10th World Congress on Structural and Multidisciplinary Optimization
May 19 -24, 2013, Orlando, Florida, USA

Development of an Optimization Software System for Nonlinear Dynamics using the Equivalent Static Loads Method

Hyun-Ah Lee[1], Gyung-Jin Park[2]

[1] Center of Novel Synthesis of Structural and Mechanical Systems, Hanyang University, Seoul, 133-791, Korea, fannail@hanyang.ac.kr
[2] Department of Mechanical Engineering, Hanyang University, Ansan-City, Gyeonggi-do, 426-791, Korea, gjpark@hanyang.ac.kr

1. Abstract

In the real world, structural systems may show nonlinear dynamic behavior. Linear static response structural optimization has been developed quite well by using linear static response analysis. Finite element analysis regarding the nonlinearities or dynamic effects has also been developed very well. However, conventional optimization methods considering the nonlinearities or dynamic effects are fairly difficult and expensive. The equivalent static loads method for non linear static response structural optimization (ESLSO) has been proposed for structural optimization of systems with various responses: linear dynamic response, nonlinear static response, and nonlinear dynamic response. Equivalent static loads (ESLs) are linear static loads which generate the same response field as that of a non linear static analysis. Non linear static analysis means an analysis which is non linear static analysis. Non linear static analysis is performed to evaluate the response (displacement, stress or strain, etc.) fields and ESLs are calculated from the responses. Linear static response structural optimization is carried out using the ESLs. The design procedure using ESLSO varies according to the characteristics of the analysis (linear dynamic, nonlinear static or nonlinear dynamic analysis) and the characteristics of the responses (displacement, stress or strain). ESLSO uses commercial structural optimization and finite element analysis software systems. Software systems should be selected based on the characteristics of the structures. Therefore, the interface between structural optimization and analysis software systems is required, and automation of the process for calculation of ESLs is also needed for the reduction of engineering cost. The software system for ESLSO is explained. A software system for the interface and calculation of ESLs is coded and it is made by C++ languages on the Windows operating system. Based on the characteristics of the structures, a design procedure is defined and a software system for ESLSO is developed for the automated design. First, a finite element model is defined and the type of analysis is selected. The user can define the information for ESLSO from menus in the software system for ESLSO. If the finite element model for structural optimization and finite element analysis exists and the information for ESLSO is provided in the software system, the software system for ESLSO runs automatically. Each module based on the characteristics of the response is coded as a class.

2. Keywords: Structural optimization, equivalent static loads (ESLs), equivalent static loads method for non linear static response structural optimization (ESLSO)

3. Introduction

Structural optimization has been developed using the finite element analysis with linear static response [1]. According to the characteristics of the design variables, structural optimization is classified into three areas. They are size, shape and topology optimization. Size optimization uses the fixed nodes data in the finite element model. In shape optimization, the node data of the finite element model are changed. Topology optimization determines the material distribution in the design domain [2, 3]. These optimization techniques are very excellent and are employed for wide applications. Some commercial software systems are available for structural optimization [4-6].

These days, computational analyses which are non linear static response analysis are frequently needed because engineering systems have very complex phenomena. However, the finite element analysis in structural optimization is linear static analysis. Therefore, the structural optimization techniques which can consider the various characteristic of analysis (linear dynamic, nonlinear static or nonlinear dynamic) are required. In this paper, the analysis which is non linear static response analysis is called non linear static response analysis. Structural optimization involving non linear static response analysis is named as non linear static response structural optimization. For examples, there are linear dynamic response structural optimization, nonlinear static response structural optimization and nonlinear dynamic response structural optimization.

For non linear static response structural optimization, the Equivalent Static Loads method for non linear static response Structural Optimization (ESLSO) was proposed by Choi and Park in 2002 [7]. The entire process of ESLSO is divided into two domains. One is an analysis domain and the other is a design domain. In the analysis domain, non linear static response analysis is performed and the load sets are calculated by multiplying the linear stiffness matrix and the response fields from non linear static response analysis. These load sets are named as Equivalent Static Loads (ESLs). In the design domain, ESLs are applied as the external loads in linear static response structural optimization. After the optimization is finished, a new design is updated and the updated design is again utilized in non linear static response analysis. This process is continued until the convergence criterion is satisfied. Algorithms and commercial software systems for non linear static response analysis and linear static response structural optimization have been very well developed. ESLSO exploits existing developed commercial software systems for the finite element analysis in the analysis domain and for linear static response structural optimization in the design domain.

ESLSO has been developed individually for linear dynamic response structural optimization [7-10], nonlinear static response structural optimization [11-13], nonlinear dynamic structural optimization [14-18] and non linear static response topology optimization [19-21]. The calculation process of ESLs is also different according to the responses (displacements or stresses) which are considered as the constraint functions in linear static response optimization. These techniques will be explained later. Interface of the commercial software systems is very important because many analysis and optimization systems are employed for ESLSO and the process of ESLSO are varied based on the characteristics of the analyses and the responses. In this research, the software system for ESLSO is explained. For the interface between the commercial software systems and the automatic process to calculate ESLs, the software system for ESLSO is coded by using C and C++ languages [22].

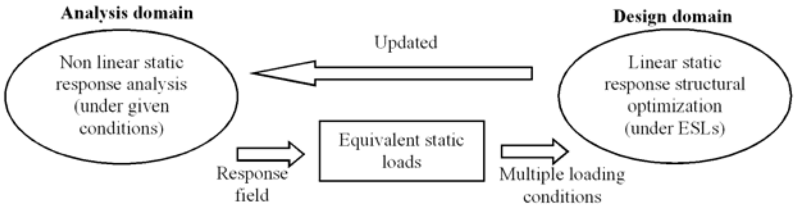
Analysis domain
Design domain
Updated
Non linear static response analysis (under given conditions)
Linear static response structural optimization (under ESLs)
Equivalent static loads
Response field
Multiple loading conditions

