded. Different seeds in the random number generator were used in the undamaged and damaged cases to simulate different ambient vibrations. Analysis led to the selection of a cutoff frequency of 190 Hz and 1024 FFT points were used in the calculations.

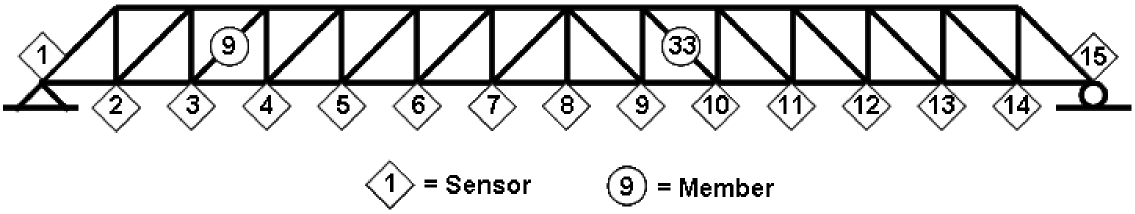

Figure 1: 14-Bay Planar Truss

Damage Cases

Several damage cases were analyzed to test the ability of the method. Damage was simulated by altering the cross-sectional area of the selected member in the MATLAB model. A 5% reduction in cross-sectional area was chosen for the analysis. For each damage case, the size and overlap of the groupings were varied to determine how each affected the performance of the proposed method. In the damage cases, groups are numbered from left to right and include the indicated number of sensors and the overlap indicates how many of the sequential sensors are shared with the previous group. For example, in Case III in Table 1, the first group contains sensors {1 2 3 4} and the second group contains sensors {2 3 4 5}. Table 1 lists all damage cases presented in this paper.

Table 1: Selected Damage Cases

Case	Damage		Grouping		
	Mbr #	% Red	Qty	Size	Overlap
I	9, 33	5%	1	15	0
II	9, 33	5%	3	7	3
III	9, 33	5%	12	4	3

ANALYSIS AND RESULTS

Case I

Case I tested the ability of the method to detect multiple damage sites in a given group. There is only a 5% reduction in the cross-sectional area, and yet both damage sites are detected with only minor false positives as shown in Figure 2. Because these false positives are grouped around the actual damage sites, they are of little consequence and the averaging that leads to the proposed approach eliminates them further. This case also illustrates the differences in the abilities of the ADPSD and the CDPSD. The small amount of damage caused

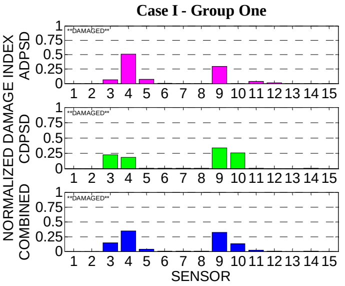

Figure 2: Case I - Group 1

the ADPSD to indicate the presence of damage but not clearly indicate the proper bay that is affected. The CDPSD still is able to indicate the damaged bay. It should also be noted that the normalization that allows for damage pattern recognition in the groups eliminates the potential of the PSD methods to indicate the severity of damage that has occurred. Even though all sensors were used in this case, the results indicate that that the methods would detect multiple damage sites if they all appeared in a group of any size in a distributed computing sensor network.

Case II

In this case, the sensors have been divided into three groups with seven sensors and three overlapping sensors in the group. Effectively this separates the two damage locations into two different groups. As seen in Figure 3, damage is successfully located at the first damage location, element 9, with a 5% cross-sectional area reduction. The damage location appears in the middle of this group and can be thought of as an independent truss with damage near its center. The sensors near the second damage location, element 33, are not included in the group so no information about the second damage location is returned as expected due to the local nature of damage.

Figure 4 shows that in Group 2, the method detected the damage of element 33 with a 5% cross-sectional area reduction. Again, this group does not include the first damage location, and so no information about damage at that site is included in the figure. The usefulness of the CDPSD as a check to the damage state is illustrated here because the ADPSD indicates damage only at sensor 9. This result could indicate that there is damage only on the vertical member attached to sensor 9 or equal damage on either side. Either case should still have an effect on the neighboring sensors due to the asymmetric nature of the bays next to them.

However, Figure 5 shows the problem of using the CPDSD on its own. When the damage locations are near the ends of the group, the methods have a difficult time detecting it. In this case, the ADPSD does not detect damage because the ends are eliminated from the analysis. Eliminating the end values from the analysis is done because the ends always include a large amount of indicated damage as a result of the virtual cut in the truss and the change in end condition assumptions for each group's operational mode shapes. As such, these values are set to zero in the graphs for easier readability of the ADIs. The CDPSD still detects damage between sensors 9 and 10 but the false positives have increased to a significant level. Based on the algorithm's methodology, no damage would be reported in this last group. This result indicates that at least three nodes should overlap in each group so that a damage case in never between the two sensors on the ends of both overlapping groups.

