

## 최적설계 관점에서의 설계 소프트웨어 성능 비교에 관한 연구

홍을표\*, 박철민\*\*, 박경진†

(2001년 3월 29일 접수, 2001년 10월 19일 심사완료)

### A Comparative Study of the Design Software Systems from the View Point of Optimization

Eul-Pyo Hong, Chul-Min Park, Gyung-Jin Park

**Key Words :** Design Software(설계 소프트웨어), Design Environment(설계 환경), Optimization(최적설계)

#### Abstract

Analysis technology is widely accepted and quite popular these days. Incorporation of the analysis result into design process is a key factor for the success of the analysis area. A few design software products have been commercialized. Generally, they are trying to make an interface between various design methods and analysis software. Optimization is a representative design method. The products are investigated and compared for the aspects of user convenience and algorithm performance. A few popular products are selected. Graphic user interface (GUI) is compared for the function and efficiency. The performances of the optimization algorithms are tested by mathematical and engineering examples. The results are discussed.

#### 1. 서론

해석 소프트웨어의 성능이 향상되고, 안정됨에 따라, 제품 설계에서 해석 소프트웨어들의 사용 범위와 역할이 점점 확대되고 있다. 기존의 설계는 해석 소프트웨어에서 얻어지는 해석 결과만을 이용하여 이루지는 경우가 많았다. 이런 설계과정은 설계자의 직관, 경험, 기술 등에 의해 좌우된다. 설계 전반에 걸친 설계자의 개입은 복잡하게 구성되어 있는 시스템을 설계하는 경우 부정확하고, 비효율적인 설계 결과를 가져 올 수 있다.<sup>(1)</sup> 이런 단점들은 최적 설계와 같은 이미 잘 정립되어 있는 설계 방식을 사용함으로써 극복될 수 있다.

많은 경우, 해석 소프트웨어에는 최적 설계와 같은 설계 기능이 포함되어 있지 않다. 이 경우, 설계자는 해석 소프트웨어에 설계기능을 추가하여 설계를 수행해야 한다. 경우에 따라, 설계 기능을 가지고 있는 상용 해석 소프트웨어들도 존재한다.

그러나 여러 분야를 동시에 고려하여 서로 연관된 요소들을 설계하는 다분야 통합 최적설계(MDO, Multidisciplinary Design Optimization)<sup>(2)</sup>와 같이 경우, 각 소프트웨어의 설계 기능만으로는 설계를 수행할 수 없다. 이 같은 경우에도 설계자는 설계 기능을 직접 추가해 주어야 한다. 그러나, 상용 소프트웨어와 같이 소프트웨어의 코드가 없거나, 코드가 존재한다고 하더라도 매우 복잡하거나, 설계자에게 익숙하지 않은 언어로 개발된 코드로 구성되어 있는 경우 설계 기능을 추가하는 것은 매우 복잡하고 어려운 작업이다.

해석 소프트웨어를 제어하여 설계 기능을 제공하는, 이하 설계 소프트웨어라 명명할 것들은 이런 문제점들을 극복하고, 최적 설계와 같은 이미 잘 정립된 설계 기법들을 사용하여 설계를 수행할 수 있도록 개발된 소프트웨어이다. 설계 소프트웨어는 설계자가 해석 소프트웨어에 설계 기능을 추가하기 위하여 직접 프로그램을 개발해야 하는 어려움 없이, 설계 소프트웨어가 제공하는 사용자 환경에서 쉽게 해석 소프트웨어에 설계 기능을 추가하여 설계를 수행할 수 있도록 도와준다. 설계 소프트웨어는 해석 소프트웨어의 입/출력 파일을 통하여 해석 소프트웨어와 데이터를 정의함으로써 설계 환경을 정의하여 해석 소프트웨어를 이용한

\* 회원, 한양대학교 대학원 기계설계학과

\*\* 포항산업과학연구원

† 책임저자, 회원, 한양대학교 기계정보산업공학부

E-mail : gipark@hanyang.ac.kr

TEL : (031)400-5246 FAX : (031)407-0755

설계가 가능하도록 하는 방식을 사용하고 있다. 그리고, 정의된 설계 환경과 설계 소프트웨어가 가지고 있는 최적 설계나 실험계획법과 같은 설계 모듈과 연결하여 체계적인 설계를 가능하도록 하고, 이를 통하여 얻어진 결과를 설계자에게 표시해 주는 역할을 한다. 이러한 기능을 가진 소프트웨어로는 VisualDOC,<sup>(3)</sup> iSIGHT,<sup>(4)</sup> OPTIMUS,<sup>(5)</sup> ModelCenter<sup>(6)</sup> 등이 있다.

이번 연구에서는 상용 설계 소프트웨어들의 설계 환경 정의를 위한 기능들의 특징과 설계 소프트웨어들이 공통적으로 가지고 있는 최적설계 모듈과 실험계획법 모듈의 성능을 비교하였다. 설계 환경을 정의하는 부분은 설계에서 사용될 해석 소프트웨어와 설계 변수들을 정의하기 위한 기능들을 가지고 있는 부분이다. 이 부분은 설계과정과 직접적으로 관계된 부분이기 때문에 설계 소프트웨어의 성능에서 빼 놓을 수 없는 부분이다. 따라서 이 부분의 성능을 먼저 비교하였다. 최적 설계와 같은 설계 모듈의 성능은 해석의 횟수와 관계된 부분이다. 해석 소프트웨어를 이용하는 경우, 해석의 횟수는 설계 시간과 비례하기 때문에 설계 소프트웨어의 성능과 직접적인 관계가 있다. 이런 관점에서 설계 모듈들의 성능을 비교하였다. 각 설계 소프트웨어들은 서로 다른 여러 설계 모듈을 제공하고 있다. 이번 연구에서는 여러 분야에서 많이 사용되고 있고, 설계 소프트웨어들이 공통적으로 제공하는 최적 설계 모듈과 실험계획법 모듈의 성능을 비교하였다. 특히 최적 설계 모듈에 비중을 두고 설계 소프트웨어의 성능을 비교하였다. 최적 설계 모듈의 성능을 비교하기 위하여 몇 가

지 수학적 예제와 공학적인 예제들을 사용하였다.

## 2. 설계 소프트웨어 성능 비교 항목

설계 소프트웨어는 Fig. 1 과 같이 크게 설계 환경을 정의하고 제어하는 부분과 설계를 수행하는 부분, 설계 결과를 사용자에게 보여주는 부분의 세 부분으로 구성되어 있다. 특히 설계 환경을 제어하는 부분과 설계를 수행하는 부분은 설계 소프트웨어의 성능을 좌우하는 매우 중요한 부분이다.

