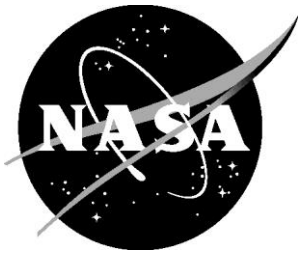
NASA/TP–2016-219186



Strain-Based Damage Determination Using Finite Element Analysis for Structual Health Management

Jacob D. Hochhalter and Thiagaraja Krishnamurthy
Langley Research Center, Hampton, Virginia

Miguel A. Aguilo
SANDIA National Laboratories, Albuquerque, New Mexico

April 2016

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

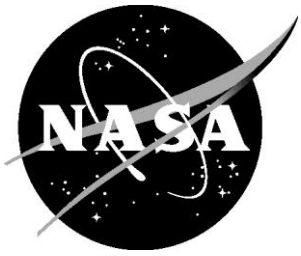
- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Phone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TP–2016-219186



Strain-Based Damage Determination Using Finite Element Analysis for Structural Health Management

Jacob D. Hochhalter and Thiagaraja Krishnamurthy
Langley Research Center, Hampton, Virginia

Miguel A. Aguilo
SANDIA National Laboratories, Albuquerque, New Mexico

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

April 2016

Acknowledgments

This work was supported by the NASA Aeronautics Research Institutes Seedling Program, under the project entitled "Coupling Damage-Sensing Particles and Computational Micromechanics to Enable the Digital Twin Concept." Sandia National Laboratories is a multiprogram laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract no. DE-AC04-94AL85000.

Available from:

NASA STI Program / Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199
Fax: 757-864-6500

Abstract

A damage determination method is presented that relies on in-service strain sensor measurements. The method employs a gradient-based optimization procedure combined with the finite element method for solution to the forward problem. It is demonstrated that strains, measured at a limited number of sensors, can be used to accurately determine the location, size, and orientation of damage. Numerical examples are presented to demonstrate the general procedure. This work is motivated by the need to provide structural health management systems with a real-time damage characterization. The damage cases investigated herein are characteristic of point-source damage, which can attain critical size during flight. The procedure described can be used to provide prognosis tools with the current damage configuration.

Nomenclature

a	=	damage size
ε_x	=	strain in the X-direction
ε_y	=	strain in the Y-direction
γ_{xy}	=	shear strain
h	=	height of the plate
i	=	iteration index
j	=	sensor index
f	=	objective function
$\bar{R}_j$	=	distance from the center of the damage to the j^{th} sensor
S	=	strain tensor
θ	=	orientation of the damage
w	=	width of the plate
X	=	location of the damage in the X-direction
Y	=	location of the damage in the Y-direction

Introduction

The development of validated multidisciplinary Integrated Vehicle Health Management (IVHM) technologies to prevent loss of aircraft control due to adverse conditions remains a safety-related challenge facing the next generation of aircraft. Such adverse conditions include environmental factors, actuator and sensor faults or failures, and point-source damage events. One major concern is the growth of undetected damage (i.e. cracks) due to such adverse effects, which can reach a critical size during flight, and ultimately result in loss of control of the aircraft. Hence, the development of efficient methodologies to determine the presence, location, and severity of damage in critical structural components is important in the progression of structural health management systems.

Approaches for the detection of damage size and location in structures can be based on changes in vibration or ultrasonic wave characteristics caused by the damage [1, 2, 3]. Approaches that use vibration characteristics are only effective for detecting relatively large damage, since small damage may have only negligible effects on vibration properties. Utilizing ultrasonic wave characteristics is effective in detecting smaller damage, but generally requires a dense network of sensors. Also, in addition to standard strain gauge technology, Fiber Optics Strain Sensing (FOSS) technology offers the ability to obtain strain measurements with minimal weight addition to the structures [4]. Such strain sensors enable real-time feedback of measured strain during usage. Even with the continuous advancement in these approaches, there still are large uncertainties associated with the determination of damage size, location, and orientation. A methodology is presented herein for employing strain sensing technologies,

coupled with the Finite Element Method (FEM) and an optimization procedure to determine existing damage.

In a precursor to the work presented herein, an attempt was made to predict damage size and location using a displacement field computed from the strain sensor measurements [5]. The displacement field based approach was used since it was found that more accurate damage determinations were made from known displacement fields, as compared to known strain fields. However, extraction of the displacement field from a measured strain field (which most sensor technologies provide) requires an additional non-trivial algorithm [6]. Also in the same previous study, the ABAQUS Extended Finite Element Method (XFEM) was used to represent the crack in the finite element mesh. Since then, it has been observed that using the ABAQUS XFEM package for the finite element simulations required significantly more computational resources than did using a standard ABAQUS FEM approach with a crack inserted explicitly into the model's topology. These findings are specific to the ABAQUS implementation of XFEM, and may not be a generally indicative comparison between XFEM and explicit representation of cracks. Therefore, this work used strain data directly from (virtual) sensors, and explicitly inserted cracks into the model's topology during the optimization process. Once determined, the damage configuration can be used by real-time prognosis tools to determine the criticality of the damage, and provide feedback to control systems. For example, Spear *et al.* [7] present a method whereby a surrogate model, based on an artificial neural network, is used to obtain in real-time updated residual-strength predictions after a damage event. In that work, design of experiments was used to define multiple damage states. The residual strength for each damage state was then computed using three-dimensional elastic-plastic simulations of crack growth in an integrally-stiffened panel. Those predicted residual strengths were then used to train the artificial neural network. The input to the trained neural network is then a characterization of damage that has occurred, which can be provided by the work presented here.