Figure 1: Schematic process between the analysis domain and the design domain

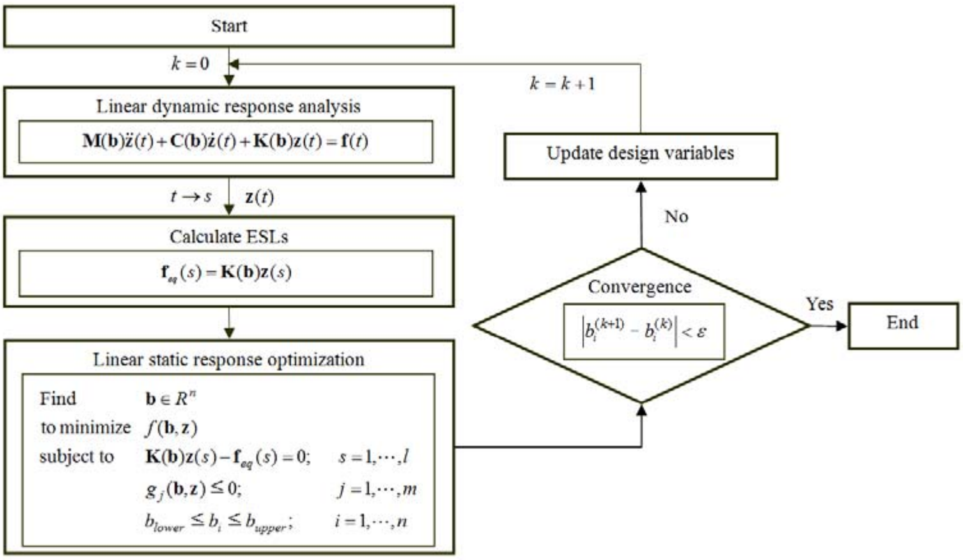
Start
$k=0$
$k=k+1$
Linear dynamic response analysis
$\mathbf{M}(\mathbf{b})\ddot{\mathbf{z}}(t)+\mathbf{C}(\mathbf{b})\dot{\mathbf{z}}(t)+\mathbf{K}(\mathbf{b})\mathbf{z}(t)=\mathbf{f}(t)$
Update design variables
$t \rightarrow s$
$\mathbf{z}(t)$
No
Calculate ESLs
$\mathbf{f}_{eq}(s)=\mathbf{K}(\mathbf{b})\mathbf{z}(s)$
Convergence
$\left|b_i^{(k+1)}-b_i^{(k)}\right|<\varepsilon$
Yes
End
Linear static response optimization
Find $\mathbf{b}\in R^n$
to minimize $f(\mathbf{b},\mathbf{z})$
subject to $\mathbf{K}(\mathbf{b})\mathbf{z}(s)-\mathbf{f}_{eq}(s)=0;$ $s=1,\cdots,l$
$g_j(\mathbf{b},\mathbf{z})\le 0;$ $j=1,\cdots,m$
$b_{lower}\le b_i\le b_{upper};$ $i=1,\cdots,n$

Figure 2: Linear dynamic response size/shape optimization with the ESLs

4. Software system for ESLSO

4.1. ESLSO

ESLSO has been developed individually for linear dynamic, nonlinear static and nonlinear dynamic response structural optimization. ESLSO consists of two domains as illustrated in Fig. 1. The optimization process using ESLSO is carried out in the analysis domain and the design domain. A non linear static response analysis is performed under the given forces in the analysis domain. The ESLs are generated with the results of non linear static response analysis in the analysis domain. The calculated ESLs are the static loads which generate the same response field in linear static analysis as those of non linear static response analysis. The ESLs are transmitted to the design domain, and linear static response structural optimization is carried out using the ESLs as the external loads in the design domain. When linear static response structural optimization is finished, the design variables are updated and a non linear static response analysis is performed again with the updated design variables in the analysis domain. The process is continued until the convergence criterion is satisfied. By using ESLSO, the results of non linear static response structural optimization can be obtained from a cyclic process between two independent domains. The cyclic process is called a design cycle. The linear dynamic, nonlinear static and nonlinear dynamic response structural optimization using ESLSO utilize the entire optimization process in Fig. 1.

4.2. Linear dynamic response structural optimization using ESLs

The procedure for linear dynamic response structural optimization is illustrated in Fig. 2. Linear dynamic analysis is performed and displacement responses at all times steps $\mathbf{z}_L(t)$ are generated from the analysis. ESLs $\mathbf{f}_{eq}(t)$ are calculated by multiplying the displacement vector $\mathbf{z}_L(t)$ and the linear stiffness matrix $\mathbf{K}_L$. ESLs are applied as the multiple loading conditions in linear static response structural optimization. After linear static response optimization, the convergence criterion is checked. If the norm of the design variable changes between the adjacent cycles is smaller than a small value ε, the process terminates. Otherwise, the design variables are updated and linear dynamic analysis is carried out again with the updated design variables.

The steps for ESLSO are as follows:

Step 1. Set initial design variables and parameters (cycle number: k = 0, design variables: $\mathbf{b}^{(k)} = \mathbf{b}^{(0)}$, convergence parameter: a small value ε).

Step 2. Perform linear dynamic analysis with $\mathbf{b}^{(k)}$.

Step 3. Calculate ESLs using $\mathbf{f}_{eq} = \mathbf{K}_L \mathbf{z}_L(t)$.

Step 4. Perform linear static response structural optimization with ESLs.

Step 5. When k = 0, go to Step 6. When k > 1, if the convergence criterion is satisfied, terminate the process. Otherwise, go to Step 6.

Step 6. Update the design variables, set k = k+1 and go to Step 2.

This process was proposed for size and shape optimization. In size optimization, the design variables are the properties of the finite element model such as the thickness and cross-sectional area. When the design variables are updated, the properties of the finite element model are changed. If shape optimization is performed, the location of nodes in the finite element model should be updated with the results. The ESLSO process for topology optimization is different from that for size or shape optimization. This will be explained in Section 4.5.

4.3. Nonlinear static response structural optimization

In linear static response optimization, the same response field between linear dynamic and static analysis can be obtained by using the ESLs based on the displacement responses (called displacement ESLs). That is, regardless of the characteristics of the constraint functions (responses) in the linear static optimization step, linear static response structural optimization can be performed with only the displacement ESLs. However, in nonlinear static response optimization, the ESLs should be individually calculated according to the characteristic of constraint functions. Until now, the ESLs for displacement constraints and stress constraints were proposed.

The process of the ESLs for displacement constraints in nonlinear static response optimization is similar to that of linear static response optimization in Fig. 2. Nonlinear static analysis is performed. The displacement ESLs $\mathbf{f}_{eq}^{z}$ are calculated with the nonlinear static displacement vector $\mathbf{z}_N$ and the linear stiffness matrix $\mathbf{K}_L$. The subscript ‘N’ means the values from nonlinear response and the subscript ‘L’ means the values from linear response. Although the displacement ESLs $\mathbf{f}_{eq}^{z}$ can generate the same displacements as the nonlinear static displacements, this does not generate the same stress responses.