설계 환경을 제어하는 부분은 해석 소프트웨어를 직접 실행하고, 결과를 얻는 부분이다. 그리고, 설계 소프트웨어와 해석 소프트웨어가 설계 변수나 목적함수와 같은 데이터들을 교환하는 부분이기 때문에 설계 소프트웨어에서는 매우 중요한 부분이다. 설계를 수행하기 위해서는 설계에서 사용될 해석 소프트웨어와 설계에서 사용될 데이터를 정의하여 설계가 수행되는 설계환경을 정의하고, 이를 제어할 수 있어야 한다. 그리고, 설계에서 사용될 데이터들을 정의하기 위해서는 해석 소프트웨어들의 입/출력 파일을 이용한다. 입/출력 파일을 이용하여 데이터에 접근하는 방식은 비효율적인 방식이지만, 소프트웨어의 소스 코드가 있는 경우와 없는 경우 모두에서 사용될 수 있는 유연한 방법이다.<sup>(7,8)</sup> 설계 소프트웨어는 해석 소프트웨어의 입/출력 파일에서의 데이터들의 위치를 정의하고, 이 위치를 이용하여 설계에 사용되는 데이터에 접근하고, 제어한다. 이 과정에서 설계 변수나 목적함수의 위치가 잘못 정의되는 경우, 설계자는 원하는지 않는 설계 변수나 목적함수를 이용하여 설계를 수행하게 되거나, 경우에 따라 해석 소프트웨어의 입력 파일에서 문제가 발생하여 설계를 수행할 수 없는 경우도 발생한다. 이런 점들 때문에 해석 소프트웨어를 제어하는 부분은 설계 소프트웨어에 있어서 매우 중요한 부분이다. 이렇게 해석 소프트웨어를 제어하고, 입/출력 파일을 통하여 데이터를 제어하는 방식은 소프트웨어에 따라 조금씩 차이가 있다. 따라서 설계 소프트웨어들의 성능을 비교하기 위하여 해석 소프트웨어 제어를 위한 기능을 우선 먼저 비교해 보았다.

설계 환경이 정의되면, 정의된 해석 소프트웨어와 데이터들은 최적 설계와 같이 설계 소프트웨어에 내장되어 있는 설계 모듈과 연결되어 설계를 수행하게 된다. 설계 소프트웨어는 최적설계와 같이 이미 체계적으로 정립된 설계 방식에 대한 설계 모듈을 가지고 있다. 소프트웨어에 따라 제공하는 설계 방식은 조금씩 차이가 있지만, 최적 설계와 실험계획법을 이용한 설계 방식을 공통적으

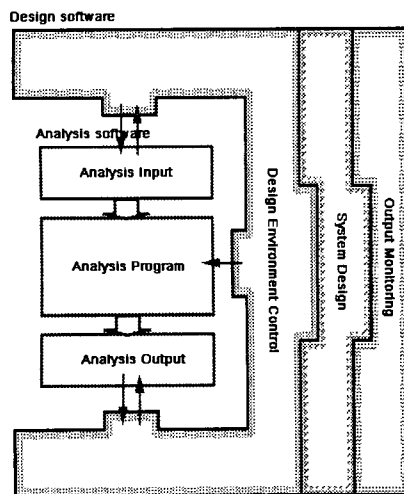


Fig. 1 Functions of the design software packages

**Table 1** Software release

Software	Release	Developer
VisualDOC <sup>(3)</sup>	1.2	VR&D
iSIGHT <sup>(4)</sup>	5.0	Engineous
OPTIMUS <sup>(5)</sup>	2.2	LMS International
ModelCenter <sup>(6)</sup>	2.0	Phoenix Integration

로 제공하고 있다. 최적 설계는 최적 설계 알고리즘을 이용하여 설계 변수의 값을 변경하고, 변경된 설계 값에서의 결과를 관찰하는 과정을 반복함으로써 더 나은 설계 값을 찾아가는 설계 방식이다.<sup>(9-11)</sup> 이 과정에서 최적 설계 알고리즘의 성능은 설계의 효율성, 정확성과 직결되어 있다. 즉, 최적 설계 알고리즘의 성능에 따라 설계 값의 정확성이나, 설계 과정에서의 효율성의 차이가 발생할 수 있다. 이런 관점에서 이번 연구에서는 각 설계 소프트웨어들의 최적 설계 모듈들의 성능을 몇 가지 예제를 통하여 비교해 보았다. 그리고, 실험 계획법을 이용한 설계에서의 특징을 살펴보고, 장단점을 관찰하였다.

위의 두 과정을 거쳐 얻어진 설계 결과들은 설계자에게 표시된다. 이렇게 결과를 표시하는 부분은 설계 성능이나 결과의 정확성과는 거리가 있는 부분이기 때문에 설계 성능의 관점에서는 위에서 설명한 두 부분보다 상대적으로 비중이 작은 부분이라고 할 수 있다. 여기에서는 설계 소프트웨어의 성능을 비교하는데 목적을 두고 있기 때문에 이 부분에 대한 비교는 포함시키지 않았다.

본 연구에서 비교된 설계 소프트웨어들의 버전 등의 정보를 Table 1 에 표시하였다.

### 3. 설계 환경 정의 기능 비교

해석 소프트웨어를 이용하여 설계를 수행하기 위해서는 설계에 사용될 해석 소프트웨어와 설계 모듈을 정의하고 입/출력 파일을 통하여 설계 데이터들을 정의하여 설계가 수행되는 설계 환경을 정의해 주어야 한다. 설계 소프트웨어들은 설계자가 설계 환경을 정의할 수 있도록 그래픽 사용자 환경(GUI)와 같은 다양한 기능들을 제공하고 있다. 그러나 설계 소프트웨어의 종류에 따라 조금씩 다른 방식들을 사용하고 있으며, 이 방식에 따라 장단점들이 존재하고 있다.

**Table 2** Characteristics of the software packages to control the analysis program

	VisualDOC	iSIGHT	OPTIMUS	ModelCenter
1	○	○	○	○
2	×	○	○	○
3	○	○	○	○
4	×	○	○	×
5	○	×	×	×
6	×	○	×	○
7	×	○	○	○

- 1 : GUI to define design problem
- 2 : Multiple input/output files
- 3 : Definition of user supplied equation
- 4 : Script to define flexible process control
- 5 : Compile process to integrate an analysis software
- 6 : Real time output monitoring
- 7 : Remote access to an analysis software distributed in the network environment

#### 3.1 해석 소프트웨어 제어 기능 비교

설계 소프트웨어들의 해석 소프트웨어 제어를 위한 기능들에 대해 비교해 보았다. 설계 소프트웨어를 이용하여 설계를 수행하기 위해서는 설계 소프트웨어가 해석 소프트웨어를 제어할 수 있도록 해석 소프트웨어를 정의하는 과정이 선행되어야 한다. 모든 설계 소프트웨어들이 그래픽 사용자 환경(GUI)을 사용하는 것과 같은 공통적인 부분을 많이 가지고 있지만, 소프트웨어에 따라 해석 소프트웨어를 제어하고, 데이터를 제어하기 위하여 조금씩 다른 방식을 사용하고 있다. 설계 소프트웨어들의 해석 소프트웨어 제어상의 특징은 Table 2 에 표시하였다.

VisualDOC 은 해석 소프트웨어의 제어 기능을 포함한 설계 환경이 정의되어 있는 함수를 설계자들에게 친숙한 Fortran, C 등의 소스 코드로 생성해 준다. 이 경우, 설계자는 컴파일에 대한 부담을 가지게 된다. 그러나, Fortran 이나 C 와 같은 프로그래밍언어에 익숙한 설계자라면, 함수에 직접 코드를 추가함으로써 목적에 맞도록 설계 환경을 조절할 수 있다는 장점이 있다. 본 연구에서도 필요한 경우, 필요에 따라 이런 코드들을 입력하여 사용하였다. 특히, 승객거동해석 소프트웨어, SAFE 를 이용한 설계 과정에서는 이러한 코드를 변경하여 설계를 수행하였다.

iSIGHT, OPTIMUS, ModelCenter 는 설계자가 컴파일을 해야 한다는 부담이 없도록 소프트웨어가 해석 소프트웨어를 자동적으로 인식하고 제어하는 방식을 사용하고 있다. 설계자는 사용하고자 하는

설계 소프트웨어를 지정해 주면, 부가적인 작업 없이 쉽게 해석 프로그램과 설계 모듈을 연결할 수 있다.

iSIGHT, OPTIMUS 의 경우에는 해석 소프트웨어의 제어를 위해 자체적인 명령어들의 집합인 스크립트 언어를 제공하고 있다. iSIGHT 는 MDOL 이라는 자체 스크립트를 이용하여 유연하고, 다양한 설계 환경을 정의할 수 있도록 하고 있다.<sup>(4,12)</sup> OPTIMUS 의 경우에도 간단한 스크립트를 제공하고 있다.