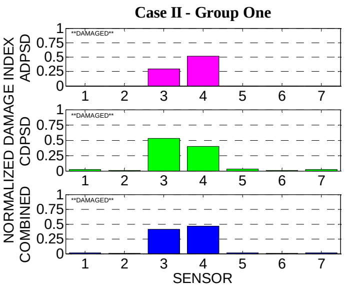

Figure 3: Case II - Group 1

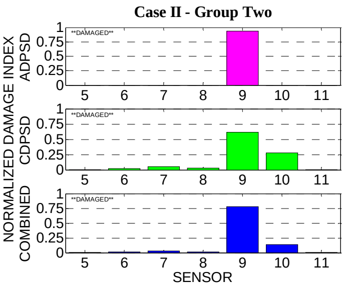

Figure 4: Case II - Group 2

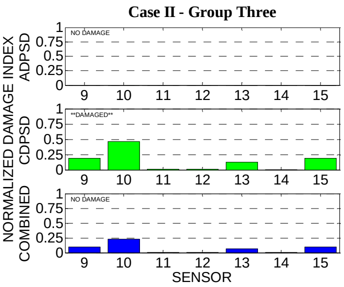

Figure 5: Case II - Group 3

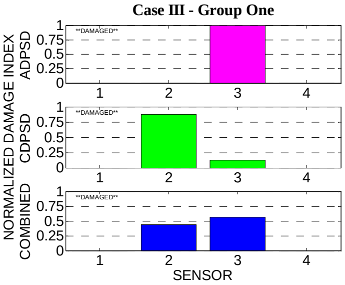

Figure 6: Case III - Group 1

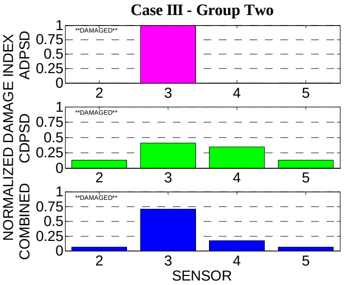

Figure 7: Case III - Group 2

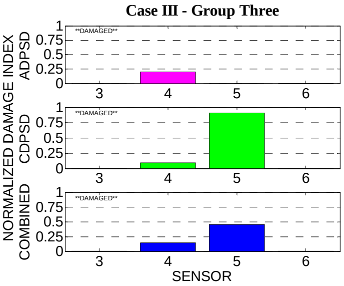

Figure 8: Case III - Group 3

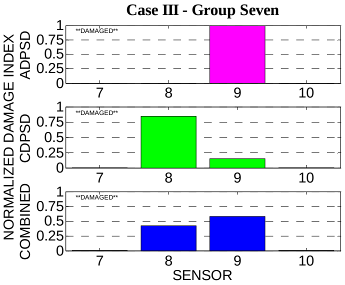

Figure 9: Case III - Group 7

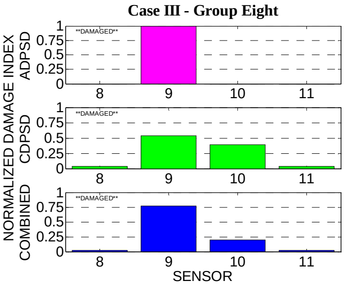

Figure 10: Case III - Group 8

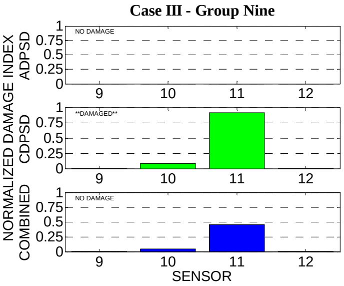

Figure 11: Case III - Group 9

Case III

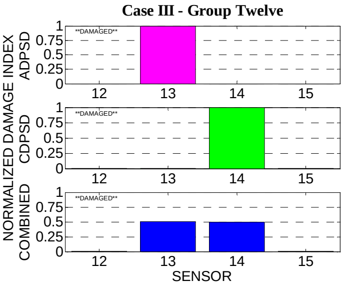

Figure 12: Case III - Group 12

Case III illustrates what happens when the group sizes are too small for proper normalization of the PSD and subsequent spline fitting to determine the derivatives. The method properly indicates damage when the two damage cases are the center two nodes in the group as shown in Figure 7 for damage in element 9 and Figure 10 for damage in element 33. Damage is also indicated in three of four groups (Figures 6, 8, 9) where the nodes around the damaged member are the end nodes of the group. Figure 11 shows group 9 where the nodes around damaged element 33 are on the end of the group but no damage was indicated. When compared with Figures 9 and 10, the strong indication of damage at node 9 is missing because the damage criterion does not allow the maximum to be on the end of the group. Arranging a group where the nodes nearest the damage location are one of the end nodes of the group should be avoided. Preventing this situation is accomplished by properly choosing the group overlap to be at least three nodes, meaning that each pair of sensors will not be an end pair in both overlapping groups. In addition, for Case III note the false positive indicated by both methods in the group 12 as shown in Figure 12. This group is far from either damage location, yet it falsely indicates damage between sensors 13 and 14. The false positive is a result of minimal differences in the small group being amplified through the normalization processes. Increasing the group size ensures that the normalization does not amplify the small changes due to different excitation records when damage is not included in the group. Having a minimum group size of five eliminates the presence of false positives as shown in this case. The proposed algorithm does not report any false positives, even when no damage is present anywhere in the structure, as long as the group size is above the minimum of five.

CONCLUSIONS

The methods introduced by Beskhyroun et al. (2004, 2005) have been ext ended to improve their performance and to facilitate implementation in the distributed computing environment intrinsic to a network of smart sensors. The proposed approach has been shown through numerical examples to be effective in damage detection and localization. Damage at multiple sites with a small damage severity is successfully detected in an arbitrary location. Care has to be taken when establishing the sensing clusters