

〈 논문 〉

엑셀 인터페이스 관점에서의 최적설계 프레임워크의 벤치마크

염 균 철^{*} · 이 세 정[†]

(2004년 6월 8일 접수, 2004년 10월 15일 심사완료)

Benchmarking of Design Optimization Frameworks In View of Excel Interface

Keun-Chul Yum and Se Jung Lee

Key Words : Optimization(최적설계), Design Framework(설계프레임워크), Software Development(소프트웨어개발), Excel(엑셀)

Abstract

This study presents the benchmark results of several design frameworks for the purpose of developing an efficient design framework in view of Excel interface. For the benchmark, three optimum design problems are chosen and solved by using the design frameworks. In order to compare the frameworks with each other, the evaluation criteria are specified and modified to fit the problems with Excel interface. Three example cases are solved and compared. The evaluation report is summarized in terms of efficiency, ease of use, internal structure, API, etc. However, the optimization result itself is not focused because the specific optimization algorithm is not our concern. Finally, some suggestions of how to develop an efficient design framework with Excel interface are presented.

1. 서 론

Park 등⁽¹⁾과 같은 새로운 최적설계 이론을 기계 설계에 적용하기 위해서는 설계자가 저수준의 컴퓨터 프로그래밍을 통하여 설계 절차를 구성해야 하는데 이 과정이 복잡하며, 숙련된 컴퓨터 프로그래밍 기술이 필요하다. 이를 해결하기 위해 최적화 프로그램과 다양한 해석 프로그램을 설계자의 의도대로, 논리적인 절차를 사용하여 설계 과정을 통합, 자동화하는 환경인 최적설계 프레임워크가 개발되었다. 즉, 최적설계 프레임워크는 Fig.

1에서와 같이 원격에 위치한 최적설계 모듈, 데이터베이스, CAD 그리고 각종 해석 프로그램과 같은 설계 자원들을 분산 컴퓨팅 환경에서 엔지니어가 쉽게 연결하고 이를 자동 실행 시켜 최적설계를 수행하는 메타 소프트웨어이다.

현재 이러한 기능의 최적설계 프레임워크로는 국내 한양대학교 최적설계신기술연구센터에서 프레임워크 초기 버전⁽²⁾을 2003년에 개발하였고, 국외에서는 iSIGHT⁽³⁾, ModelCenter⁽⁴⁾, VisualDOC⁽⁵⁾ 등이 개발되었다. 또한 이러한 프레임워크들의 일반적 성능을 비교한다든지, 또는 새로운 프레임워크의 개발을 위한 연구가 최근에 다수 발표되었다.⁽⁶⁻⁹⁾

본 논문의 주안점은 설계프레임워크를 이용하여 제품에 대한 최적설계를 수행할 경우 설계 데이터를 저장하고 이를 관리하는 방법에 관한 것이다. 이것이 중요한 이유는 최적설계를 수행하면서 생성된 설계 데이터의 변화 추이를 이용하여,

* 서울시립대학교 대학원 기계정보공학과

[†] 책임저자, 회원, 서울시립대학교 기계정보공학과

E-mail : selee@uos.ac.kr

TEL : (02)2210-2537 FAX : (02)2248-5110

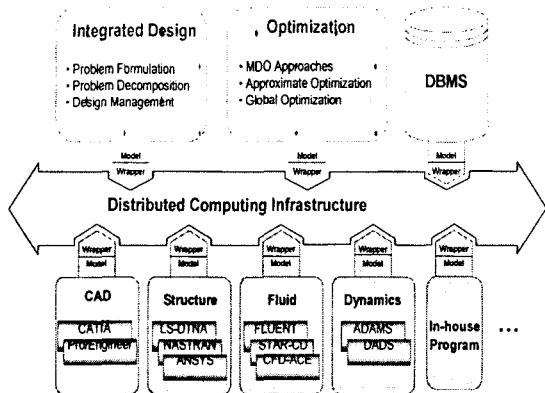


Fig. 1 The concept of design optimization framework

엔지니어가 최적설계의 적합성을 판단할 수 있으며 최적설계 완료 후 만들어진 결과 데이터보다 향상된 값을 최적설계 중간에 생성된 데이터에서 얻을 수도 있기 때문이다.

이를 위하여 대부분의 최적설계 프레임워크에서는 손쉬운 데이터 저장 방식인 파일 시스템을 사용하고 있다. 파일 시스템은 설계 데이터를 기록하기 쉽고, 저장에 필요한 비용이 적게 들며, 관리가 용이하다. 하지만 데이터의 부분 또는 전부의 가공이나 재사용에 부적합한 단점이 있다. 이러한 문제점 때문에 엑셀과 같이 데이터의 저장 및 가공이 용이한 프로그램을 이용하여 최적설계에서 생성된 데이터를 저장하고 관리하기를 원한다. 그래서 현재 개발되어 상용화된 최적설계 프레임워크인 iSIGHT, ModelCenter, VisualDOC 등은 엑셀을 이용한 데이터 관리 방법을 지원하고 있다.