ModelCenter 는 설계 환경을 정의하기 위해 사용되는 Analysis server 라는 소프트웨어가 존재한다. Analysis server 는 설계 환경을 정의하기 위해 사용되는 소프트웨어로, JAVA 언어로 개발되어 운영체제에 관계없이 작동된다.<sup>(11)</sup> ModelCenter 는 네트워크로 연결된 Analysis server 에서 정의된 설계 환경을 가져와 설계 문제를 정의하고, 설계를 수행하기만 하는 기능을 한다. 이런 특징들 때문에 ModelCenter 는 네트워크 상에 존재하는 해석 소프트웨어를 이용하여 설계를 수행하기에 매우 적합한 소프트웨어이다.

### 3.2 설계 데이터 정의 기능 비교

설계 소프트웨어들은 해석 소프트웨어의 입/출력 파일을 통하여 해석 소프트웨어와 데이터를 교환하고 설계를 수행한다. 이 과정은 설계와 직접 연관된 데이터를 다루는 부분이기 때문에 설계 소프트웨어에서 매우 중요한 부분이다. 해석 소프트웨어는 종류에 따라 정해진 형식에 맞추어 입력 파일을 정의해야 하거나, 출력 파일의 형태가 변하는 경우가 존재한다. 이런 특징들 때문에 설계 소프트웨어가 데이터를 교환하는 부분은 세심한 주의가 필요한 부분이기도 하다.

설계 소프트웨어들은 입/출력 파일을 이용하여 데이터를 교환하기 위해 다음과 같은 두 가지 방법이 사용하고 있다. 데이터를 특수 문자의 형태

로 정의된 템플릿을 이용하는 방식과 파일들에서의 데이터들의 위치를 이용하는 방식이 그것이다. 템플릿을 이용하는 방식은 입력 파일과 같이 형태가 변하지 않는 경우 효율적인 방법이다. 그리고, 데이터들의 위치를 이용하는 방식은 기준이 되는 문자열을 정의하고, 정의된 문자열로부터 데이터를 찾아가는 방식을 사용하면 파일의 형태가 변하는 경우에도 사용될 수 있는 방식이다. Table 3 은 해석 소프트웨어의 입/출력 파일을 통하여 데이터를 공유하는 경우 나타나는 특징들을 나타낸 것이다. 특히 입력 파일의 형식이 정해져 있는 입력 파일을 사용하는 경우에 대해 비교한 것이다.

iSIGHT 의 입/출력 파일을 제어하는 부분은 타 소프트웨어에 비해 많은 장점을 가지고 있다. iSIGHT 는 데이터의 행과 열에 대한 위치를 모두 사용하고, f8.3 과 같이 입력 데이터들의 형식(format)을 정의할 수 있도록 하는 등, 매우 다양한 기능을 가지고 있다. 이런 다양한 기능을 이용하면 대부분의 입/출력 파일에서 데이터들을 무리 없이 제어할 수 있다.

OPTIMUS 는 입력 파일에서는 데이터를 정의하기 위해 템플릿을 사용하고, 출력 파일의 경우에는 데이터들의 위치를 이용하는 방식을 사용한다. 입력 파일은 동일한 모델에 대해서는 형태가 변하는 경우가 없기 때문에 이 방식도 무리가 없다고 판단된다. 그러나, 하나의 행에서 둘 이상의 변수가 존재하는 경우에는 문제가 발생하고 있다. 이 부분에 대해서는 개선이 필요하다고 판단된다.

VisualDOC 을 제외한 모든 설계 소프트웨어는 설계자에 의해 정의된 상대적인 원점이 되는 텍스트를 기준으로 데이터들에 접근하는 방식을 사용하여 출력 파일의 형태가 변하는 경우에도 정확한 결과를 얻을 수 있도록 하였다. VisualDOC 은 이 부분에 관해 수정이 필요하다고 판단된다.

그리고, ModelCenter 는 입력 파일에서 정의되는 설계 변수들이 존재하는 행에서의 위치로써 필드의 정보를 사용하고 있다. 이런 방식 때문에 OPTIMUS 와 마찬가지로 하나의 행에서 둘 이상의 변수가 존재하는 정해진 형식을 가진 입력파일을 제어하는 경우에는 문제가 발생하게 된다. 이 부분에 대해서는 개선이 필요하다고 판단된다.

## 4. 설계 모듈의 성능 비교

설계 환경이 정의 되었다면, 이를 이용하여 설계를 수행해야 한다. 설계 소프트웨어들은 설계자에 의해 좌우되는 설계의 단점들을 극복하고 효율

Table 3 File parsing characteristics of fixed format input file

	VisualDOC	iSIGHT	OPTIMUS	ModelCenter
1	×	○	○	○
2	○	○	×	×
3	○	○	×	○
4	○	○	×	×

1 : Relative position handling in files

2 : Column position handling in files

3 : Format defining for the variables

4 : Multiple variable handling in a line

**Table 4** Design methods of design software

	VisualDOC	iSIGHT	OPTIMUS	ModelCenter
Optimization	○	○	○	○
Design of Experiments	○	○	○	○
Parameter study	○	○	○	○
Quality Engineering	×	○	×	×

적인 설계를 수행하는 것이 목적이다. 설계를 효율적으로 수행할 수 있도록 각 소프트웨어들은 최적설계나 실험계획법 등과 같은 설계에서 많이 사용되고 있는 설계 방식들에 대한 설계 모듈들을 가지고 있다. 각 소프트웨어들은 Table 4 와 같은 몇 가지의 설계방식을 제공하고 있지만, 이번 연구에서는 공통적으로 제공하고 있는 설계 방식인 최적 설계와 실험계획법에 대한 성능을 비교해 보았다. 설계 모듈의 성능을 비교하기 위하여 소프트웨어들이 제공하는 기본 설정을 이용하였다.

4.1 최적설계 모듈의 성능 비교

최적 설계는 최적화 알고리즘들을 이용하여 현재의 설계점에서 성능이 개선되는 설계점을 찾고, 이렇게 개선된 설계점에서 다시 개선된 설계점을 찾는 과정의 반복을 통하여 가장 좋은 설계를 찾아가는 방법이다. 최적 설계에는 많은 다양한 알고리즘들이 존재하고 있고, 소프트웨어들은 이런 알고리즘들 중에서 몇 가지를 제공하고 있다. Table 5 에 각 소프트웨어가 제공하는 알고리즘들을 표시하였다.

VisualDOC 과 ModelCenter 의 경우에는 구속 조건이 있는 경우와 없는 각각의 경우에 대하여 적합한 알고리즘을 선택할 수 있는 방식을 사용하고 있기 때문에, 각각의 경우에 대해 해의 정확성을 높이고, 함수계산의 횟수를 감소 시킬 수 있도록 배려하고 있다. iSIGHT 와 OPTIMUS 의 경우에는 구속조건이 없는 경우 사용되는 알고리즘이 없고, 구속조건의 유무와는 상관없이 사용자가 알고리즘을 선택하도록 하고 있다. 그러므로, 사용자는 문제에 따라 적절한 알고리즘을 선택하여 설계를 수행하여야 한다. iSIGHT 의 경우, 이를 보완하기 위해 Advisor 기능을 제공하고 있다.