The process of the ESLs for stress responses is illustrated in Fig. 3. The stress response $\boldsymbol{\sigma}_N$ is obtained from nonlinear static analysis. The stress responses $\boldsymbol{\sigma}_N$ are utilized as initial stresses in linear static analysis (initial stress analysis). Linear displacement vector $\mathbf{z}_L^{\sigma}$ is obtained from the initial stress analysis and the ESLs for the stresses $\mathbf{f}_{eq}^{\sigma}$ (called stress ESLs) are calculated by multiplying the linear displacement vector $\mathbf{z}_L^{\sigma}$ and linear stiffness matrix $\mathbf{K}_L$. The stress ESLs $\mathbf{f}_{eq}^{\sigma}$ is applied as the external loads in linear static response optimization. When linear static analysis is carried out with the stress ESLs $\mathbf{f}_{eq}^{\sigma}$, the linear stress response $\boldsymbol{\sigma}_L$ is obtained. The linear stress response $\boldsymbol{\sigma}_L$ should be the nonlinear stress response $\boldsymbol{\sigma}_N$. But, these stress responses are not exactly the same because the integral points for calculating the stresses are different between nonlinear static analysis and the initial stress analysis. Therefore, as shown in Fig. 3, the correction factors α_i are calculated using the stress responses $\boldsymbol{\sigma}_L$ and $\boldsymbol{\sigma}_N$. i is the element number of the FE model. By using Eq. (1), the corrected stresses $\hat{\sigma}_{L_i}^j$ are calculated and the corrected stresses $\hat{\sigma}_{L_i}^j$ from linear static analysis are the same as those of the nonlinear stress responses.

$$\hat{\sigma}_{L_i}^j = \sigma_{L_i}^j \times \alpha_i \tag{1}$$

If the displacement and stress responses should be both considered as the constraint function in linear static response optimization, the displacement ESLs $\mathbf{f}_{eq}^z$ and stress ESLs $\mathbf{f}_{eq}^\sigma$ are applied as the external loads.

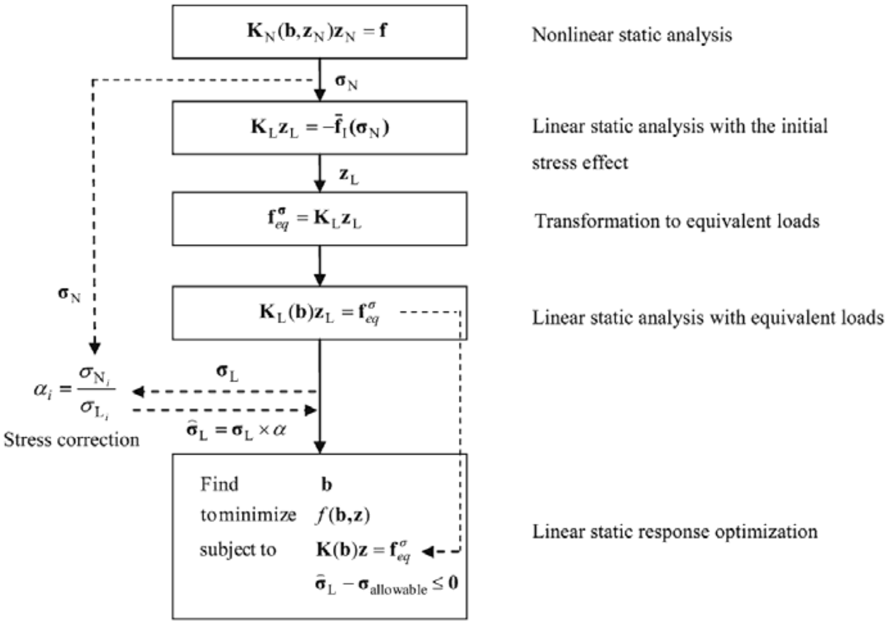
$\mathbf{K}_N(\mathbf{b},\mathbf{z}_N)\mathbf{z}_N = \mathbf{f}$
Nonlinear static analysis
$\boldsymbol{\sigma}_N$
$\mathbf{K}_L\mathbf{z}_L = -\bar{\mathbf{f}}_I(\boldsymbol{\sigma}_N)$
Linear static analysis with the initial stress effect
$\mathbf{z}_L$
$\mathbf{f}_{eq}^{\boldsymbol{\sigma}} = \mathbf{K}_L\mathbf{z}_L$
Transformation to equivalent loads
$\boldsymbol{\sigma}_N$
$\mathbf{K}_L(\mathbf{b})\mathbf{z}_L = \mathbf{f}_{eq}^{\sigma}$
Linear static analysis with equivalent loads
$\alpha_i = \frac{\sigma_{N_i}}{\sigma_{L_i}}$
Stress correction
$\boldsymbol{\sigma}_L$
$\hat{\boldsymbol{\sigma}}_L = \boldsymbol{\sigma}_L \times \alpha$
Find $\mathbf{b}$
tominimize $f(\mathbf{b},\mathbf{z})$
subject to $\mathbf{K}(\mathbf{b})\mathbf{z} = \mathbf{f}_{eq}^{\sigma}$
$\hat{\boldsymbol{\sigma}}_L - \boldsymbol{\sigma}_{\text{allowable}} \le \mathbf{0}$
Linear static response optimization

Figure 3: Generation of ESLs for stress responses in a nonlinear static system

4.4. Nonlinear dynamic response structural optimization

In nonlinear dynamic response structural optimization, the time variables are added to nonlinear static response structural optimization. Nonlinear dynamic response structural optimization is a combination of the methods in Sections 4.2 and 4.3. At each time step, nonlinear dynamic analysis is performed with Eq. (2).

$$\mathbf{M}(\mathbf{b})\ddot{\mathbf{z}}_N(t) + \mathbf{C}(\mathbf{b})\dot{\mathbf{z}}_N(t) + \mathbf{K}_N(\mathbf{b},\mathbf{z}_N(t))\mathbf{z}_N(t) = \mathbf{f}(t) \tag{2}$$

The displacement ESLs are generated by multiplying the nonlinear dynamic displacement vector $\mathbf{z}_N(t)$ from nonlinear dynamic analysis and linear stiffness matrix $\mathbf{K}_L$. The ESLs for stress response should be separately calculated from the same manner in Section 4.3. These ESLs are utilized as multiple loading conditions in linear static response structural optimization.

4.5. Non linear static response topology optimization

Topology optimization is different from size or shape optimization. In topology optimization, the compliance and mass fraction are only considered as the objective and constraint functions. When topology optimization is performed, only the displacement ESLs are utilized because the stress response is not considered in the linear topology optimization step. Even though the system is nonlinear static or a dynamic system, stress ESLs are not required. The process of ESLSO for topology optimization is illustrated in Fig. 4. The finite element analysis is performed and the displacement ESLs are made. Linear static topology optimization is carried out with the calculated ESLs.