The remainder of the paper is organized into three sections. In Section 2, the development of the objective function and optimization procedure is described. In Section 3, two numerical examples are shown to illustrate the gradient-based optimization process, efficacy, and convergence behavior. Following those examples is a detailed discussion on possible problems that can be encountered using a gradient-based approach to the optimization problem investigated herein. Section 4 briefly illustrates how a nongradient-based optimization algorithm can also be used, and how it compares to the gradient-based approach.

Objective Function and Gradient-based Optimization

For the present study, damage in a two-dimensional plate is considered. The damage is assumed to be in the form of a crack in a plate with height, h, and width, w, as shown in Figure 1. The damage is characterized by four independent variables, which are also illustrated in Figure 1. The first two variables are the location of the center of the damage, X and Y, the third variable is the damage size, a, and the fourth variable is the damage orientation angle, θ, defined with respect to the X-axis.

Since no physical experiments were done in the present study, prospective damage states were simulated. Subsequently, the simulated strain data were extracted from the results at locations corresponding to virtual strain sensor locations, illustrated by the black dots in Figure 1. These reference strain data (identified by the superscript *'r'*) were then used during the optimization procedure to mimic the strain data that would be measured from sensors in service, after a damage event. The reference strain at a sensor location, j, is denoted as:

$$S_j^r = \begin{Bmatrix} \varepsilon_x^r \\ \varepsilon_y^r \\ \gamma_{xy}^r \end{Bmatrix}_j \tag{1.1}$$

The optimization procedure then evolves the four damage variables, *X*, *Y*, *a*, and *θ*, starting from some arbitrary initial guess, until convergence with the reference strain data is achieved. Each optimization iteration involves adding damage, defined by the above four variables, to a finite element model and computing the resulting strains. These variables, which are to be optimized, will be referred to as:

$$x_n^i = \begin{Bmatrix} X^i \\ Y^i \\ a^i \\ \theta^i \end{Bmatrix}_n \tag{1.2}$$

where *i* is the current iteration, and *n* (equal to four in this case) is the number of variables being optimized. Convergence is defined with respect to an objective function, which quantifies the difference between the reference strains and the strains computed at each of the optimization iteration. The computed strain at a sensor location, *j*, and optimization iteration, *i*, is denoted as:

$$S_j^i = \left\{ \varepsilon_y^i \right\} \tag{1.3}$$

It was found that improved convergence behavior was achieved when the strains were multiplied by a weight factor equal to the distance, $\overline{R}_j$, between the current iteration's estimated damage location and the sensor location, *j* (shown in Figure 1). The resulting objective function, which is to be minimized by the optimization procedure, is defined as:

$$f(x^i) = \left[\sum_{j=1}^{N} (S_j^r - S_j^i)(S_j^r - S_j^i)^T \bar{R}_j^2\right]^{1/2} \tag{1.4}$$

where *N* is the total number of sensors.

A quasi-Newton optimization algorithm was employed to minimize the objective function defined in Eq. 1.4. In quasi-Newton methods, the Hessian is updated by analyzing successive gradient vectors instead of computing the Hessian matrix. A limited memory Broyden, Fletcher, Goldfarb and Shanno (L-BFGS) method was used to approximate the inverse Hessian operator [8]. The search direction is determined by the product of the inverse Hessian operator and the objective function, Eq. 1.4, $M(x^i)\nabla f(x^i)$, and can be computed by performing a sequence of inner products and vector summations involving the gradient operator $\nabla f(x^i)$ and the pairs of changes in the trial values, Eq. 1.2, and objective function values, Eq. 1.4, $\{\Delta x_k, \Delta g_k\}$. Here $k = 1, \ldots, m$ and m denotes the current number of pairs $\{\Delta x, \Delta g\}$ stored. The maximum number of pairs $\{\Delta x, \Delta g\}$ stored, *i.e.* m_{max}, was set to 12 for this problem. At a given optimization iteration, if $m < m_{max}$, the pair $\{\Delta x^i, \Delta g^i\}$ is added to the limited memory working set. If $m = m_{max}$, the algorithm selects the latest half, *i.e.* $m_{max}/2$,$\{\Delta x, \Delta g\}$ pairs and adds the current pair, *i.e.* $\{\Delta x^i, \Delta g^i\}$, to the limited memory working set.

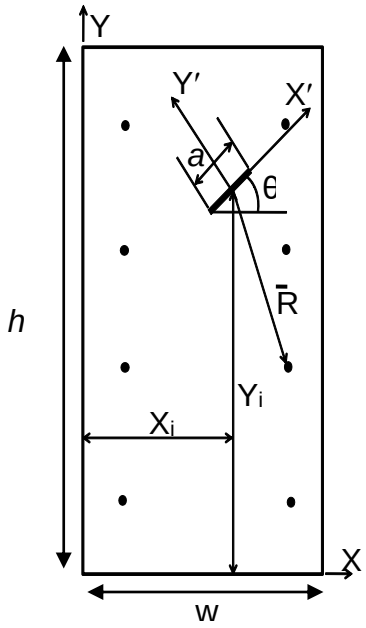


Figure 1: Schematic of the plate geometry and the four variables (X_r, Y_r, a_r, and θ_r) used to define the damage configuration. The filled black circles indicate virtual sensor locations.