이 기능을 이용하면 엑셀에 저장된 설계 데이터를 저수준의 컴퓨터 프로그래밍 없이 최적설계 프레임워크에서 사용할 수 있다. 그런데 각각의 최적설계 프레임워크가 지원하는 엑셀 인터페이스 방식의 차이로 인하여 프레임워크를 이용한 문제 구성의 작업 효율성에 차이를 보일 뿐 아니라 각각의 최적설계 프레임워크가 엔지니어가 실제로 원하는 엑셀 인터페이스의 기능을 정확히 반영하지 못하고 있다. 따라서 본 논문에서는 현재 가장 널리 사용되고 있는 세 개의 프레임워크, iSIGHT, ModelCenter, VisualDOC 을 선정하여, 엑셀 인터페이스를 평가하기 위해 다섯 개의 평가 지표를 만들고 예제 문제를 수행하여 비교, 평가 함으로써 향후 개발되는 최적설계 프레임워크가 보다 향상된 엑셀 인터페이스 기능을 가질 수 있도록 방법을 제안하고자 한다.

2. 평가 지표의 선정

개발 초기단계에 프레임워크의 설계를 위한 요구 사항에 대한 연구⁽¹⁰⁾가 이루어져 기본적으로 프레임워크가 갖추어야 할 요건에 대하여 공개되어 있지만, 그 이후 컴퓨터 환경의 변화가 급진적으로 이루어졌으며, 본 논문의 관심사가 엑셀 인터페이스에 한정되어 있으므로 기존의 평가지표를 수정하고 보완하여 사용한다. 아래에 엑셀 인터페이스에 적용시키기 위한 다섯 가지 평가 지표를 설명한다.

2.1 사용 편의성

최적설계 프레임워크가 엑셀 인터페이스를 지원하는 방식을 분류하면, GUI를 이용하여 스크립트 파일을 만드는 형식과 사용자가 GUI의 지원 없이 스크립트 파일을 만드는 형식으로 나누어 볼 수 있다. 엑셀에 저장된 설계 데이터의 양이 방대할 경우 사용자의 숙련도와 GUI의 기능에 따라 프레임워크에서 사용할 엑셀 컴포넌트를 만드는 노력과 시간에 많은 영향을 미친다. 따라서 현재 개발된 프레임워크의 엑셀 인터페이스 방식을 비교 평가하여 더 효율적인 방법을 모색하고자 한다.

2.2 기능성

엑셀에 저장된 데이터를 이용하는 데 있어서 하나의 워크북, 한 워크 쉬트에 저장된 데이터만을 이용하지는 않는다. 여러 워크북에, 여러 쉬트에 저장되어 있는 데이터를 사용할 수도 있다 따라서 본 논문에서 비교할 프레임워크의 엑셀 인터페이스도 이러한 요구를 만족 시킬 수 있는 기능이 있는지 확인한다.

2.3 후처리 기능

저장된 데이터를 이용하여 설계 작업을 수행했다면 설계 데이터의 이력 및 결과 값을 엑셀에 저장함으로써 사용자가 조금 더 효율적으로 설계 작업을 완성 할 수 있을 것이다. 따라서 비교 대상인 참조 최적설계 프레임워크를 사용하여 만들어진 결과물을 자동으로 엑셀의 차트기능 등을 이용하여 나타낼 수 있는지가 세 번째 평가 지표이다.

2.4 외부 해석기와의 연결성

엑셀의 함수 기능만을 이용하여 설계 작업을 수행하는 데는 한계가 있다. 따라서 실제로 설계 작업을 수행하는데 있어서 해석기의 사용은 중요하다. 엑셀의 함수 기능을 이용하여 계산된 설계 테

이터를 해석기에 자동적으로 넘겨 주어 설계 작업을 수행하는 것이 가능한지를 벤치마킹 지표로 넣어 참조 프레임웍에서 확인한다.

2.5 사용을 위한 지원 능력

엔지니어가 최적설계 프레임웍의 엑셀 인터페이스를 사용하기 위해서는 쉽게 기술된 안내서는 필수적이다. 따라서 참조 프레임웍의 안내서를 비교하여 그 결과를 지표의 하나로 정하였다.