The design variables for topology optimization are the artificial density function [23]. The number of design variables is the same as the number of the finite elements. In the ESLSO process, the process to update the design variables for topology optimization is different from that of size/shape optimization. Before updating the design variables, values of the design variables are transformed into values of the transformation variables using Eq. (3).

$$\beta_i = \begin{cases} 0 \text{ when } b_i \le \varepsilon_1 \\ 1 \text{ when } b_i > \varepsilon_1 \end{cases} : \qquad i = 1,\ldots,n \tag{3}$$

where b_i is the ith optimum design variable from linear static topology optimization. β_i is the transformation variable of b_i and the separation parameter ε_1 is a certain value between $b_{\min}$ and 1. If the value of a design variable is smaller than the separation parameter, the corresponding transformation variable has a value of 0. Otherwise, the corresponding transformation variable is regarded as 1. Therefore, all the transformation variables are replaced by 0 or 1. The finite element model is updated using the transformation variables. If β_i is 1, the corresponding element remains in nonlinear response analysis. Otherwise, the corresponding element is removed.

The convergence criterion for topology optimization is as follows:

$$\text{countif}\left(\left|\beta_i^{(k)} - \beta_i^{(k-1)}\right| \ge \varepsilon_2\right) \le n \times \varepsilon_3 : \qquad i = 1,\ldots,n \tag{4}$$

The convergence parameter ε_2 is an arbitrary value between 0 and 1, and ε_3 is a certain percentage. The number of transformation variables, which are changed more than ε_2 between adjacent cycles, is checked. If the checked number is less than $(n \times \varepsilon_3)$, the ESLSO process is finished.

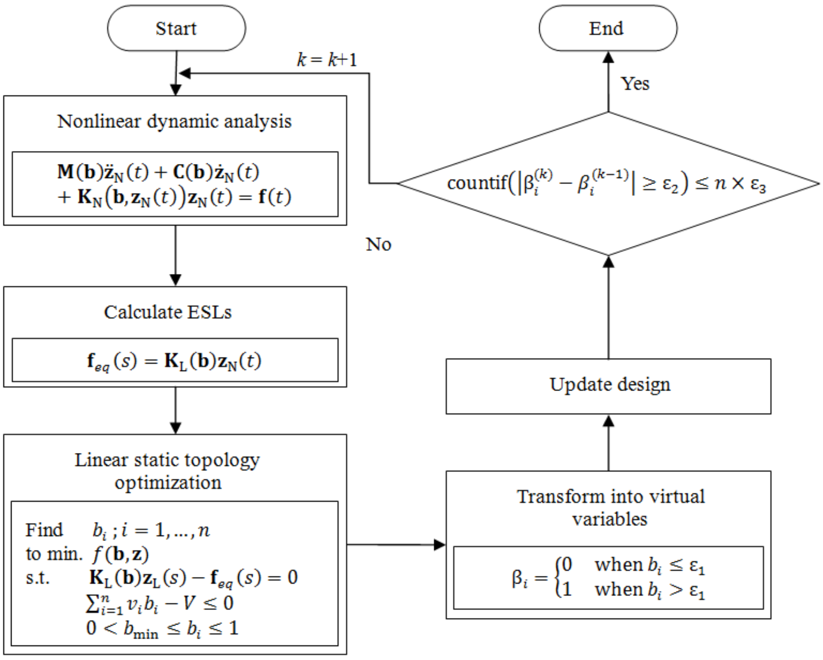
Start
End
$k = k+1$
Yes
Nonlinear dynamic analysis
$\mathbf{M}(\mathbf{b})\ddot{\mathbf{z}}_N(t) + \mathbf{C}(\mathbf{b})\dot{\mathbf{z}}_N(t) + \mathbf{K}_N(\mathbf{b},\mathbf{z}_N(t))\mathbf{z}_N(t) = \mathbf{f}(t)$
$\text{countif}(|\beta_i^{(k)} - \beta_i^{(k-1)}| \ge \varepsilon_2) \le n \times \varepsilon_3$
No
Calculate ESLs
$\mathbf{f}_{eq}(s) = \mathbf{K}_L(\mathbf{b})\mathbf{z}_N(t)$
Update design
Linear static topology optimization
Find $b_i ; i = 1,\ldots,n$
to min. $f(\mathbf{b},\mathbf{z})$
s.t. $\mathbf{K}_L(\mathbf{b})\mathbf{z}_L(s) - \mathbf{f}_{eq}(s) = 0$
$\sum_{i=1}^{n} v_i b_i - V \le 0$
$0 < b_{\min} \le b_i \le 1$
Transform into virtual variables
$\beta_i = \begin{cases} 0 & \text{when } b_i \le \varepsilon_1 \\ 1 & \text{when } b_i > \varepsilon_1 \end{cases}$

Figure 4: Nonlinear dynamic response topology optimization with the ESLs

5. Software system for ESLSO

Until now, the optimization procedures of ESLSO according to the characteristics of analysis (linear dynamic, nonlinear static and nonlinear dynamic analysis), the kinds of optimization (size/shape and topology optimization), and the type of response (displacement and stress response) were explained. If commercial software systems are selected for the finite element analysis and linear static optimization, interface between the two software systems should be made. The process for calculating ESLs should be performed automatically. Until the whole ESLSO process is terminated, all steps in the ESLSO process should be automatically carried out. The software system for ESLSO is developed for covering the entire optimization procedure of ESLSO. The software system is coded using the C and C++ languages on a Windows operating system. Each module based on the characteristics of the analysis, the selection of optimization, and the type of response is coded as a class.

5.1. Definition of input information for the software system

Figure 5 represents the design flow of the software system for ESLSO. The input data for ESLSO are obtained from the input screen. The developed input screen is user-friendly by using the graphics resource in Windows and the Microsoft Foundation Class library is employed for the purpose [24]. Figure 6 shows the main input screen. In the main input screen, the user can select the characteristic of finite element analysis using ESLSO. There are three types: linear dynamic, nonlinear static or nonlinear dynamic response optimization. The user should click the button on the main input screen to select the type of structural optimization: size, shape, size and shape, or topology optimization.