The matrix-free quasi-Newton algorithm is described as follows:

While (tolerance>stopping tolerance or *i*<max iterations)
Compute search direction: $s^i = M(x^i)\nabla f(x^i)$, where $M(x^i)$ is the inverse Hessian operator
Update parameter vector: $x^{i+1} = x^i + \alpha^i s^i$
The step α^i is computed from a line search
$\Delta x^i = x^{i+1} - x^i$ and $\Delta g^i = \nabla f(x^{i+1}) - \nabla f(x^i)$
Update limited memory working set, i.e. store pairs $\{\Delta x^i, \Delta g^i\}$ in working set $\{\Delta x_k, \Delta g_k\}$, where $k = 1, \ldots, m$
i = i+1
end

To approximate the application of the gradient operator to the inverse Hessian, the L-BFGS two-loop recursion algorithm is used, which is described as follows:

$Q = \nabla f(x^i)$
For *(k = m – 1) to(k >= 0)*
$\rho_k = 1/(\Delta x_k^T \Delta g_k)$
$\beta_k = \rho_k \Delta x_k^T Q$
$Q = Q - \beta_k \Delta g_k$
k = k-1
end

$R = \nabla f(x^i)$
For *(k = 0) to (k<= m – 1)*
$\rho_k = 1/(\Delta x_k^T \Delta g_k)$
$\gamma = \rho_k \Delta g_k^T R$
$R = R + \Delta x_k(\beta_k - \gamma)$

k = k+1

end

Set $s^i = M(x^i)\nabla f(x^i) = R$

The gradient operator was approximated via a central difference approach. The central difference approximation is defined as

$$g(x_n^i) = [f(x_n^i + {}^1/_2\,\varepsilon_n) - f(x_n^i - {}^1/_2\,\varepsilon_n)]/\varepsilon_n \tag{1.5}$$

where ε_n denotes the perturbation on the $n\text{-}th$ parameter. Suitable values for ε_n were determined by quantifying the sensitivity of the objective function, Eq. 1.4 with respect to each parameter in x_n. The objective function is more sensitive to perturbations of the damage location parameters, *X* and *Y*, than to orientation and size, θ and *a*. Therefore, perturbations ε_1 and ε_2 are relatively small, defined here as 0.001, while perturbations ε_3 and ε_4 are 0.01. By considering the sensitivity of each component in x_n, the gradient is better approximated. The sensitivity of the damage parameters is determined from Eq. 1.5, with respect to each variable.

Since the full optimization problem being solved in this study has four variables, *i.e. n=4*, eight finite element simulations are required for the central difference gradient computation. However, the eight requisite simulations are independent of each other, and hence are run in parallel. Therefore, the gradient computation time is reduced to the expense of one finite element simulation. More precisely, in the case of the work presented in Section 3, this is the equivalent of about 15 seconds on a desktop with two quad-core AMD Opteron processors. The finite element models typically contained about 10^5 degrees of freedom, where characteristic quadratic triangular element edge lengths were about 1/20th of the crack size, *a*.

After computing the gradient, a backtracking line search routine was implemented to find a step α_k that gives a sufficient decrease in the objective function, $f(x^i)$, in the sense of the Armijo rule holding, *i.e.* $f(x^i + \alpha^i s^i) < f(x^i) + \sigma\alpha^i \nabla f(x^i)^T s^i$, where $\sigma = 0.0001$. In other words, a decrease in *f* of at least σ is sought along the search direction. The reader is referred to [9] for a detailed description of the backtracking line search algorithm utilized in this work.

Numerical Examples

Thus far, the objective function, $f(x^i)$, and the procedure by which it was minimized have been detailed. In this Section, the efficacy of the objective function and optimization procedure is tested on two numerical examples. In the first example, Case I, only the damage location is determined, while in the second example, Case II, all four damage parameters are determined. In both test cases, the reference damage configuration was simulated, and strains were queried at virtual sensor locations to mimic measured strain sensor data. Figure 2 shows the two reference damage configurations. For both, the reference configurations were generated on a rectangular Aluminum plate (Young's modulus =10^7 psi and Poisson's ratio=0.3) with height, *h*, and width, *w*. As is illustrated in Figure 3 (also in Figure 5), displacement boundary conditions were used exclusively to apply deformation in the test cases. Furthermore, the objective functional, Eq. 1.4, is dependent on the strain (not stress). Therefore, the mechanical properties of the plate, i.e. Young's modulus and Poisson's ration, do not influence the convergence of the inverse problem for the presented test cases.