최적 설계는 반복적인 함수계산을 이용하여 최적의 설계점을 찾아가는 방식이기 때문에, 최적점을 찾아가갈 때까지 반복적으로 함수계산을 수행하게 된다. 만일 수학적으로 정식화된 문제인 경우나 해석에 필요한 시간이 작은 경우에는 해석의

**Table 5** Optimization algorithm of design software packages<sup>(9-11, 14-20)</sup>

	VisualDOC	iSIGHT	OPTIMUS	ModelCenter
1	BFGS	MFD	SQP	VMM
2	FR	MMFD	GRG	CGM
3	MMFD	SLP		MFD
4	SLP	SQP		SLP
5	SQP	SAM		SQP
6		DHS		
7		EP		
8		GA		
9		HJ		
10		SA		
11		GRG		

- BFGS : Broyden-Fletcher-Goldfarb-Shannon algorithm
- CGM : Conjugate Gradient Method
- DHS : Direct Heuristic Search
- EP : Exterior Penalty
- FR : Fletcher-Reeves algorithm
- GRG : Generalized Reduced Gradient
- GA : Genetic Algorithm
- HJ : Hooke-Jeeves algorithm
- MFD : Method of Feasible Directions
- MMFD: Modified Method of Feasible Directions
- SLP : Sequential Linear Programming
- SQP : Sequential Quadratic Programming
- SA : Simulated Annealing
- SAM : Successive Approximation Method
- VMM : Variable Metric Method

횟수는 중요한 문제가 아니다. 그러나, 해석에 많은 시간이 필요한 복잡한 문제인 경우에는 해석의 횟수는 설계 시간과 직결되는 문제이기 때문에 매우 중요한 부분이라고 할 수 있다. 이런 관점에서 설계 모듈의 성능을 비교하기 위하여 최적설계를 수행하여 얻어진 결과와 최적의 결과를 찾는데 사용된 해석의 횟수를 비교하였다. 몇 가지 수학 문제와 해석 소프트웨어를 이용한 설계 문제들이 설계 모듈의 성능을 비교하기 위하여 이용되었다.

최적 설계 모듈의 성능을 비교하기 위하여 먼저, 식 (1), (2)와 같이 정식화된 수학 문제, 예제 1, 2 에 대해 최적화를 수행해 보았다. 식 (1)은 구속조건이 없는 문제이고, 식 (2)는 구속조건을 가지는 문제이다. VisualDOC 과 ModelCenter 의 경우 구속조건이 있는 문제와 없는 문제에 대해, 그에 적합한 알고리즘을 사용하였으며, iSIGHT 와 OPTIMUS 의 경우에는 일반적으로 많이 사용되는 알고리즘인 MMFD 와 SQP 알고리즘을 이용하였다. Fig. 2, 3 은 최적설계를 통하여 얻어진 설계 결과와 함수계산횟수를 비교한 그림이다.

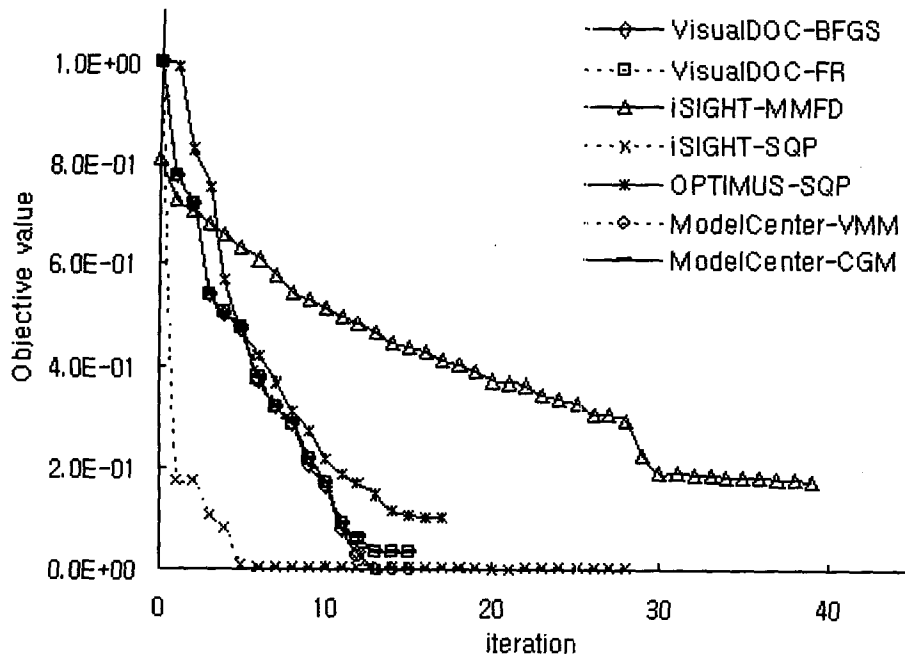


Fig. 2 Optimization results of Rosenbrock's valley problem

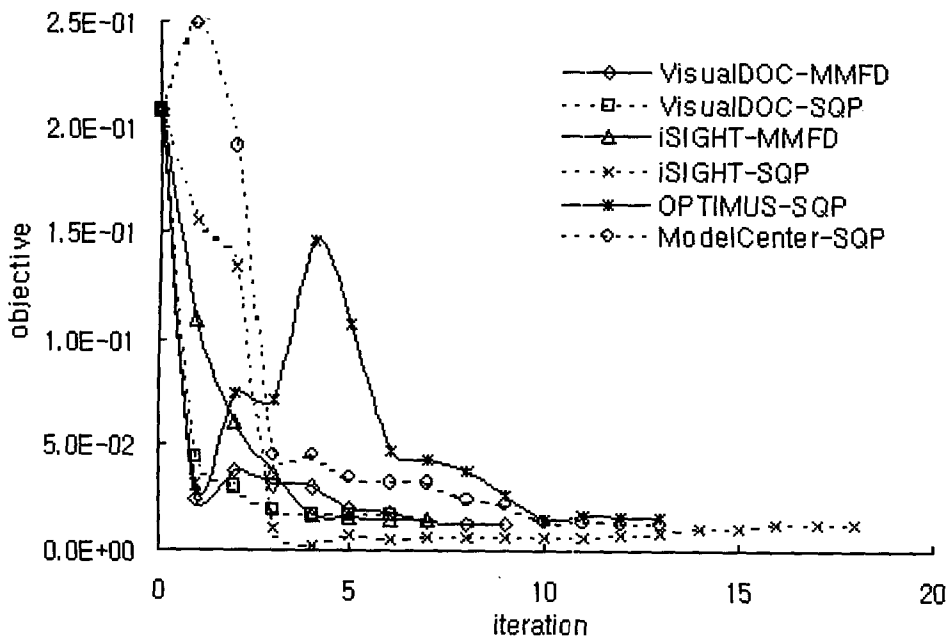


Fig. 3 Optimization results of spring design problem

Example 1. Rosenbrock's valley problem<sup>(21)</sup>

Find  $x_1, x_2$   
 To minimize  $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$  (1)

Example 2. Spring design problem<sup>(9)</sup>

Find  $d, D, N$   
 To minimize  $f = (N + 2)Dd^2$  (2)  
 Subject to  $g_1 = 1.0 - \frac{D^3 N}{71875d^4} \leq 0$   
 $g_2 = \frac{D(4D - d)}{12566d^3(D - d)} + \frac{2.46}{12566d^2} - 1.0 \leq 0$   
 $g_3 = 1.0 - \frac{140.54d}{D^2 N} \leq 0$   
 $g_4 = \frac{D + d}{1.5} - 1.0 \leq 0$

Fig. 2 는 예제 1 에 대한 결과이다. iSIGHT 의 SQP 알고리즘의 경우에는 함수 계산횟수가 다른

알고리즘들에 비해 상대적으로 많은 것을 볼 수 있었다. Fig. 3 은 예제 2 의 결과를 나타낸 그림이다. 두 번째 예제의 경우, 모든 알고리즘이 비슷한 설계 결과를 얻어내고 있다. 그러나, OPTIMUS 의 SQP 알고리즘과, 특히 iSIGHT 의 SQP 알고리즘은 함수 계산횟수가 상당히 많다는 것을 알 수 있다.