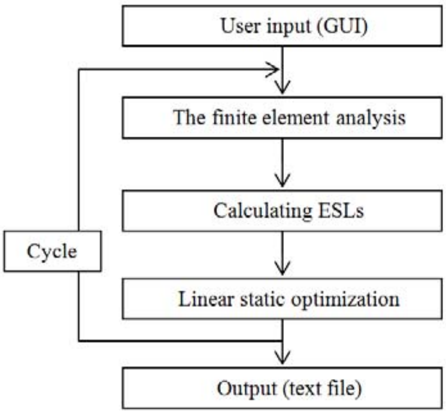

Figure 5: Design flow of the software system for ESLSO

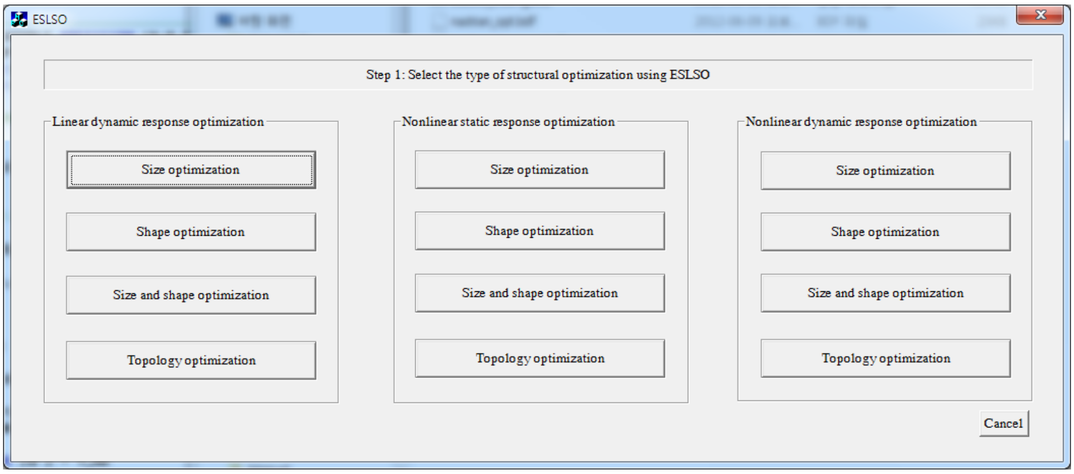
ESLSO
Step 1: Select the type of structural optimization using ESLSO
Linear dynamic response optimization
Size optimization
Shape optimization
Size and shape optimization
Topology optimization
Nonlinear static response optimization
Size optimization
Shape optimization
Size and shape optimization
Topology optimization
Nonlinear dynamic response optimization
Size optimization
Shape optimization
Size and shape optimization
Topology optimization
Cancel

Figure 6: The main input screen of the software system for ESLSO

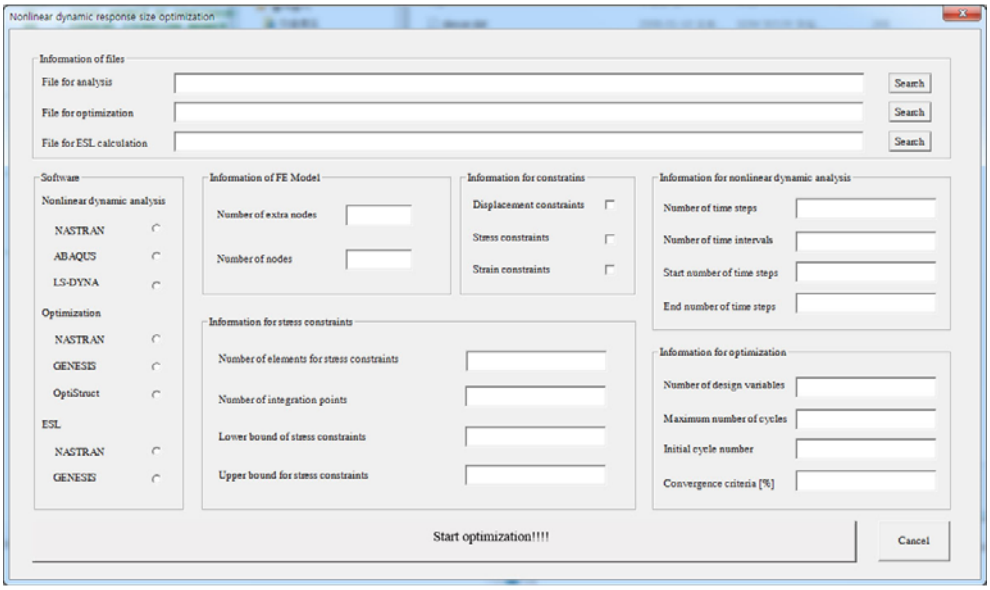

Figure 7: The second input screen of the software system for ESLSO (nonlinear dynamic size optimization)

When one button is selected on the main input screen, the second input screen pops up as shown in Figs. 7 and 8.

Figure 7 shows the second input screen for nonlinear dynamic response size optimization and Fig. 8 shows the second input screen for nonlinear dynamic response topology optimization. The 12 categories in Fig. 6 have an individual second input screen. As shown in Figs. 7 and 8, the individual second input screen requires different information for optimization using ESLs. In the second input screen, all the information for optimization using ESLs should be inserted.

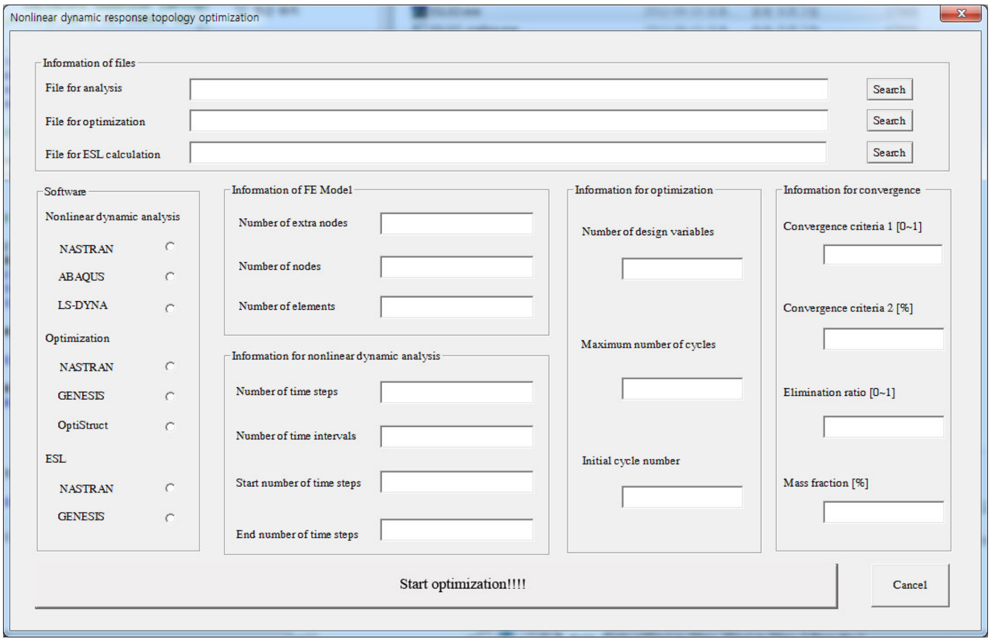

Figure 8: The second input screen of the software system for ESLSO (nonlinear dynamic topology optimization)