3. 평가를 위한 설계문제의 선정

각 프레임웍의 엑셀 인터페이스를 비교, 평가하기 위한 설계 문제를 선정하였는데, 이 문제들은 기존에 최적설계가 시도되어 결과 값들이 검증되어 있는 문제들이므로 엑셀 인터페이스의 기능을 평가하기에 충분하다. 선정된 설계 문제를 해결하기 위해 엑셀의 역할과 해석기의 종류에 따라서 세 가지로 나누었다.

3.1 코일 스프링 최적화 문제

문제 1은 Fig. 2와 같이 비틀림 모멘트와 인장하중을 받고 있는 스프링의 무게를 최소화 하는 문제로서 스프링 와이어 지름과 코일 지름을 결정하는 최적설계 문제이다.⁽¹¹⁾

이 문제에서는 외부 해석기 없이 엑셀에 저장되어 있는 데이터만을 이용하여 최적설계를 수행한다. 즉, 엑셀에 설계에 필요한 데이터가 있고, 이 데이터를 가지고 최적설계를 위한 목적함수를 계산하여 원하는 설계변수 값을 얻는다. 이러한 형태는 엑셀 인터페이스의 가장 기본적인 문제 구성이라고 할 수 있다.

3.2 10부재 트러스 최적화 문제

문제 2는 Fig. 3의 구조물이 하중을 받을 때 주어진 허용 응력과 허용 범위를 만족하면서 전체 트러스의 중량을 최소화하는 각 부재의 단면적을 구하는 문제이다.⁽¹²⁾

문제 2에서는 엑셀에 설계에 필요한 모든 데이터를 저장하고 상용 구조해석 프로그램인 ANSYS를 이용하여 해석을 한다. 이것은 설계 데이터의 이동 및 전달의 중심에 엑셀을 두고 여기에서 설계 데이터를 얻어서 ANSYS에서 필요한 입력 파일을 만들고, 해석을 수행하며 출력 파일로부터 의미 있는 값을 얻어 엑셀에 저장한다. 결국 엑셀에 저장된 변화된 데이터를 이용하여 최적설계를 수행하게 되는 것이다.

3.3 브레이크 패드 비용 최적화 문제

문제 3은 Fig. 4와 같이 브레이크에 사용되는 브레이크 패드의 비용을 최소화하기 위한 브레이크 패드 넓이, 중심 길이 그리고 두께를 구하는 최적설계 문제이다.⁽³⁾

이 문제에서는 엑셀이 한 해석 분야의 해석기가 되어 세 개의 실행 파일과 연결되어 해석을 수행하게 된다. 문제 2에서는 모든 설계 데이터를



Fig. 2 Coil spring optimization

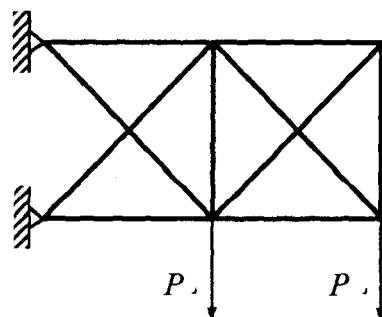


Fig. 3 Ten bar truss optimization

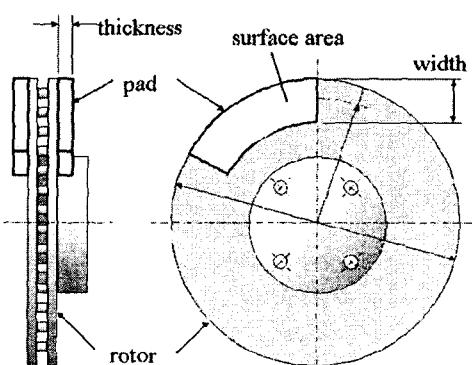


Fig. 4 Brake pad optimization

엑셀에 저장하고 있었지만 이번 문제에서는 모든 설계 데이터를 관리하지 않고 한 분야에 대한 필요한 데이터만을 관리하고 수치적 계산을 수행하게 된다.

4. 프레임워크에 따른 지표의 평가

4.1 VisualDOC

VisualDOC은 3.0 버전에서 처음으로 엑셀 인터페이스 기능을 GUI 형태로 제공하고 있다. Fig. 5와 같이 하나의 워크북으로부터 여러 워크 쉬트에 저장되어 있는 설계 데이터에 접근이 가능하다. 하지만 엑셀에 저장되어 있는 모든 데이터에 접근이 가능한 것은 아니다. 기본적으로 VisualDOC이 지정한 입력과 응답 값들만이 엑셀 인터페이스 상에 표시된다. 따라서 엑셀에 저장되어, 사용자가 엑셀로부터 VisualDOC으로 불러오고 싶은 데이터를 엑셀 인터페이스를 통하여 모두 얻어 올 수는 없다.