다음으로, 실제 해석 소프트웨어와 연결하여 최적설계를 수행한 결과를 비교해 보았다. 예제로는 3 부재 트러스 구조물과 포탈 프레임 구조물이 사용되었다. 두 예제 모두 구속 조건을 가지고 있고, 질량을 최소화하는 문제이다. 3 부재 구조물에 대한 최적화 문제는 식 (3)과 같이 정식화 하였고, 포탈 프레임 구조물에 대한 최적화 문제는 식 (4)와 같이 정식화 하였다. 3 부재 트러스 구조물은 초기 설계가 구속 조건을 만족하고, 포탈 프레임 구조물의 경우 초기 설계가 구속 조건을 위배하고 있다는 차이가 있다. 이상의 두 예제들도 수학 문제들의 경우에서와 같이 최적 설계로 얻어진 결과와 함수계산횟수를 비교하였다. 해석 소프트웨어로는 Genesis 6.0<sup>(22)</sup>을 사용하였다.

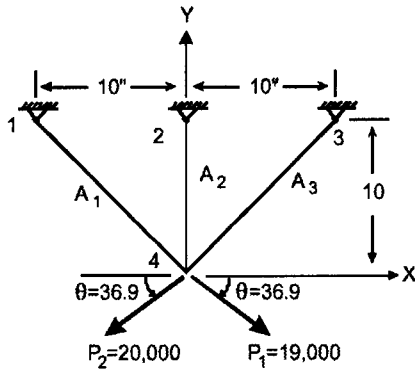


Fig. 4 Design problem of three bar truss

Example 3. Optimization problem for three bar truss structure<sup>(9,22)</sup>

Find  $A_1, A_2, A_3$   
 To minimize Mass (3)  
 Subject to  $-15000 \leq \sigma_i \leq 20000 \quad i = 1, 2, 3$   
 $-2.0 \leq \delta_{4,j} \leq 2.0 \quad j = x, y$

where

$\sigma_{ij}$  is the stresses of each elements

$\delta_3$  is the displacement in the x and y direction at grid point 4

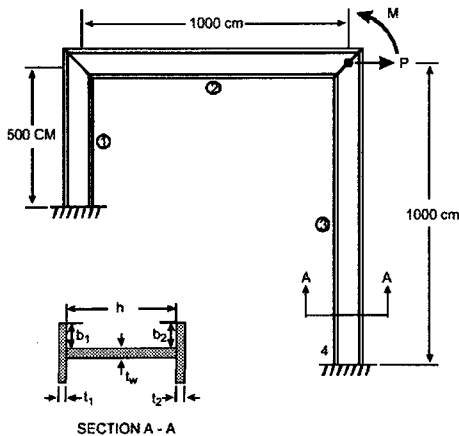


Fig. 5 Design problem of portal frame structure

Example 4. Optimization problem for portal frame structure<sup>(9,22)</sup>

Find Shape of crosssections of the frame (4)  
 To minimize Mass  
 Subject to  $-20000 < \sigma_y < 20000 \quad j = 1, 2, 3, j = 1, 2$   
 $-4.0 < \delta_3 < 4.0$   
 $-0.015 < \theta_3 < 0.015$

where

$\sigma_{ij}$  is the axial stresses on the top and bottom at each end of the elements