As mentioned in Section 4, there is a different process for ESLSO based on the characteristic of the analysis, the selection of optimization, and the characteristic of the response. When size optimization for nonlinear static or dynamic is selected in the main input screen, the characteristic of the constraint function (response) should be selected as shown in Fig. 8 because the displacement ESLs and the stress ESLs should be generated individually. If nonlinear dynamic response topology optimization is selected in the main input screen, the second input screen in Fig. 8 pops up. As shown in Fig. 8 and in Section 4.5, the user does not need to select the type of constraint function because the displacement ESLs are only calculated. The three parameters to update the design variables and to check the convergence in Eqs. (3) and (4) should be defined in the second input screen.

Table 1: Information of the software system

The main input screen		The second input screen																		
		Commercial software system								Information for finite element		Information for constraints		Time variables	Information for optimization					
		Finite element analysis			Generating the linear stiffness matrix		Linear static optimization			The number of nodes in FE model	The number of elements	Displacement ESLs	Stress ESLs	The total number of time steps	The number of design variables	The number of shape variables	Convergence parameters			
		Nastran	Abaqus	LS-dyna	Nastran	Genesis	Nastran	Genesis	OptiStruct				The number of integral points				ε	ε_1	ε_2	ε_3
Linear dynamic respoonse optimization	Size optimization	△			△	△	△	△	△	O		O			O		O			
	Shape optimization	△			△	△	△	△	△	O		O			O		O			
	Size and shape optimization	△			△	△	△	△	△	O		O			O	O	O			
	Topology opitmization	△			△	△	△	△	△	O	O	O			O			O	O	O
Nonlinear static response optimization	Size optimization	△	△		△	△	△	△	△	O		▲	▲	O	O		O			
	Shape optimization	△	△		△	△	△	△	△	O		▲	▲	O	O		O			
	Size and shape optimization	△	△		△	△	△	△	△	O		▲	▲	O	O	O	O			
	Topology opitmization	△	△		△	△	△	△	△	O	O	O		O	O			O	O	O
Nonlinear dynmaic response optimization	Size optimization	△	△	△	△	△	△	△	△	O		▲	▲	O	O		O			
	Shape optimization	△	△	△	△	△	△	△	△	O		▲	▲	O	O		O			
	Size and shape optimization	△	△	△	△	△	△	△	△	O		▲	▲	O	O	O	O			
	Topology opitmization	△	△	△	△	△	△	△	△	O	O	O		O	O			O	O	O

The required information for the second input screen of the software system is explained in Table 1. In Table 1, the triangle sign means that only one thing among the existing options must be definitely selected. The circle sign means that the information must be defined. The black colored triangle is optional information. Assume that nonlinear dynamic response size optimization is performed. Three commercial software systems (Nastran [5], Abaqus [25] and Ls-dyna [26]) are supported for finite element analysis. The user has to positively select only one commercial software system. In generating the linear stiffness matrix, Nastran and Genesis is supported and Nastran [5], Genesis [4] and OptiStruct [6] can be used for linear static optimization. The number of nodes in the finite element model must be inserted. For calculating ESLs, information of the constraint functions is selected. If displacement and stress responses are considered as the constraint functions, the user can select both. Then the displacement ESLs and stress ESLs will be individually calculated. If the stress ESLs should be generated, the commercial software system (Abaqus) is used for initial stress analysis. Because the problem has nonlinear dynamic response, information about the time variables is required. The number of design variables and convergence parameter must be defined. In this way, information about the problem must be correctly filled.

As shown in Table 1, Nastran [5] is supported for linear dynamic, nonlinear static and nonlinear dynamic response analysis. Abaqus [25] can be utilized for nonlinear static and nonlinear dynamic analysis, and Ls-dyna [26] can be used for only nonlinear dynamic analysis. Nastran [5] and Genesis [4] can be used for generating the linear stiffness matrix. Nastran [5], Genesis [4] and OptiStruct [6] are supported for linear static optimization. When all the information for the problem is inserted in the second input screen, the user can start the optimization using ESLs. The process is automatically continued until the optimization using ESLs is terminated.

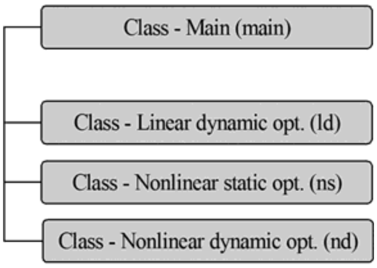
Class - Main (main)
Class - Linear dynamic opt. (ld)
Class - Nonlinear static opt. (ns)
Class - Nonlinear dynamic opt. (nd)