문제 1의 경우, 엑셀에 저장되어 있는 설계 데이터가 목적함수와 제한조건으로 사용되므로 엑셀 인터페이스에서 원하는 모든 데이터를 확인할 수 있다. VisualDOC을 이용하여 문제 1에 대해서 최적설계를 수행한 결과 Fig. 6과 같은 목적함수 값의 추이를 얻을 수 있었으며 함수값은 참고문헌⁽¹¹⁾을 통하여 확인하였다.

문제 2와 3을 VisualDOC에 적용 할 때, 엑셀 인터페이스의 문제점을 확인 할 수 있었다. 우선 제공된 인터페이스로 얻어 올 수 있는 데이터가 제한 되어 있다는 것이다. 문제 2를 해석하기 위해 ANSYS를 사용했는데 ANSYS로부터 해석된 각 노드의 변위는 x, y 축 방향의 스칼라 값으로 주어진다. 이를 엑셀에서 벡터로 계산하여 제한 조건으로 사용해야 하는데, x, y 축 방향의 변위 값이 VisualDOC에서 제한 조건으로 설정되어 있지 않기 때문에 x, y 축 방향의 변위 스칼라 값을 얻을 수 없었으며 VisualDOC에서 x, y 축 스칼라 값을 계산하여 벡터량으로 만들 수 있는 방법 역시 제공되지 않았다.

문제 1의 경우, 엑셀에 저장되어 있는 설계 데이터가 목적함수와 제한조건으로 사용되므로 엑셀 인터페이스에서 원하는 모든 데이터를 확인할 수 있다. VisualDOC을 이용하여 문제 1에 대해서 최적설계를 수행한 결과 Fig. 6과 같은 목적함수 값의 추이를 얻을 수 있었으며 함수값은 참고문헌⁽¹¹⁾을 통하여 확인하였다.

문제 2와 3을 VisualDOC에 적용 할 때, 엑셀

인터페이스의 문제점을 확인 할 수 있었다. 우선 제공된 인터페이스로 얻어 올 수 있는 데이터가 제한 되어 있다는 것이다. 문제 2를 해석하기 위해 ANSYS를 사용했는데 ANSYS로부터 해석된 각 노드의 변위는 x, y 축 방향의 스칼라 값으로 주어진다. 이를 엑셀에서 벡터로 계산하여 제한 조건으로 사용해야 하는데, x, y 축 방향의 변위 값이 VisualDOC에서 제한 조건으로 설정되어 있지 않기 때문에 x, y 축 방향의 변위 스칼라 값을 얻을 수 없었으며 VisualDOC에서 x, y 축 스칼라 값을 계산하여 벡터량으로 만들 수 있는 방법 역시 제공되지 않았다.

기본적으로 VisualDOC은 해석기 인터페이스를 위한 자체 스크립트를 이용하여 해석기 사이의 데이터를 연결하고 있다. 따라서 엑셀을 하나의 해석기로 생각했을 때 엑셀에 저장된 설계 데이터와

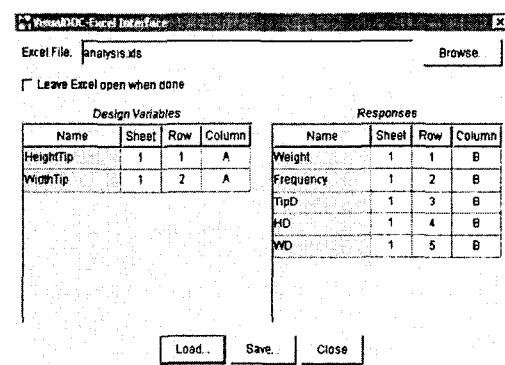


Fig. 5 VisualDOC excel interface

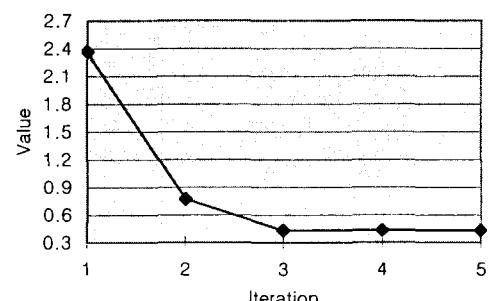


Fig. 6 Object function history of coil spring design

다른 해석기에 저장된 설계 데이터를 연결하기 위해서는 자체 스크립트를 이용해야 한다. 하지만 이를 이용할 경우 해석 프로그램 형태로 엑셀 인터페이스를 통해 만들어진 스크립트 파일이 자체 스크립트에 등록은 가능하지만, 데이터가 저장된 입, 출력 파일이 없기 때문에 입, 출력 데이터를 설정할 수 없어 다른 해석기와의 연결이 불가능하다. 또 다른 문제점은 자체 스크립트에서 제공하는 해석기 사이의 연결 형식이 하향식 한 방향으로 되어 있어 엑셀에 저장된 데이터를 이용하여 해석을 수행하고 이를 다시 엑셀에 저장하는 것이 어렵다.