to ensure that there are adequate numbers of sensors and proper overlap to increase the likelihood that damage can be detected properly in all members. Overlap should be a minimum of three sensors and group size should be no smaller than five. Through the normalization of the ADI and the use of a universal damage criterion, programs to determine clusters dynamically can be included to maintain the appropriate size and overlap should a sensor malfunction without the need to alter the damage detection algorithm. Thus the fault tolerance of the smart sensor network is improved by using the proposed algorithm by allowing for the dynamic cluster assignments. Another advantage to the proposed method is that large acceleration records do not have to be transmitted within the network. By computing the PSD on the local sensor, only it needs to be transmitted to the cluster head for additional calculation. The PSD represents an aggregated and compressed form of the structural information derived from the acceleration record. This affects not only the transmission speed and therefore power consumption, but it also reduces the data that needs to be stored on the sensor node. By reducing power consumption and operating effectively in a distributed computing environment, the proposed combined PSD method will be useful in SHM applications. Further analytical and physical experiments are underway to prove confirm the results presented heretofore.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the research by the National Science Foundation under NSF grants CMS 03-01140 and CMS 06-00433, Dr. S.C. Liu, Program Manager. The first author would also like to acknowledge the support of NSF's Graduate Research Fellowship Program.

REFERENCES

1. Sohn H, Farrar CR, Hermez FM, Shunk DD, Stinemates DW, Nadler BR. “A Review of Structural Health Monitoring Literature: 1996-2001.” *Los Alamos National Laboratory Report LA-13976-MS*, 2003.
2. Doebling SW, Farrar CR, Prime MB, Shevitz DW. “Damage Identification and Health Monitoring of Structural and Mechanical Systems from Changes in their Vibration Characteristics: a Literature Review.” *Los Alamos National Laboratory Report LA-13070-MS*. 1996
3. Gao Y, Spencer BF Jr, Ruiz-Sandoval M. “Distributed Computing Strategy for Structural Health Monitoring.” Structural *Control and Health Monitoring*. 2006 13:488-507
4. Nagayama T, Rice JA, Spencer BF, Jr. “Efficacy of Intel's Imote2 Wireless Sensor Platform for Structural Health Monitoring Applications” APWSHM – Dec 2006
5. Beskhyroun S, Oshima T, Mikami S. “Structural Health Monitoring of Bridges Based on Vibration Measurements.” International Workshop on Modern Science and Technology, Kitami, Japan , September 2-3, 2004
6. Beskhyroun S, Oshima T, Mikami S, Yamazaki T. “A Numerical Analysis of Structural Damage Detection Using Changes in the Curvature of Power Spectral Density.” *Journal of Structural Engineering*. March 2005 vol. 51A
7. Beskhyroun S, Oshima T, Mikami S, Tsubota Y. “Structural Damage Identification Algorithm Based on Changes in Power Spectral Density.” *Journal of Applied Mechanics*. August 2005 vol. 8