Figure 9: Main class diagram of software system for ESLSO

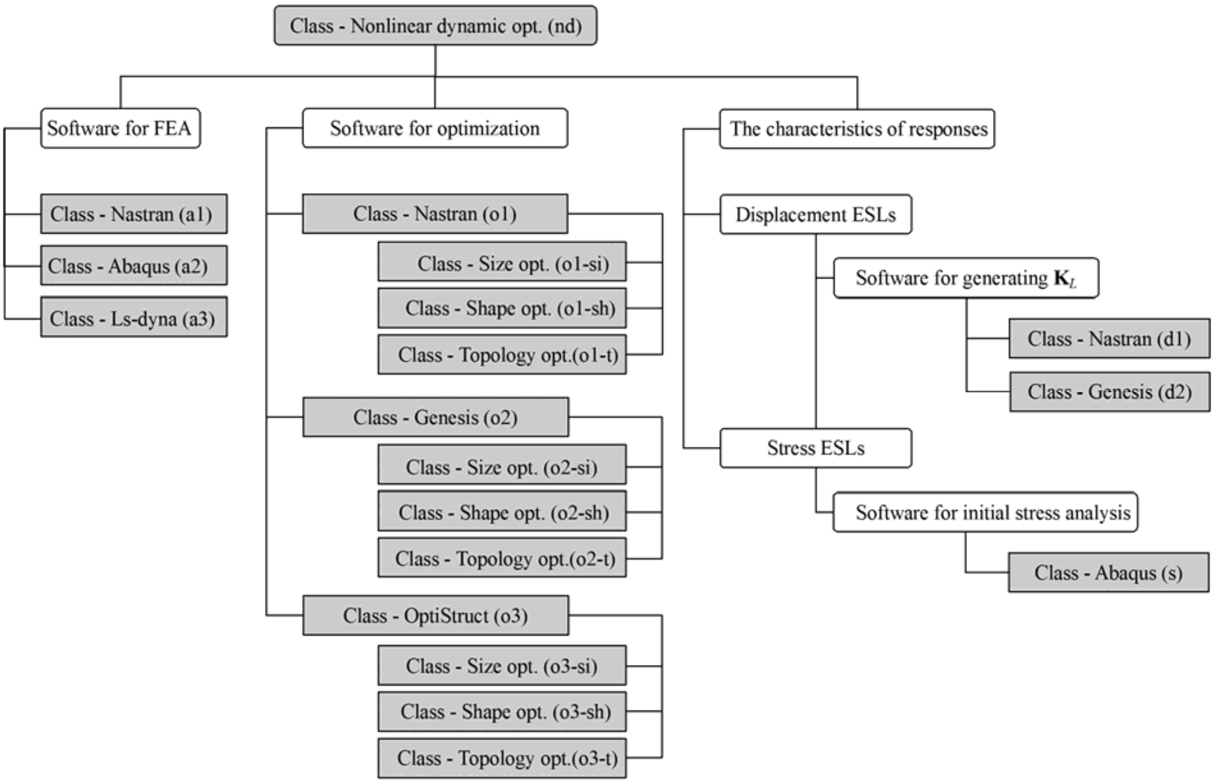
Class - Nonlinear dynamic opt. (nd)
Software for FEA
Class - Nastran (a1)
Class - Abaqus (a2)
Class - Ls-dyna (a3)
Software for optimization
Class - Nastran (o1)
Class - Size opt. (o1-si)
Class - Shape opt. (o1-sh)
Class - Topology opt.(o1-t)
Class - Genesis (o2)
Class - Size opt. (o2-si)
Class - Shape opt. (o2-sh)
Class - Topology opt.(o2-t)
Class - OptiStruct (o3)
Class - Size opt. (o3-si)
Class - Shape opt. (o3-sh)
Class - Topology opt.(o3-t)
The characteristics of responses
Displacement ESLs
Software for generating $\mathbf{K}_L$
Class - Nastran (d1)
Class - Genesis (d2)
Stress ESLs
Software for initial stress analysis
Class - Abaqus (s)

Figure 10: Class diagram for nonlinear dynamic response optimization using ESLs

5.2. Definition of classes

Each model according to the characteristics of the analysis (linear dynamic, nonlinear static and nonlinear dynamic analysis), the types of optimization (size, shape and topology optimization), the kinds of commercial software system, and the characteristics of the response (displacement and stress) is coded as a class. In Fig. 9, the main class diagram of the software system for ESLSO is illustrated. The main class "main" is defined by the set of three classes in Fig. 9. Three classes are defined as the characteristics of the analysis. The class is handled depending on the selection of the user in the main input screen. Three classes consist of several sub-classes.

In Fig. 10, the class diagram for nonlinear dynamic optimization using ESLs is illustrated. The relationships of the classes are identified from the class diagram in Fig. 10. Class "nd" for nonlinear dynamic response optimization consists of three parts: software for finite element analysis, software for optimization, and characteristics of the responses. Each part has several classes and these classes are classified and developed depending on the commercial software system because the software system for ESLSO is operated by using commercial software systems. As shown in Fig. 10, three classes construct the "software for optimization" part. Three classes are independent of each other and these classes consist of three sub-classes based on the types of optimization. In Fig. 11, the flow of the coding in the class Abaqus (s) in Fig. 10 is illustrated. As shown in Fig. 11, the class is coded using the theory of ESLSO explained in Section 4.3 and Fig. 3.

If linear dynamic response structural optimization is performed, class "ld" in Fig. 9 is performed. In this case, only Nastran is supported for linear dynamic analysis as shown in Table 1. Therefore, only one class such as class "a1" in Fig. 10 is developed in the "software for FEA" part. When nonlinear dynamic topology optimization is selected in the main input screen in Fig. 6, class "s" in Fig. 10 does not operate because the stress ESLs do not need to be calculated based on the theory of ESLSO in Section 4.5. Using the above process, the entire software system is coded. Coding is conducted based on the classes and their relationships.

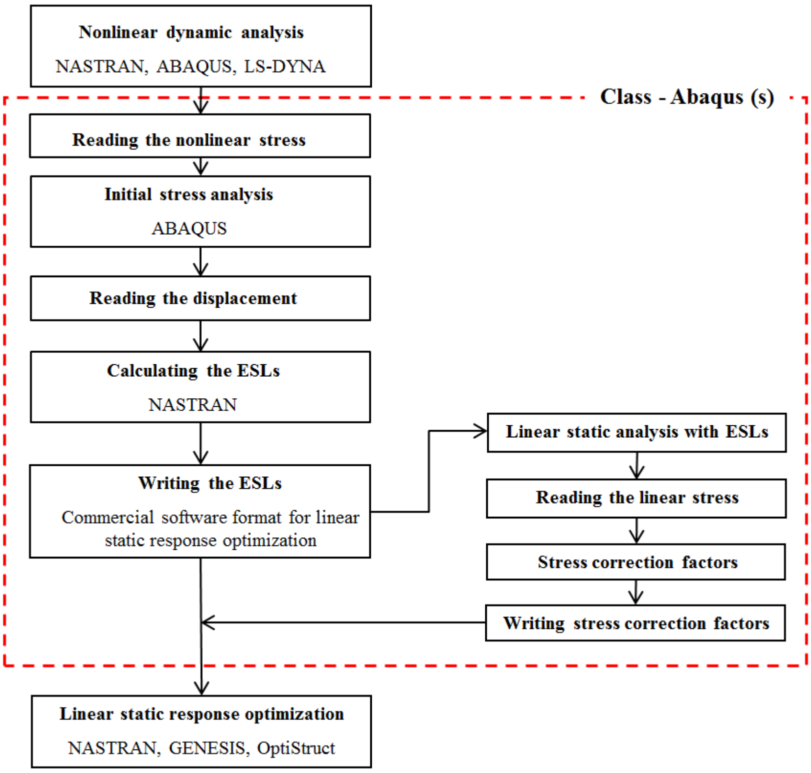
Nonlinear dynamic analysis
NASTRAN, ABAQUS, LS-DYNA
Class - Abaqus (s)
Reading the nonlinear stress
Initial stress analysis
ABAQUS
Reading the displacement
Calculating the ESLs
NASTRAN
Writing the ESLs
Commercial software format for linear static response optimization
Linear static analysis with ESLs
Reading the linear stress
Stress correction factors
Writing stress correction factors
Linear static response optimization
NASTRAN, GENESIS, OptiStruct