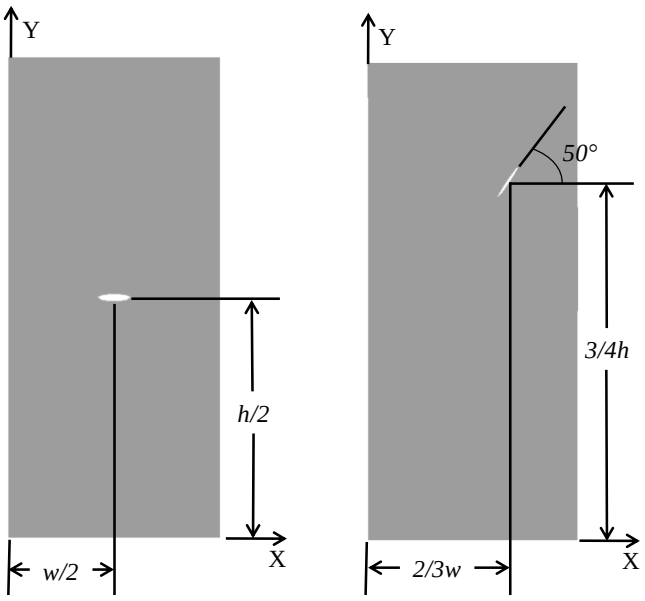


Figure 2: Reference damage configurations. Case I (left) and Case II (right).

Case I: Determining Damage Location (n=2)

The plate geometry, boundary conditions, reference damage and strain contours for Case I are shown in Figure 3. The sensor locations and spacing used during the optimization procedure are shown as white dots superimposed on the plate. In Case I, only the damage location is determined: *X* and *Y* are the only unknowns in the problem. The damage size was fixed at $a = 0.167w$, and the orientation was fixed at $\theta = 0$. This simplified problem illustrates how quickly the damage location can be determined for situations where crack size and orientation are not required.

The initial guess for the damage configuration was set to the lower-left corner, while keeping the crack fully contained within the plate: edge cracks were not considered. The optimization procedure was started, and a stopping condition of obtaining an absolute objective function, Eq. 1.4, value less than 0.005 was set. The results are summarized in Table 1. Both X_r and Y_r are determined to within 1 percent. Figure 4 illustrates the rate of convergence for this test problem. Convergence occurred in 10 iterations.

Table 1: Case I center damage configuration: Two-unknown variables

Initial Guess		Reference		Optimization Result		| % Difference |	
X_o	Y_o	X_r	Y_r	X_f	Y_f	X_{error}	Y_{error}
0.25w	0.125h	0.500w	0.500h	0.498w	0.496h	0.4	0.8

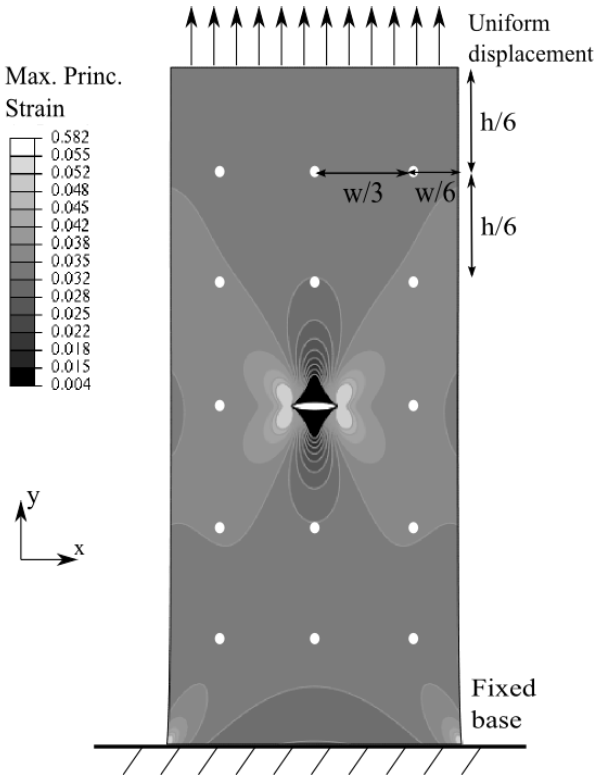


Figure 3: Reference strain field for Case I, a center crack with $a = 0.167w$, and $\theta = 0$. The virtual strain sensor locations are shown as white dots.

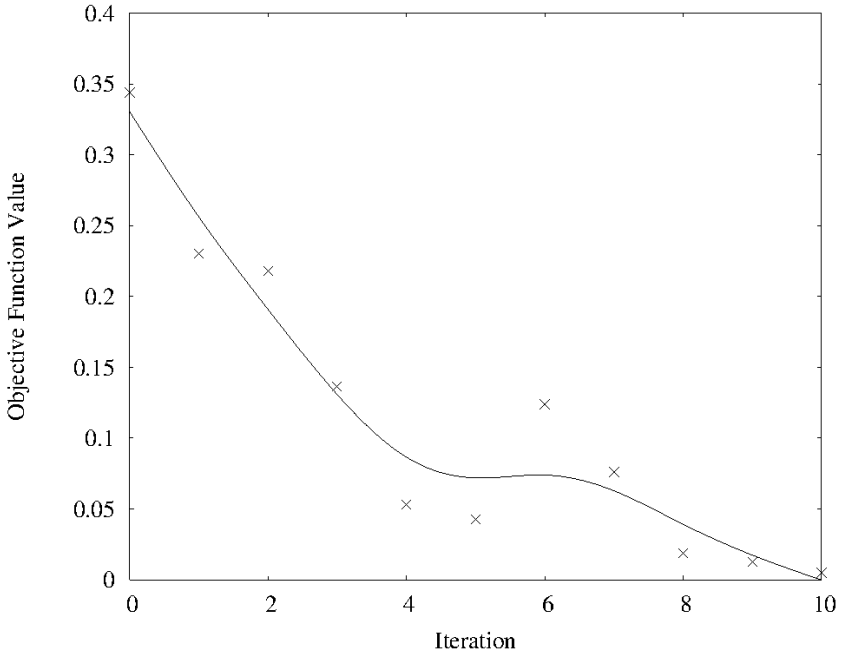


Figure 4: Convergence behavior for Case I. The line fit through the data illustrates the convergence trend.