이러한 문제점들로 인하여 VisualDOC에서 엑셀과 다른 해석기들을 연결하기 위해 파이썬(Python) 스크립트를 이용하여 엑셀 인터페이스를 개발, 사용하였으며, 문제 2와 3에 대한 결과가 Fig. 7, 8과 같이 구해졌다.

VisualDOC에서 지원하는 후처리 기능을 사용하지 않고, 엑셀을 통하여 후처리를 자동화 하기 위해서는 많은 어려움이 예상된다. 우선 VisualDOC이 C/C++을 이용한 외부프로그램 인터페이스를 제공하지만 이를 이용하여 엑셀로 데이터를 읽기기 위해서는 결국 저수준의 컴퓨터 프로그래밍 작업이 필요하다.

VisualDOC은 3.0 버전부터 엑셀 인터페이스 사용에 대한 내용을 VisualDOC HOWTO 매뉴얼에 기술하고 있다. 그 내용은 문제 1 정도를 해결할 수 있는 수준이다. 따라서 많은 사용자가 원하는 문제 구성 형태인, 문제 2와 3을 해결 할 수 있는 도움말 문서가 더 필요하다.

4.2 ModelCenter

모델센터에서 지원하는 엑셀 인터페이스 형태는 스크립트 형태이다. 따라서 사용자가 Fig. 9와 같은 형태의 스크립트 파일을 만들어 해석 서버에 등록하면 모델센터에서 이를 컴포넌트로 인식을 한다. 또한 입력 값과 출력 값 중 원하는 값을 목적 함수, 설계 변수, 제한 조건으로 지정할 수 있으며 다른 해석기의 입, 출력값과 연결기능을 이용하여 데이터를 연결하므로 엑셀에 저장된 설계 데이터와 다른 해석기의 설계 데이터 이동이 비교적 자유롭다.

우선 문제 1에 대하여 스크립트 형태의 엑셀 인터페이스를 만들고, 이를 해석 서버를 통하여 모델센터에 컴포넌트로 등록된다. 이 등록된 컴포넌트와 모델센터에서 제공하는 최적설계 컴포넌트를 연결하면 손쉽게 문제구성이 완성된다. Fig. 10은 그 결과이다.

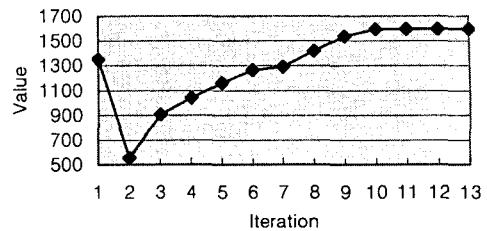


Fig. 7 Object function history of ten bar truss design

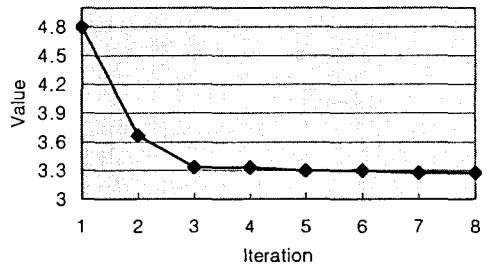


Fig. 8 Object function history of brake pad design

```
file = spring.xls
cell: B11 weight output
cell: B12 d      input
cell: B13 D      input
cell: B14 costrnt output
```

Fig. 9 ModelCenter excel interface script

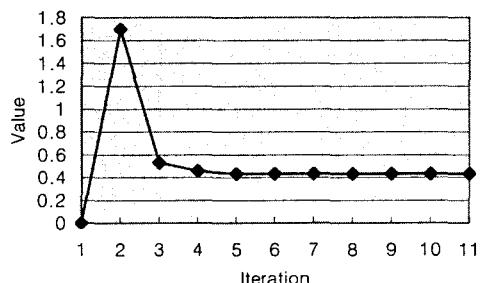


Fig. 10 Object function history of coil spring design using ModelCenter

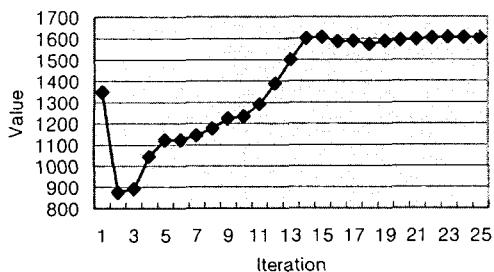


Fig. 11 Object function history of ten bar design using ModelCenter

입력 데이터는 엑셀에 저장되어 있고 ANSYS에서 계산된 출력 값 역시 엑셀에 저장된다. 따라서 엑셀이 설계에 필요한 모든 데이터를 관리하고 있는 형태이다. 각 컴포넌트 사이의 설계 데이터 연결은 연결 기능을 이용하였으며, 그 결과는 Fig. 11과 같다.