$\delta_3$  is the displacement in the x-direction at grid point 3

$\theta_3$  is the rotation in the z-direction at grid point 3

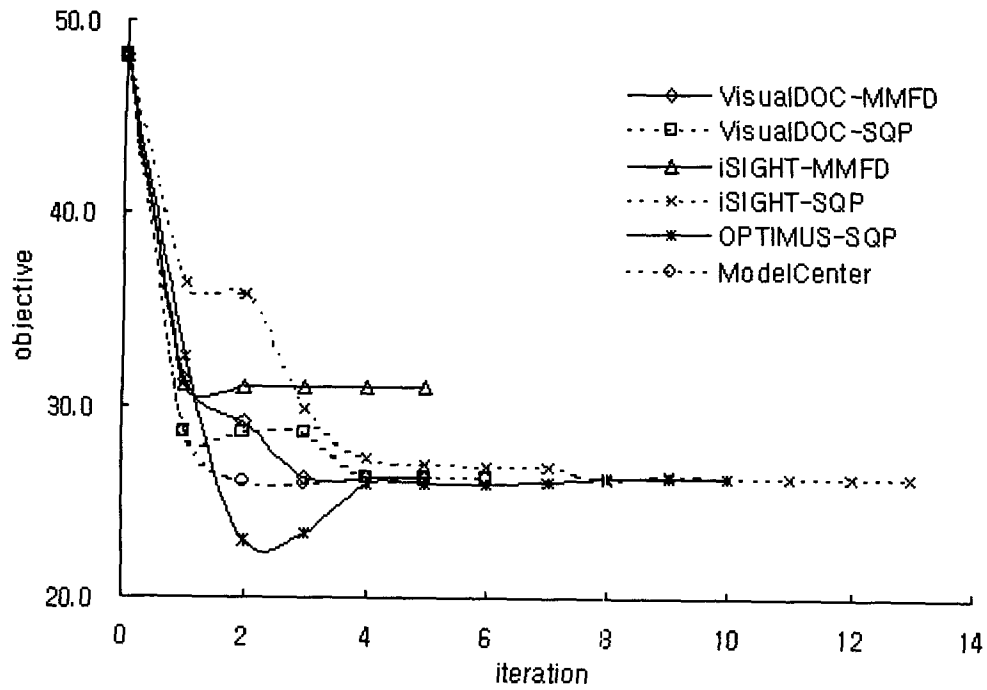


Fig. 6 Optimization results of the three bar truss

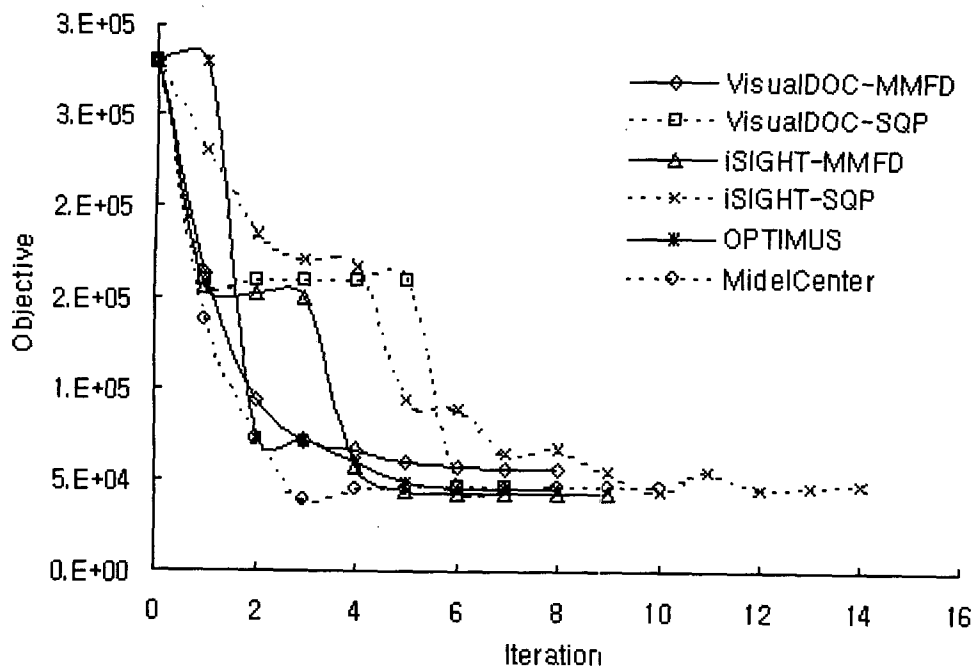


Fig. 7 Optimization results of the portal frame



Fig. 6 과 Fig. 7 에 설계 결과를 표시하였다. 두 결과는 수학 예제들에서의 결과와 비슷한 양상을 나타낸다. Fig. 6 는 3 부재 트러스 구조물의 최적 설계 결과이다. VisualDOC, OPTIMUS, ModelCenter 의 경우 거의 비슷한 결과를 얻고 있다. 그러나, 이 경우에는 iSIGHT 의 경우의 결과가 상대적으로 좋지 못하며, 해석 횟수도 많은 것을 알 수 있다. Fig. 7 의 포탈 프레임 구조물의 최적화 결과에서도 비슷한 결과를 보여주고 있다.

마지막으로, 실제 차량 부품 설계과정에서 설계 소프트웨어들의 성능을 비교해 보았다. 특히, 승객 거동해석을 통하여 승객의 상해를 감소시킬 수 있도록 제품을 설계하는 문제에 설계 소프트웨어를 적용하여 제품을 설계하고, 결과를 살펴 보았다. 설계 소프트웨어의 성능을 비교하기 위하여 조향장치 설계 문제를 이용하였다.

승객거동해석을 위해 SAFE(Safety Analysis For crash Environment)를 이용하였다.<sup>(26)</sup> Fig. 8 은 충돌시 승객이 받는 충격 하중을 얻기 위하여 SAFE 를 이용하여 모델링 된 모형몸체 실험을 나타낸 것이다. SAFE 는 정해진 형식에 맞도록 입력 파일을 정의해 해야 하는 방식으로, 형식이 맞지 않는 경우에는 에러가 발생하고, 프로그램이 실행 되지 않는다. SAFE 와 같이 입력 파일을 형식에 맞도록 정의해야 하는 경우는 설계 소프트웨어와 해석 소프트웨어를 연결하기 가장 어려운 경우로, 설계 소프트웨어들의 성능을 비교하는데 하나의 지표가

될 수 있다. OPTIMUS 와 ModelCenter 의 경우에는 정해진 형식에 따라 입력 파일을 정의해야 하는 경우, 하나의 행에 둘 이상의 데이터가 존재하는 경우를 제어할 수 없기 때문에 이번 예제에는 적용하지 못하였다.

조향장치는 안전벨트를 착용하고 있지 않은 상태에서 승객과 충돌하는 경우, 상측과 하측,상주와 하주가 연결되어 있는 부분이 파손되면서 충돌 에너지를 흡수하여 승객의 상해를 줄여주는 역할을 한다. 그러므로, 에너지 흡수 특성은 조향장치의 중요한 부분중의 하나라고 할 수 있다. 이런 관점에서, 에너지 흡수 특성을 향상 시킬 수 있도록 하는 조향장치를 설계하는 문제를 예제로 선정하였다. 조향장치의 에너지 흡수 특성 향상을 위하여 몰딩부, 플레이트부, 칼럼압입부 및 고무 플랜지부의 힘-변형 곡선의 스케일 팩터(Scale factor) 를 설계변수로 선정하였다. 그리고, 충돌시 승객을 단순화하여 표현한 모형 몸체에 작용하는 충격하중을 목적함수로 선정하여, 이를 최소화하는 문제를 식 (3)과 같이 정식화하였다. 그리고, 모형 몸체실험을 통하여 얻어지는 결과들을 설계에 이용하였다.<sup>(23-25)</sup>

승객거동해석의 경우에는 비선형성이 매우 강하기 때문에, 시스템의 민감도를 이용하는 최적설계 방법으로 설계를 수행하는데 무리가 있다.<sup>(27)</sup> 따라서, 이런 경우에는 민감도 정보를 이용하지 않는 설계 방법을 이용하는 설계 방법을 사용하여야 한다. VisualDOC 의 경우 변형된 2 차 근사법의 형태의 반응표면법(Response Surface Method) 알고리즘이 존재한다. 이 알고리즘은 실험계획법에서 사용하는 반응표면법과 동일한 명칭을 가지고 있지만, 내용상의 차이가 있다.<sup>(28)</sup> iSIGHT 의 경우에도, 실험계획법에서 설명하는 반응표면법을 반복적으로 사용하여, 민감도 정보를 이용하지 않고 최적설계를 수행하는 알고리즘을 가지고 있다. iSIGHT 는 또한 진화 알고리즘(Genetic algorithm)과 같은 민감도 정보를 이용하지 않는 알고리즘들이

Example 5. Optimization problem for steering column system

- Find FD curves of energy absorbing parts (3)
- To minimize Impact load of body block
- Subject to Contact force of body block  $\leq F_{all}$

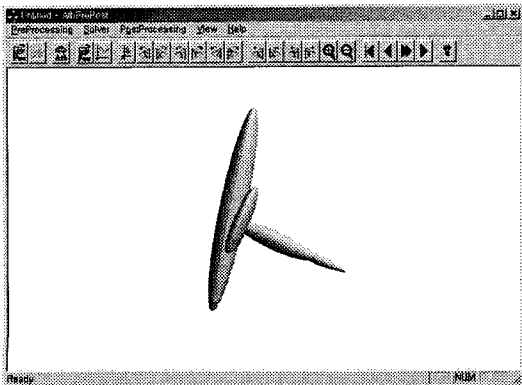


Fig. 8 Body block test to design steering column with SAFE

Table 6 Optimization result of steering column problem with SAFE

	Initial	VisualDOC	iSIGHT
Design Variables	S03	1.0000	0.4320
	S06	1.0000	0.3000
	S11	1.0000	0.3600
	S12	1.0000	0.3000
Objective	1.000	0.883	0.921
Constraint	1.000	0.989	0.990
Iteration	/	33	21

존재하지만 이들은 함수 계산횟수가 너무 많기 때문에 이 알고리즘들을 이용하지 않았다. 본 예제에서는 위의 두 가지 방법을 이용하여 조향장치를 설계하였고, 그 결과를 Table 6에 표시하였다. Table 6에 표시된 결과들은 초기값에 대해 정규화되었다.

결과를 살펴 보면, 정도의 차이는 있지만, 두 경우 모두 성능의 개선을 가져오고 있는 것을 볼 수 있다. 이를 통해 두 알고리즘 모두 비선형성이 강한 문제에서도 좋은 결과를 얻을 수 있을 것으로 판단된다. 또한, VisualDOC의 경우, 함수 계산 횟수가 iSIGHT에 비해 조금 많지만, 목적함수가 상대적으로 많이 줄어든 것을 볼 수 있다.