Figure 11: Flow of the coding in the class - Abaqus (s)

6. Conclusions

A software system is developed to perform the structural optimization using ESLs. The way to calculate ESLs is different depending on the characteristics of the response (displacement and stress). The process of ESLSO is different depending on the characteristics of analysis (linear dynamic, nonlinear static and nonlinear dynamic) and the types of optimization (size/shape and topology optimization). In addition, interface between the commercial software systems is required in ESLSO. However, coding is needed for the interface because the user can select a variety of commercial software systems. In ESLSO, when the user determines the characteristics of the response, the characteristics of analysis, the types of optimization and the commercial software systems, and the software system automatically performing ESLSO is developed. In the software system for ESLSO, the information for analysis and optimization is conveniently provided from a GUI type input screen. The software system for ESLSO is operated until the optimization process is terminated. The software system consists of many classes, and the classes are operated organically with other classes according to the input data of optimization. At a later date, if the new theory of ESLSO has been developed, this content can be installed in the software system as the class.

7. Acknowledgments

This work was supported by Korean Advanced Institute of Women in Science, Engineering and Technology Support Programs for Conferences, 2013.

8. References

[1] G.N. Vanderplaats, Thirty years of modern structural optimization, *Advances in Engineering Software*, 16 (2), 81-88, 1993.

[2] G.J. Park, *Analytical Methods in Design Practice*, Springer, Germany, 2007.

[3] R.T. Haftka and Z. Gürdal, *Elements of Structural Optimization*, Kluwer, Dordrecht, Netherland, 1992.

[4] GENESIS 10.0 User's Manual, Vanderplaats Research and Development, Inc., 2009.

[5] MD NASTRAN R3 User's Manual, MSC Software Co., Santa Ana, CA, U.S.A., 2008.

[6] Altair OptiStruct, Introduction to OptiStruct FEA Version 10.0, Altair Engineering, Inc., MI, U.S.A., 2009.

[7] W.S. Choi and G.J. Park, Structural optimization using equivalent static loads at all the time intervals, *ComputerMethods in Applied Mechanics and Engineering*, 191 (19-20), 2105-2122, 2002.

[8] G.J. Park and B.S. Kang, Validation of a structural optimization algorithm transforming dynamic loads into equivalent static loads, *Journal of Optimization Theory and Applications*, 118 (1), 191-200, 2003.

[9] K.J. Park, J.N. Lee and G.J. Park, Structural shape optimization using equivalent static loads transformed from dynamic loads, *International Journal for Numerical Methods in Engineering*, 63, 589-602, 2005.

[10] H.A. Lee, Y.I. Kim, G.J. Park, R.M. Kolonay, M. Blair and R.A. Canfield, Structural optimization of a joined-wing using equivalent static loads, *Journal of Aircraft*, 44 (4),1302-1308, 2007.

[11] M.K. Shin, K.J. Park and G.J. Park, Optimization of structures with nonlinear behavior using equivalent loads, *Computer Methods in Applied Mechanics and Engineering*, 196 (4-6), 1154-1167, 2007.

[12] Y.I. Kim, G.J. Park, R.M. Kolonay, M. Blair and R.A. Canfield, Nonlinear response structural optimization of a joined-wing using equivalent loads, *AIAA Journal*, 46 (11), 2703-2713, 2008.

[13] S.I. Yi, H.A. Lee and G.J. Park, Optimization of a structure with contact conditions using equivalent loads, *Journal of Mechnical Science and Technology*, 25 (3) 773-782, 2011.

[14] S.B. Jeong, S.I. Yi, C.D. Kan, V. Nagabhushana and G.J. Park, Structural optimization of an automobile roof structure using equivalent static loads, *Proceedings of the Institution of Mechanical Engineers, Part D, Journal of Automobile Engineering*, 222 (11), 1985-1995, 2008.

[15] Y.I. Kim, G.J. Park, R.M. Kolonay, M. Blair and R.A. Canfield, Nonlinear dynamic response structural optimization of a joined-wing using equivalent static loads, *Journal of Aircraft*, 46 (3), 821-831, 2009.

[16] Y.I. Kim and G.J. Park, Nonlinear dynamic response structural optimization using equivalent static loads, *Computer Methods in Applied Mechanics and Engineering*, 199 (9-12), 660-676, 2009.

[17] S.B. Jeong, S. Yoon, S. Xu and G.J. Park, Non-linear dynamic response structural optimization of an automobile frontal structure using equivalent static loads, *Proceedings of the Institution of Mechanical Engineers, Part D, Journal of Automobile Engineering*, 224 (4), 489-501, 2010.

[18] S.I. Yi, J.Y. Lee and G.J. Park, Crashworthiness design optimization using equivalent static loads, *Proceedings of the Institution of Mechanical Engineers, Part D, Journal of Automobile Engineering*, 226 (1), 23-38, 2012.

[19] S.J. Lee, H.A. Lee, S.I. Yi, D.S. Kim, H.W. Yang and G.J. Park, Design flow for the crash box in a vehicle to maximize energy absorption, *Proceedings of the Institution of Mechanical Engineers, Part D, Journal of Automobile Engineering*, 227 (2), 179-200, 2013.

[20] H.H. Jang, H.A. Lee, J.Y. Lee and G.J. Park, Dynamic response topology optimization in the time domain using equivalent static loads, *AIAA Journal*, 50 (1), 226-234, 2012.

[21] H.A. Lee and G.J. Park, Topology optimization for structures with nonlinear behavior using the equivalent static loads, *ASME-Journal of Mechanical Design*, 134 (3), 031004-14, 2012.

[22] Stroustrup B (2000) The C++ programming language. Addison Wesley, USA

[23] M.P. Bendsøe, and O. Sigmund, Topology Optimization by Distribution of Isotropic Material, *Topology Optimization: Theory, Methods and Applications*, Springer, Germany, 2003.

[24] Microsoft Press, MFC Library Reference, Redmond, WA, 1998.

[25] Abaqus Analysis User's Manual Version 6.8, SIMULIA, Providence, RI, 2008.

[26] Ls-dyna User's Manual, Livermore Software Technology Co., Livermore, CA, 2006.