문제 3은 비용을 최소화하는 브레이크 패드 최적설계 문제인데 원래 모델센터에서 제공하는 예제이기 때문에 어려움 없이 문제를 구성할 수 있다. 구성된 문제는 Fig. 12와 같으며 그 결과는 Fig. 13에 나타내었다.

모델센터에서 제공하는 매뉴얼에는 엑셀 데이터 중 입력과 출력 값을 지정하여 엑셀 인터페이스를 만드는 수준의 내용이 포함되어 있다. 이를 바탕으로 만들어진 엑셀 컴포넌트는 모델센터에서 만들 수 있는 해석 컴포넌트와 같은, 동일한 개념의 컴포넌트로 간주되므로 다른 해석기와 엑셀을 연결시키는데 모델센터에서 지원하는 자원만을 이용하여 문제 구성과 해결이 가능하다.

4.3 iSIGHT

iSIGHT는 GUI 형태의 인터페이스에 엑셀 파일 및 설계 데이터의 위치를 입력하면 스크립트 파일을 자동 생성해 준다. 특히 iSIGHT는 자체 프레임워크에서 제공하는 엑셀 인터페이스뿐 아니라 엑셀에 내장하여 사용 할 수 있는 GUI를 제공하여 사용자가 iSIGHT에서 뿐만 아니라 엑셀 자체에서도 엑셀 인터페이스 생성을 위한 작업을 할 수 있다.

이렇게 만들어진 스크립트 파일은 모델센터와 마찬가지로 하나의 컴포넌트로 인식되어 iSIGHT에서 사용된다. 이 컴포넌트의 스크립트 파일에 입, 출력 데이터에 대한 정보가 저장되어 있다. 이 저장된 정보를 바탕으로 다른 컴포넌트의 데이터와 이름으로 연결되어 문제를 구성하게 된다.

문제 1을 해결하기 위해 iSIGHT에서 제공하는 엑셀 컴포넌트 기능을 사용하였으며 이를 통하여 입출력 값을 설정하게 된다. iSIGHT는 이를 입, 출력 값 중 필요한 값을 목적함수, 제한조건, 설계변수로 정의할 수 있으며 사용자가 원하는 최적설계 모듈을 선택하여 최적설계를 수행한다. 수행된 결과는 Fig. 14와 같다.

문제 2에서는 iSIGHT의 SimCode 기능을 이용하여 ANSYS 입, 출력 데이터를 정의했으며, 엑셀 인터페이스를 이용하여 엑셀에 저장된 데이터에 대한 입, 출력 데이터를 정의하였다. 이 두 컴포넌트의 데이터는 이름으로 연결된다. 이름연결은 여러 개의