위의 수학 예제와 공학적 문제들의 결과를 살펴 보면, VisualDOC의 최적설계 알고리즘들과 iSIGHT의 SQP 알고리즘이 전반적으로 좋은 결과를 얻어내고 있는 것을 알 수 있다. 하지만 iSIGHT의 SQP 알고리즘의 경우에는 함수 계산의 횟수가 상당히 많은 특징을 가지고 있다는 것을 알 수 있었다. 전체적인 결과를 통하여 살펴보면 함수계산횟수와 해의 정확성 관점에서 최적설계모듈의 성능은 VisualDOC이 가장 우수하다는 결과를 얻을 수 있다.

4.2 실험계획법 모듈의 성능 비교

최적 설계 이외에도 체계적으로 정립된 설계 방식들이 존재한다. 그 중 하나가 실험 계획법이다. 실험계획법은 설계 대상인 모델을 잘 표현할 수 있도록 하는 최소의 실험 점들을 배치하고, 이 점들에서의 결과를 통계적으로 분석하여 설계를 수행하는 방식이다. 설계 소프트웨어들은 모두 실험계획법을 이용한 설계 모듈을 가지고 있다.

실험 계획법의 경우, 데이터들을 통계적으로 분석하여 결과를 얻는 과정이기 때문에 모든 소프트

Table 7 Supplied type of experiments

	VisualDOC	iSIGHT	OPTIMUS	Model-Center
Full/Fractional Factorial	○	○	○	×
Central Composite Design	○	○	○	○
Orthogonal Arrays	○	○	○	×
Latin Hypercubes	×	○	○	×
Box-Behnken	○	×	○	×
Plackatt-Burman	○	×	○	×
User Define	○	○	○	○
Etc	○	○	○	×

웨어에서 동일한 입력에 대해 동일한 결과를 나타낸다.<sup>(29-31)</sup> 따라서 성능을 비교하는 것은 의미가 없다. 실험 계획법을 이용하여 설계를 수행하기 위해서는 데이터를 수집하기 위하여 사용되는 실험 배치와 얻어진 결과를 분석하는 과정이 중요하다. 이런 관점에서 각 소프트웨어들이 제공하는 실험 배치 방법들과 통계적 분석을 통하여 얻어지는 결과들의 종류를 비교하였다.

Table 7과 8은 소프트웨어들이 제공하는 실험 배치 방법들의 종류와 설계자가 설계대상모델의 특성을 파악할 수 있도록 하기 위하여 제공되는 설계대상모델에 대한 정보들을 표시한 것이다. 표들을 통하여 VisualDOC과 iSIGHT, OPTIMUS는 많은 종류의 실험 배치법을 제공하고 있는 것을 알 수 있다. 그리고, 통계적인 처리를 통하여 얻어진 결과들을 살펴보면 iSIGHT의 경우 설계에 필요한 많은 정보를 설계자에게 제공하고 있는 것을 볼 수 있다. 위의 결과들을 통하여 살펴보면, 실험 계획법을 이용하여 설계를 수행하는 경우에는 iSIGHT가 상대적으로 장점을 가지고 있다는 결론을 얻을 수 있다.

5. 소프트웨어들의 한계

최근 다양한 분야에 대한 해석 소프트웨어들이 많이 개발되고, 또한 다분야 통합설계(MDO)와 같이 여러 분야에 대해 동시에 고려하여 설계 하는

Table 8 Outputs of DOE(Design Of Experiments)

	VisualDOC	iSIGHT	OPTIMUS	Model-Center
ANOVA	○	○	×	×
Main Effects	×	○	×	○
Interaction Effects	×	○	×	×
Scatter	○	○	○	×
Pareto graph	×	○	×	×
Contribution	○	○	○	×

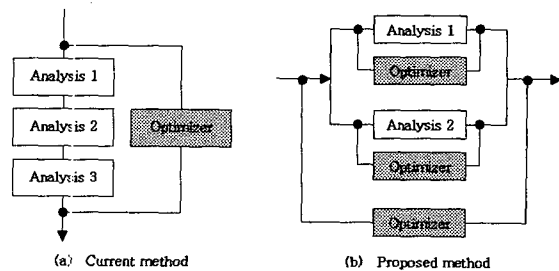


Fig. 9 Comparison of current method and proposed method

경우도 많이 발생하고 있다. 즉, 구조 해석과 유체 해석이 동시에 고려되어야 한다면, 구조 해석과 진동 해석이 동시에 이루어져야 하는 경우와 같은 설계에 대한 연구가 활발히 이루어지고 있다.

비교 대상이 된 소프트웨어들 역시 여러 종류의 해석 소프트웨어들을 연결하여 설계를 수행할 수 있도록 제작되어 있다. 그러나, Fig. 9의 (a)와 같이 설계 문제를 정의하는 과정에서 단지 하나의 설계 모듈만을 연결할 수 있다. 즉 여러 분야의 소프트웨어들의 입력과 출력의 상호관계를 정의하여 하나의 설계 환경을 정의하고, 정의된 설계 환경에 하나의 설계 모듈만을 연결하여 설계를 수행할 수 있도록 하는 방식을 사용하고 있다.

그러나, 문제의 특징에 따라 Fig. 9의 (b)와 같이 각각의 해석 모듈에 대해 설계 모듈을 정의하고, 이렇게 정의된 두 종류의 설계 환경에 대하여 설계를 수행해야 하는 경우도 발생할 수 있다. 그러나, 현재의 소프트웨어들로는 이러한 설계 과정을 직접적으로 수행할 수 없다. 다분야 통합 최적설계에 대한 연구가 활발하게 이루어지고 있고, 이런 부분에 대한 요구가 빈번히 발생하고 있다는 점에서, 설계 소프트웨어의 설계에 이런 부분도 고려되어야 할 필요가 있다고 판단된다.

## 6. 결론

설계를 위한 모듈이 없는 해석 소프트웨어와 연결하여 설계를 할 수 있도록 설계된 상용 설계 소프트웨어 VisualDOC, iSIGHT, OPTIMUS, ModelCenter의 특징을 살펴보고, 그 성능을 비교해 보았다. 설계 소프트웨어들의 설계 환경을 정의하는 부분과 설계 데이터를 정의하는 부분에서의 각 설계 소프트웨어들의 특징을 살펴보고 이들의 장단점을 비교해 보았다. 그리고 설계 소프트웨어들의 성능을 비교해 보았다. 특히, 수학적 예제와 공학적 예제들을 통하여 얻어진 설계 결과와 함수계산횟수를 통하여 최적 설계 모듈들의 성능에 대해 중점적으로 비교해 보았다.

(1) VisualDOC은 최적 설계에 비중을 두고 있는 소프트웨어로서 최적설계 모듈의 성능이 매우 뛰어난 장점을 가지고 있었다. 그러나, 사용자 환경이나, 해석 프로그램과의 연결 부분에서 몇 가지 문제점들이 존재하고 있고, 이에 대한 개선이 필요하다고 판단된다.

(2) iSIGHT는 설계자의 사용상의 편의성이나 사용자 환경이 매우 우수하여, 설계자가 소프트웨어를 쉽게 사용할 수 있는 특징이 있었다. 그리고, 실험계획법을 이용한 설계 모듈에서 설계 모델에

대한 다양한 정보를 제공한다는 장점을 가지고 있었다. 그러나, 최적 설계 모듈의 경우 수렴 조건등의 값들에 대한 조정을 통해 최적설계 모듈의 성능을 개선할 필요가 있다고 판단된다.

(3) OPTIMUS의 경우에는 설계 변수와 입력 파일, 해석 소프트웨어, 출력 파일과 결과값들의 관계를 도식적으로 표현하여 사용자가 정의된 문제의 구조를 쉽게 파악할 수 있도록 구성되어 있다는 장점을 가지고 있었다. 그러나 입/출력 파일을 통하여 설계 데이터를 교환하는 부분에서 존재하는 몇 가지 문제점들에 대한 개선이 필요하다고 판단된다.