Case II: Determining Damage Location, Size, and Orientation Angle (n=4)

The plate geometry, boundary conditions, reference damage, and strain contours for Case II are shown in Figure 5. The sensor locations used during the optimization procedure are shown as white dots superimposed on the plate. Case II investigates the full problem, where damage location, size, and orientation are all unknown. Case II contains damage located in the upper right corner, at $X = (2/3)w$ and $Y = (3/4)h$, and with $a = w/6$, and $\theta = 50^o$, with respect to the x-axis. As can be seen from Table 2a, both X_r and Y_r are detected to within 1 percent. It can be seen from Table 2b, that the damage orientation was detected to within 1 percent, while damage size was determined to within 5 percent. Figure 6 illustrates the rate of convergence of the problem. Comparing Figure 4 to Figure 6 illustrates the additional optimization iterations that are required to fully determine the damage characteristics, as opposed to determining only the damage location. In both Figures 4 and 6 it is seen that after approximately 5 iterations, the objective function value had been reduced to about 0.05. In both Cases, I and II, that initial enhanced rate of decrease in the objective function was due to the convergence of the crack location parameters. In Figure 6, after the fifth iteration, the convergence rate is significantly reduced, which occurs when location has already converged, and only orientation and size are still being optimized.

Table 2a: Case II damage location determination results

Initial Guess		Reference		Optimization Result		| % Difference |	
X_o	Y_o	X_r	Y_r	X_f	Y_f	X_{error}	Y_{error}
0.25w	0.125h	0.667w	0.750h	0.669w	0.746h	0.39	0.51

Table 2b: Case II damage size and orientation determination results

Initial Guess		Reference		Optimization Result		| % Difference |	
a_o	θ_o [rad]	a_r	θ_r [rad]	a_f	θ_f [rad]	a_e	θ_e
0.067w	0.1	0.167w	0.87	0.160w	0.86	4.5	0.96

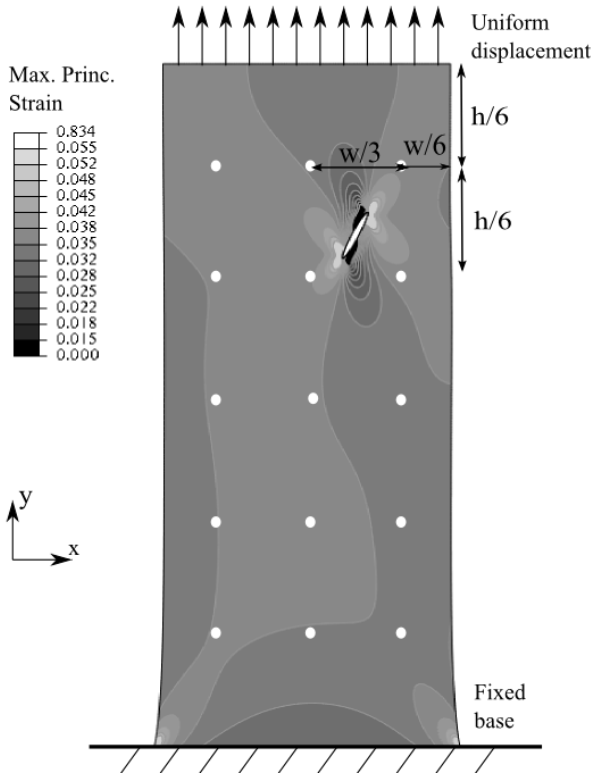


Figure 5: Reference strain field for Case II, a center crack with $a = 0.167w$, and $\theta = 0$. The virtual strain sensors used are shown as white dots.