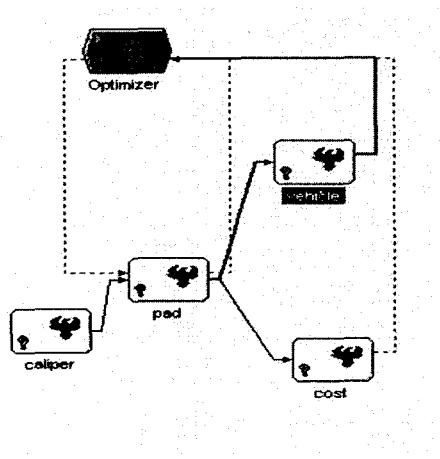


Fig. 12 Construction of brake pad design using ModelCenter

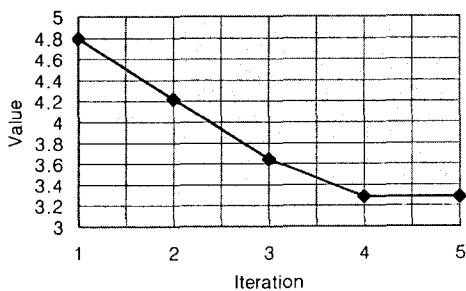
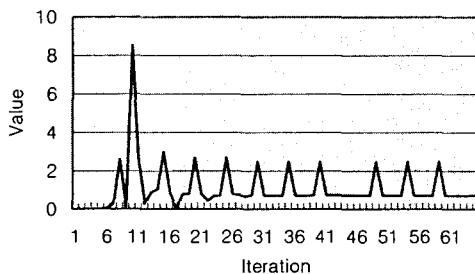
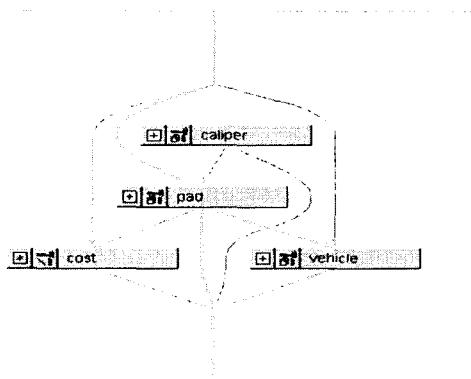
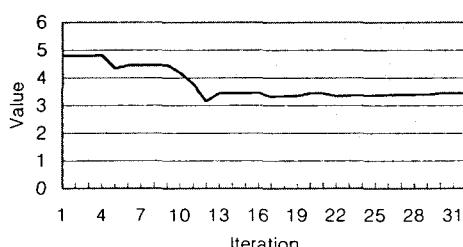


Fig. 13 Object function history of brake pad design using ModelCetner

문제 2의 각 부재의 응력과 각 노드의 변위를 계산하기 위해 ANSYS를 사용했다. ANSYS에 필요한

Table 1 Summary of benchmark results

	VisualDoc	ModelCenter	iSIGHT
사용 편의성	보통	보통	우수
기능성	미흡	보통	우수
후처리	미흡	보통	보통
API	미흡	보통	보통
사용지원능력	미흡	우수	우수

**Fig. 14** Object function history of coil spring design using iSIGHT**Fig. 15** Construction of brake pad design using iSIGHT**Fig. 16** Object function history of brake pad design using iSIGHT

컴포넌트를 연결할 수 있는 방법을 제공하지만 모델센터처럼 직관적인 형태의 그래픽 연결 기능이 아니기 때문에 사용자가 데이터의 흐름을 정확히 파악하는데 어려움이 따른다.

문제 3 역시 엑셀 컴포넌트와 각각의 컴포넌트를 이름연결을 이용하여 구성했다. 문제 구성은 Fig. 15와 같으며 그 결과는 Fig. 16이다.

iSIGHT의 결과를 엑셀에 반영하기 위해서는 외부프로그램 인터페이스를 이용해야 하는데 이는 Tcl(Tool Command Language)을 이용한 저수준의 컴퓨터 프로그래밍을 해야 함을 의미한다.

iSIGHT는 사용자 매뉴얼에서 엑셀 인터페이스에 대해서 문제 1을 해결할 수 있는 예제를 소개하고 있다. 이 내용을 바탕으로 만들어진 엑셀 컴포넌트는 iSIGHT에서 만들어진 다른 컴포넌트와 동일하게 정의되므로 사용자 매뉴얼에 있는 내용으로 다른 컴포넌트를 연결하여 복잡한 문제를 구성하는 것이 가능하다.

5. 결 론

본 논문에서는 세 개의 설계문제를 통하여 상용 최적설계 프레임워크의 엑셀 인터페이스를 비교 평가했다. 비교 평가 결과, 각 프레임워크의 특징에 따라서 장, 단점을 확인할 수 있었으며 이를 요약하면 Table 1과 같다.

VisualDOC은 GUI 형태의 인터페이스를 제공한다. 이 기능은 문제 설정 방법의 한계로 인하여 사용자가 원하는, 엑셀을 적용한 복잡한 형태의 문제 구성에 한계를 가지고 있다. 문제점을 정리하면, 1) 엑셀 인터페이스로 얻어 올 수 있는 데이터의 제한, 2) 엑셀 인터페이스와 자체 스크립트로 정의된 해석기 사이의 연계의 어려움, 3) 하향식 데이터 흐름 등이다. 이러한 문제점으로 인해 VisualDOC에서 제공하는 엑셀 인터페이스를 이용하여 해결 할 수 있는 설계문제는 매우 제한적이고, 단순한 형태의 문제만이 가능하다. VisualDOC의 자체적인 문제가 엑셀 인터페이스를 제한하는 면이 많기 때문에 문제를 해결하기 위해서는 근본적으로 VisualDOC이 컴포넌트 기반의 데이터 입, 출력 형식으로 바뀌어야 하며 현재 상황에서 엑셀 인터페이스의 활용도를 높이기 위해서는 자체 스크립트안에 엑셀 인터페이스의 설정이 가능하도록 개선되어야 한다.