(4) ModelCenter는 해석 프로그램과 연결을 담당하는 부분과 설계 문제를 정의하는 부분을 분리하여, 서로 다른 소프트웨어로 구성하고 있는 특징이 있었다. 이런 방식을 이용함으로써 네트워크 상에 존재하는 여러 프로그램들을 쉽게 연결할 수 있도록 만들어져 있다는 장점이 있었다. ModelCenter의 경우에도 입/출력 파일을 통하여 데이터를 교환하는 부분에 대한 좀더 세심한 고려가 필요하다고 판단된다.

마지막으로, 비교 대상이 된 소프트웨어들은 하나의 설계 환경 내에서는 단지 하나의 설계 모듈만을 사용하는 방식을 사용하고 있다는 사실을 알 수 있었다. 최근 연구가 활발히 이루어지고 있는 다분야 통합 설계의 경우에는 하나의 설계 문제 내부에서 여러 개의 설계 모듈이 사용되는 경우도 존재하고 있다. 이런 문제들의 경우, 비교 대상 소프트웨어들로부터 직접적으로 정의될 수 없었다. 설계 소프트웨어들은 이런 부분에 대해서 고려해야 할 필요가 있다고 판단된다.

## 후 기

이 연구는 한국과학재단지정 최적설계신기술센터와 교육부 지원 BK21 사업 연구비 지원으로 수행되었습니다.

## 참고문헌

- (1) Arora, J. S., 1989, *Introduction to Optimum Design*, McGraw-Hill book company, International edition.
- (2) Sobieszczyński-Sobieski, J. and Haftka, R.T., 1996, "Multidisciplinary Aerospace Design Optimization : Survey of Recent Developments," AIAA Paper 96-0711, pp. 1~32.
- (3) VisualDOC Manual : version 1.0, Vanderplaats

- Research & Development, Inc., 1767 S. 8th Street, Suit M-210, Colorado Springs, CO, 1998.
- (4) iSIGHT Designer' s Guide : version 5.0, Engineous Software, Inc., 2000 Centregreen Way, Suite 100, Cary, NC 27513, North Carolina, 1999.
- (5) OPTIMUS Users Manual : version 2.1, LMS International, Inc., Interleuvenlaan 68-70, B-3001 LEUVEN, BELGIUM., 1999.
- (6) Model Center Training and Help, Phoenix Integration, Inc., 1750 Kraft Drive, Suite 2200, Blacksburg, Virginia, 1999.
- (7) Ghosh, D.K., Garcelon, J.H., Balabanov, V.O., Venter, G. and Vanderplaats, G.N., 2000, "VisualDOC - A Flexible Design Optimization Software System," presented at the 8<sup>th</sup> AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA.
- (8) Ghosh, D.K., Garcelon, J.H., Balabanov, V.O. and Vanderplaats, G.N., 1999, "Development Of A Flexible Design Optimization Study Tool," presented at the 7<sup>th</sup> AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louse, MO.
- (9) Haug, E.J. and Arora, J.S., 1979, *Applied Optimal Design*, Wiley & Sons, New York, U.S.A.
- (10) Haftka, R.T. and Gurdal, Z., 1991, *Elements of Structural Optimization*, Kluwer Academic Publishers, The Netherlands.
- (11) Rao, S. S., 1996, *Engineering Optimization Theory and Practice*, Third Edition, John Wiley and Sons Inc., U.S.A.
- (12) iSIGHT Developer' s Guide Volume 2 MDOL : version 5.0, Engineous Software, Inc., 2000 Centregreen Way, Suite 100, Cary, NC 27513, North Carolina, 1999
- (13) Brett Malone and Michael Papay, 1999, "ModelCenter: An Integration Environment for Simulation Based Design," SISO Workshop.
- (14) Bradley, S.P., Hax, A.C. and Magnanti, T.L., 1977, *Applied Mathematical Programming*, Addison-Wesley, Mass.
- (15) Fletcher, R., 1970, "A New Approach to Variable Metric Algorithms," *Computer J.*, Vol 13, pp. 317~322.
- (16) Vanderplaats, G.N. and Sugimoto, H., 1985, "Application of Variable Metric Methods to Structural Synthesis," *Engineering Computations*, Vol. 2, No 2.
- (17) Zoutendijk, K.G., 1960, *Methods of Feasible Directions*, Elsevier Publishing Co., Amsterdam, Netherlands.
- (18) Vanderplaats, G. N., 1984, "An Efficient Feasible Direction Algorithm for Design Syn-thesis," *AIAA Journal*, Vol. 22, No. 11.
- (19) Fletcher, R. and Powell, M.J.D., 1964, "Function Minimization by Conjugate Gradients," *The Computer Journal*, Vol. 7, pp. 149~160.
- (20) Thanedar, P.B., Arora, J.S., Tseng, C.H., Lim, O.K., and Park, G.J., 1987, "Performance of Some SQP Algorithms on Structural Design Problems," *International Journal for Numerical Methods in Engineering*, Vol. 23, No. 12, pp. 2187~2203.
- (21) Kroo, I., Altus, S., Braun, R., Gage, P., Sobieski, I., 1994, "Multidisciplinary Optimization Methods For Aircraft Preliminary Design," AIAA Paper 94-4325, pp. 697~707.
- (22) Genesis User Manual : version 5.0, Vanderplaats Research & Development, Inc., 1767 S. 8th Street, Suit M-210, Colorado Springs, CO, 1998.
- (23) Park, Y.S., Lee, J.Y., Lim J.M., Park, G.J., 1996, "Optimum Design of a Steering Column to Minimize the Injury of Passenger," *International Journal of Vehicle Design*, Vol. 17, No. 4.
- (24) 박영선, 이주영, 박경진, 1995, "승객상해의 감소를 위한 승용차 조향주의 최적설계," 한국자동차공학회논문집 Vol. 3, No. 1, pp. 33~44.
- (25) 신문균, 홍성우, 박경진, 2000, "안전도를 고려한 전동 텔트/텔레스코픽 조향주의 해석 및 설계," 대한기계학회논문집, 제 24 권, 6 호, pp. 1479~1490.
- (26) 임재문, 1997, "차량충돌시 승객거동해석을 위한 소프트웨어의 개발 및 관련 부품의 설계방안 제시," 한양대학교 박사학위 청구논문.
- (27) Bennett, J.A. and Park, G.J., 1995, "Automotive Occupant Dynamics Optimization," *Shock and Vibration*, Vol. 2, No. 6, pp. 471~479.
- (28) Vanderplaats, G. N., 1984, *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill book company, New York.
- (29) 박성현, 1991, "현대실험계획법," 민영사.
- (30) Phadke, Madhav S., 1989, *Quality Engineering Using Robust Design*, Prantice Hall, Englewood Cliffs, New Jersey 07632.
- (31) 이권희, 박경진, 1998, "제한조건이 있는 문제의 이산설계 공간에서의 강건 최적설계," 대한기계학회논문집, 제 22 권, 제 5 호, pp. 728~737.
- (32) DOC Users Manual, Vanderplaats Research & Development, Inc., 1767 S. 8th Street, Suit M-210, Colorado Springs, CO, 1995.
- (33) DOT Users Manual, Vanderplaats Research & Development, Inc., 1767 S. 8th Street, Suit M-210, Colorado Springs, CO, 1995.
- (34) Guy Gendron, 1997, "A Review of Four PC Packages for FE Structural Analysis," *Finite Elements in Analysis and Design*, Vol. 28, pp. 105~114.