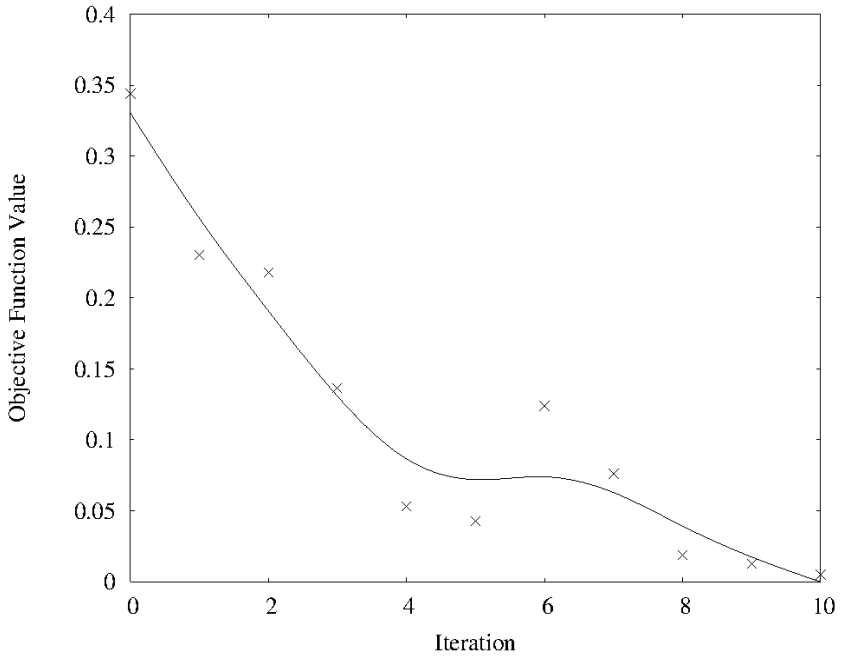


Figure 6: Convergence behavior for Case II. The line fit through the data illustrates the convergence trend.

Discussion

Investigating the use of a reduced sensor density

In the method presented in [5], it was assumed that the measured data were known for the entire plate geometry. This is not a practical assumption. From the numerical examples presented in this paper, it is concluded that five columns of three sensors for a total of fifteen sensors were sufficient to detect the particular damage configurations to within one percent error. However, fifteen sensors may not represent the minimum possible density of sensors to ensure accurate damage detection in the numerical examples presented. This is especially true for applications where error in the determined damage configuration that is greater than one percent can be tolerated. Furthermore, determination of the damage size is directly linked with the sensor density. In other words, equivalent convergence behavior results in cases with much smaller or much larger cracks when the sensor density is scaled accordingly.

Possible Gradient-Based Algorithm Shortcomings

In our initial studies, it was observed that error due to numerical approximations were inherent in Eq. 1.4, triggered by the finite element simulations and central difference approximation, and could prevent a and θ from converging to the reference values. To study this further, a simplified problem was explored, where a and θ were the only unknown variables. The reference location was set to $X = 0.5w$, $Y = 0.5h$. It was found that the damage orientation, θ_f, did not vary from its initial value, while the converged size, a_f followed Eq. 4.1 to within 4%. Eq. 4.1 was obtained by relating the direction cosines of the converged and reference values of the damage configuration.

$$a_f = a_r \left[\cos(\theta_f)/\cos(\theta_r)\right] \tag{4.1}$$

The reason for this is made obvious when the error surface is visualized, see Figure 7. The solid line represents Eq. 4.1. The filled circles are the converged values of a_f and θ_f. Figure 7 illustrates that the gradient of Eq. 1.4 is relatively small with respect to the line defined by Eq. 4.1. Hence, once the optimization stepped along that direction, the numerical approximation error began to overcome the true gradients.

To overcome issues associated with numerical approximations inherent in Eq. 1.4, the magnitude of those errors must remain small with respect to the change in values of Eq. 1.4 computed during the central difference computation. Because Eq. 1.3 is less sensitive to the damage size and orientation, size and location are particularly prone to this problem.

At least one contribution to the numerical error results from querying strains at virtual sensor locations, which do not necessarily correspond to locations where strain is computed in the finite element model, *i.e.* the Gauss points. The strains at the sensor locations, therefore, must be determined using the finite element shape functions to extrapolate the Gauss point strains to the nodes then interpolate that nodal data to the sensor location, which introduces approximation error into Eq. 1.4. To minimize the error introduced by the shape functions, the mesh can be refined near the sensor locations. Figure 8 shows a one-dimensional slice of Figure 7 for two mesh refinements. The objective function computed using the baseline mesh in this region is not smooth, and it is seen in this magnified sub-domain that the approximation error magnitude obscures the objective function gradient. However, upon mesh refinement, the gradient surpasses the approximation error amplitude. In the finite element simulations for the two Cases presented above, the baseline mesh contained approximately 10^4 degrees of freedom, while the fine mesh contained approximately 10^5.