ModelCenter는 텍스트 기반 엑셀 인터페이스를 제공하며, 이는 숙련된 사용자가 엑셀 인터페이스를 사용하는데 좋은 환경이다. 하지만 모델센터나 엑셀

인터페이스를 처음 사용하는 사람에게는 텍스트 형태의 작성 방법을 익혀야 하므로, 초보자가 사용하기에는 부담스럽다. 따라서 텍스트 기반의 엑셀 인터페이스뿐만 아니라 이를 자동 생성해 주는 GUI도 함께 제공되어야 한다.

iSIGHT는 GUI 형태의 엑셀 인터페이스를 제공한다. 이는 두 가지 형태로 제공되는데, 사용자가 iSIGHT내에서 사용할 수 있는 형태와 엑셀에 내장된 인터페이스를 실행시켜 사용할 수 있는 형태로 되어 있다. 이러한 엑셀 인터페이스는 초보자가 사용하기에 적합한 형태로써 쉽게 엑셀 컴포넌트를 만들 수 있다. 하지만 데이터의 양이 많을 경우 여러 번의 반복 작업을 GUI를 통하여 수행해야 하므로 작업 속도가 너무 느려지는 단점이 있다. 따라서 모델센터와 같은 텍스트 기반 엑셀 인터페이스의 지원이 병행 지원되어야 한다.

그러므로 설계 프레임워크에서 엑셀을 효과적으로 활용하기 위해서는 프레임워크이 컴포넌트 기반의 문제 구성 형태를 지원해야 하며, 엑셀 인터페이스의 지원 방식을 텍스트와 GUI, 두 형태를 모두 지원함으로써 초보자뿐 아니라 숙련자도 효율적으로 사용할 수 있도록 해야 한다. 또한 지금과 같이 단순히 엑셀 인터페이스만을 사용할 수 있는 형태의 매뉴얼이 아닌 복잡한 설계 문제를 초보자도 쉽게 따라 할 수 있는 형태의 매뉴얼 혹은 자습서가 지원되어야 한다. 더불어 단순히 설계 데이터의 입력과 출력 값을 일시적으로 저장하고 결과 값만을 최종적으로 저장하는 형태가 아닌, 설계 과정에서 생성된 모든 중간 데이터를 저장할 수 있는 기능을 프레임워크이 제공한다면 엑셀을 통하여 더욱 효율적인 데이터 관리가 가능한 설계 프레임워크이 될 것이다.

후기

이 논문은 2004년도 서울시립대학교 학술연구 조성비의 지원에 의하여 연구되었음.

참고문헌

- (1) Park, H.-W., Lee, S.J., Lee, H.-S. and Choi D-H., 2004, "Adaptive Parallel Decomposition for Multidisciplinary Design," *KSME International Journal*, Vol. 18, No. 5, pp. 814~819.
- (2) Hong,E.J.,Lee,S.J.,Lee,J.H. and Kim,S.M., 2003, "Data Server Oriented Computing Infrastructure for Process Integration and Multidisciplinary Design Optimization," *Trans. of the Society of CAD/CAM Engineers*, Vol. 8, No. 4, pp. 231~242.
- (3) Engineous Software, Inc., 2003, *iSIGHT User's Guide: version 7.0*, Engineous Software, Inc.
- (4) Phoenix Integration Inc., 2002, *ModelCenter Traning Basics Part A : Version 4.0*, Phoenix Integration Inc., 1750 Kraft Drive, Suite 2200, Blacksburg, VA.
- (5) VR & D Inc., 2002, *VisualDOC Manual : version 3.0*, VR & D Inc.
- (6) Lee, S. J., 2003, "Benchmark Study of Design Frameworks for Multidisciplinary Design Optimization", *Journal of Institute of Information and Technology*, Univ. of Seoul, Vol. 5, pp. 22~27.
- (7) Hwang, W.J., 2001, *Benchmark of Commercial Design Frameworks For Multidisciplinary Design Optimization Problem Formulation*, MS Thesis, Hanyang University, Dept. of Mechanical Engineering.
- (8) Hong, E.P., Park, C.M. and Park, G.J., 2001, "Study on the Comparison of design software performance in view of optimal design," *Proceedings of the KSME Spring Meeting*, pp. 176~181.
- (9) Cho, S.O., Lee, J.W and Byun, Y.H. 2002, "Study on the Integration of Analysis Codes and Optimization Procedures to develop MDO Framework," *Trans. of the Korean Society of Aeronautical and Space Sciences*, Vol. 30, No. 5, pp. 1~10.
- (10) Salas, A.O. and Townsend, J. C., 1998, "Framework Requirements For MDO Application Development", *AIAA-98-4740*, NASA Langley Research Center, pp. 1~9.
- (11) Haug, E.J. and Arora, J.S., 1979, *Applied Optimal Design*, Wiley & Sons, New York, USA, pp. 70~72
- (12) Haug, E.J. and Arora, J.S., 1979, *Applied Optimal Design*, Wiley & Sons, New York, pp. 171~173.