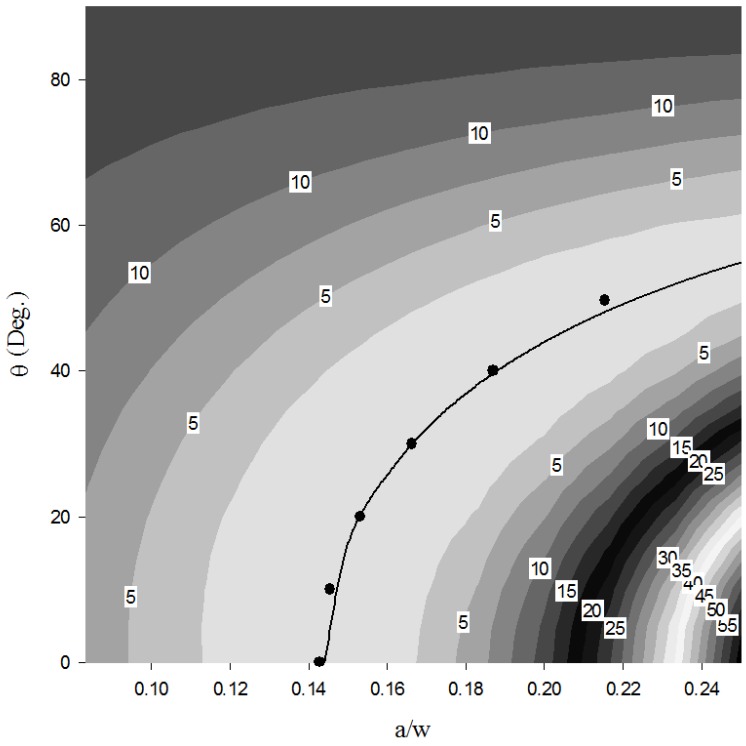


Figure 7: Error surface for the two-variable problem with the solution for Eq. 4.1 shown as a solid line and the results from the gradient-based method shown as filled circles.

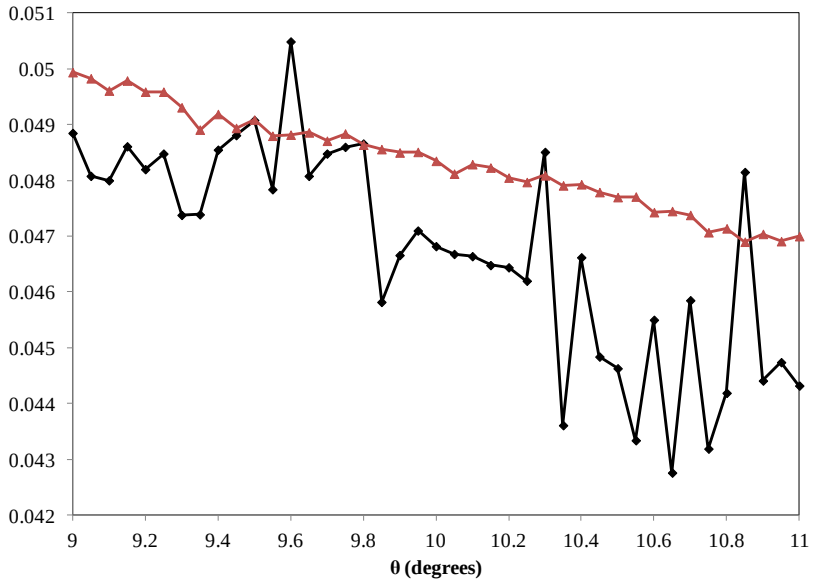


Figure 8: Objective function for a two-unknown variables problem with *a=0.147w* for two mesh size restraints. The red line illustrates the objective function values obtained from the refined mesh, while the black line illustrates those obtained from the coarse mesh.

An analytical solution would resolve much (or all) of the observed numerical approximation error. However, analytical solutions are only available for simplified geometries, and consequently, are not likely to be applicable in practical engineering applications. Instead, it is best to determine the optimal step sizes used in the finite difference computation that will allow for convergence, even with the objective function errors due to the numerical approximations. However, as seen in the next Section, non-gradient based algorithms can also be used for accurate damage determination. Such methods can be less sensitive to the numerical errors illustrated in Figure 8 since they do not rely on computing a gradient. To illustrate this point, the following Section presents results of using a genetic algorithm to solve Case II, while employing the coarse baseline mesh model.

Non-Gradient-Based Genetic Algorithm

One alternative approach to overcome the issues with the gradient-based approach is to employ a genetic algorithm (GA). To compare with the gradient-based method presented above, the GA implemented in Matlab [10] was employed. The GA is a non-gradient based approach to solving highly nonlinear objective functions and is more robust in converging to the global minimum. The GA operates by selecting the trial points from the population that are the most fit, *i.e.* resulted in the lowest objective function value, and generates a new population of points based on the fittest members. For a complete discussion of the Matlab GA implementation see [10]. However, a major disadvantage to using a GA for this problem is, unlike the gradient-based technique, which generates a single trial point upon each iteration, the GA generates a population of points at each iteration. This results in the need to perform many more finite element simulations during the optimization procedure. Furthermore, since the trial points in the first iteration are randomly generated, there is no guarantee that the optimization will produce equivalent converged solutions upon repeated trials. In addition, there are several empirical parameters that must be defined for the GA to operate: *population size, creation function, selection, mutation, and crossover*.

Population size is the number of trial points to solve upon each optimization iteration. It was found that a *population size* of 20 was large enough to provide sufficient variation diversity, while minimizing the number of function calls required. The *creation function* and *mutation* define how the trial points are generated. Here, by using a 'feasible population' *creation function* and 'adaptive feasible' m*utation*, the damage variables were guaranteed to stay contained within the upper and lower bounds. The *selection* approach defines how the fittest members from each population (iteration) are determined. The 'tournament' *selection*, used herein, chooses parents of the next generation by randomly sampling the current population and keeping only the fittest. Although, there were several combinations of possible inputs for these GA parameters, those discussed here provided the most consistent convergence behavior. In general, the most important consideration for the problem presented was found to be finding the empirical GA parameters that provide the ability to maintain diversity in the population after many generations.

The GA, with the specified parameters, was applied to solve the Case II test problem from Section 3. The GA algorithm results for Case II are shown in Tables 3a and 3b. From Tables 3a and 3b, it is seen that the GA resulted in damage detection accuracy similar to that of the gradient-based procedure. However, the cost of running $O(2000)$ finite element analyses for the GA, as opposed to $O(100)$ with the gradient-based technique exemplifies an advantage to using the gradient-based approach.

Table 3a: Genetic Algorithm Case II damage location determination results

Initial Guess		Reference		Optimization Result		| % Difference |	
X_o	Y_o	X_r	Y_r	X_f	Y_f	X_{error}	Y_{error}
0.25w	0.125h	0.667w	0.750h	0.668w	0.752h	0.12	0.22

Table 3b: Genetic Algorithm Case IIdamage size and orientation determination results

Initial Guess		Reference		Optimization Result		| % Difference |	
a_o	θ_o[rad]	a_r	θ_r[rad]	a_f	θ_f[rad]	a_{error}	θ_{error}[rad]
0.067w	0.1	0.167w	0.87	0.172w	0.904	3.4	3.7

Summary

In this paper, a strain-based damage optimization procedure combined with the finite element method was developed to determine existing damage size and location. It was demonstrated that the strains measured at a limited number of sensor locations can be effectively used to determine the location, size, and orientation of damage.

Numerical examples are presented to demonstrate a gradient-based optimization procedure in two cases. First, the damage location was estimated, while the damage size and orientation angle are held constant. It was shown that the damage location was estimated to within one percent for the damage configuration tested. Next, determination of the damage size, location, and orientation were tested. It was found that the location and orientation were determined to within one percent, while damage size was within 5 percent, for the cases tested. A discussion of possible issues involving the gradient step size and numerical approximation error was also provided.

Finally, it was demonstrated that a Genetic Algorithm (GA) estimates all the damage parameters to accuracy similar to that of the gradient-based approach. The GA is investigated since it is likely to be relatively insensitive to numerical errors inherent in using finite element models to compute the objective function value upon each iteration. However, the GA requires a large number of finite element analyses to estimate the damage parameters and hence is computationally more expensive.

References

[1] Kim, J.T. and Stubbs, N., “Improved Damage Identification Method Based on Modal Information,” Journal of Sound and Vibration, Vol. 252, pp. 223–238, 2002.

[2] Mal, A.K., Ricci, F., Banerjee, S., and Shih, F., “A Conceptual Structural Health Monitoring System Based on Vibration and Wave Propagation,” *Structural Health Monitoring: An International Journal*, Vol. 4, pp. 283 - 293, 2005.

[3] Wang, L., and Yuan, F.G., “Active Damage Localization Technique Based on Energy Propagation of Lamb Waves,” Smart Structures and Systems, Vol. 3, pp. 201-217, 2007.

[4] Quach, C., Vazquez, S., Tessler, A., Moore, J., Cooper, E., and Spangler, J., “Structural Anomaly Detection using Fiber Optic Sensors and Inverse Finite Element Method,” AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, CA. AIAA Paper 2005-6357, 2005.

[5] Krishnamurthy, T., and Gallegos, Adam M., “Damage Characterization Using the Extended Finite Element Method for Structural Health Management,”AIAA-2011-1701,Presented at the52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials

Conference 13th AIAA Non-Deterministic Approaches Conference, Denver, CO, April 4-7, 2011.
[6] Tessler, A., and Spangler, J., “A Least-Squares Variational Method for Full-Field Reconstruction of Elastic Deformations in Shear Deformable Plates and Shells,” Computer Methods in Applied Mechanics and Engineering, Vol. 194, pp. 327-335, 2005.
[7] Spear, A., Priest, A., Veilleux, M., Ingraffea, A., Hochhalter, J., “Surrogate Modeling of High-Fidelity Fracture Simulations for Real-Time Residual-Strength Predictions,” AIAA Journal, Vol. 49, No. 12, pp. 2770-2782, 2011.
[8] Nocedal, J., and Wright, J., “Numerical optimization.” Springer Science+ Business Media, 2006.
[9] Bertsekas, D., “Nonlinear Programming.” Athena Scientific, 2nd edition, September 1999.
[10] MathWorks, http://www.mathworks.com/help/toolbox/gads/.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-04-2016	Technical Publication	

4. TITLE AND SUBTITLE

Strain-based Damage Determination using Finite Element Analysis for Structural Health Management

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Hochhalter, Jacob D.; Krishnamurthy, Thiagarajan; Aguilo, Miquel A.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

533127.02.16.07.06

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

L-20689

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)

NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

NASA/TP-2016-219186

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified
Subject Category 26
Availability: STI Program (757) 864-9658

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A damage determination method is presented that relies on in-service strain sensor measurements. The method employs a gradient-based optimization procedure combined with the finite element method for solution to the forward problem. It is demonstrated that strains, measured at a limited number of sensors, can be used to accurately determine the location, size, and orientation of damage. Numerical examples are presented to demonstrate the generalprocedure. This work is motivated by the need to provide structural health management systems with a real-time damage characterization. The damage cases investigated herein are characteristic of point-source damage, which can attain critical size during flight. The procedure described can be used to provide prognosis tools with the current damage configuration.

15. SUBJECT TERMS

Crack detection; Finite element analysis; Optimization; Structural health monitoring

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	19	(757) 864-9658