

## 다분야통합최적설계를 지원하는 분산환경 기반의 설계 프레임워크 개발

주민식\*, 이세정\*\*, 최동훈\*\*\*

### Development of a Distributed Computing Framework for Implementing Multidisciplinary Design Optimization

Chu, M. S.\*, Lee, S. J.\*\* and Choi, D.-H.\*\*\*

#### ABSTRACT

A design framework to employ the multidisciplinary design optimization technologies on a computer system has been developed and is named as the Extensible Multidisciplinary Design Integration and Optimization System (EMDIOS). The framework can not only effectively solve complex system design problems but also conveniently handle MDO problems. Since the EMDIOS exploits both state-of-the-art of computing capabilities and sophisticated optimization techniques, it can overcome many scalability and complexity problems. It can make users who are not even familiar with the optimization technology use EMDIOS easily to solve their design problems. The client of EMDIOS provides a front end for engineers to communicate the EMDIOS engine and the server controls and manages various resources such as scheduler, analysis codes, and user interfaces. EMDIOS client supports data monitoring, design problem definition, request for analyses and other user tasks. Three main components of the EMDIOS are the Engineering Design Object Model which is a basic idea to construct EMDIOS, EMDIOS Language (EMDIO-L) which is a script language representing design problems, and visual modeling tools which can help engineers define design problems using graphical user interface. Several example problems are solved and EMDIOS has shown various capabilities such as ease of use, process integration, and optimization monitoring.

**Key words :** MDO, Framework, EDOM, EMDIOS, 설계 프레임워크, EMDIO-L

#### 1. 서 론

복잡한 설계문제나 최적화 문제와 같은 공학 설계 문제를 해결하는데 설계 프레임워크라는 소프트웨어가 산업체에서 많이 적용되고 있다. 이러한 설계 프레임워크의 사용 목적은 최적 설계 문제에 대해 엔지니어들이 해당분야의 지식 결핍으로 인한 시행착오를 줄이며, 입증된 솔루션의 제공 및 해석 프로그램과 최적화 모듈의 연결, 최적화 문제의 설정 등에 소요되는 비용을 줄이고자 함이며<sup>1)</sup>, 일반적인 설계문제에 있어서 매우 효과적인 시스템으로 사용되어 왔다. 하지만

기존 설계 프레임워크들의 문제점들 중 가장 큰 두 가지는 시스템 개발 시 설계환경과 다분야통합최적화(Multidisciplinary Design Optimization, MDO) 방법론을 고려하지 않고 설계프레임워크가 개발 되었다는 점이다.

대표적으로 ModelCenter<sup>2)</sup>는 해석 리소스의 통합화에 중점을 두어 개발되었으며, iSIGHT<sup>3)</sup>는 다양한 최적화 기술과 해석 프로그램의 통합을 위한 강력한 파싱 스크립트 기술을 가지고 있으며, 기타 프레임워크들은 웹 환경의 지원, CAD 시스템과의 접속을 특징으로 하고 있다<sup>4)</sup>.

하지만 이러한 설계 프레임워크들은 일반적인 구조가 아닌 복잡한 구조의 설계문제 및 다분야통합최적화문제의 경우 이를 표현하고 설정할 수 있는 아키텍처를 가지고 있지 않기 때문에 이를 표현 또는 설정하기에는 어려운 실정이었다. 이러한 문제점 때문에 다

\*한양대학교 기계설계 박사과정

\*\*중신회원, 서울시립대학교

\*\*\*한양대학교 기계설계학과

- 논문투고일: 2004. 05. 12

- 심사완료일: 2004. 11. 04

분야통합최적화의 특징인 해석 프로세스의 병렬화 및 분산환경지원<sup>1)</sup>, 방법론들 간의 상호 전환, 시스템 분해 기법의 적용은 여전히 문제점으로 남아 있다.

이를 위하여 본 연구에서는 EMDIOS(Extensible Multidisciplinary Design Integration and Optimization System)라는 소프트웨어 시스템을 개발하였다. EMDIOS는 EDOM(Engineering Design Object Model) 아키텍처와 이 아키텍처를 언어로 표현한 EMDIO-L(EMDIO-Language), GUI(Graphic User Interface) 기술인 비주얼 모델링 기술을 기반으로 개발되었다. 본 논문에서는 기존의 설계 프레임워크들 중 대표적인 설계 시스템인 iSIGHT, ModelCenter와 비교 분석하였다.

일반적인 최적화 적용 사례로는 실제 산업체 문제에 적용한 결과를 제시하였으며, 다분야 통합최적화 문제 적용 사례로는 Heart Dipole(NASA) 문제를 적용하였으며 다분야통합최적화 방법론 중 MDF(Multiple Discipline Feasible, IDF(Individual Discipline Feasible), CO(Collaborative Optimization)<sup>6)</sup> 방법론을 세가지 시스템에 적용하여 최적화를 수행하였다.

본 연구에서는 최적 설계 성능은 비교의 대상이 아니므로 비교, 분석은 제외하였으며, 동일 예제를 다분야통합최적화 문제로의 설정 가능성과 이에 따른 프로세스의 병렬화, 확장성 및 다분야통합최적화 방법론 간의 전이 가능성을 비교, 분석하였다.

### 2. EMDIOS

Fig. 1은 다분야통합최적설계 기법 중심으로 개발된 EMDIOS로 설계문제가 설정된 화면이다. 이 시스템은 세 부분의 구성 요소로 분리하여 개발되었다.

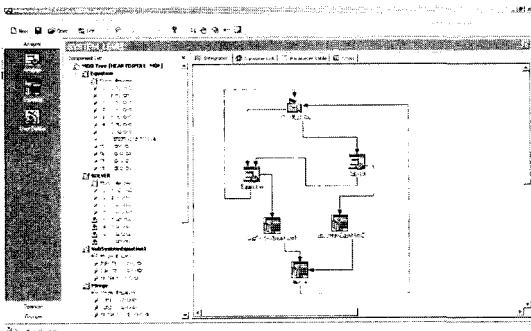


Fig. 1. EMDIOS.

첫 번째로는 EDOM으로써 다분야통합최적화 방법론을 패턴화한 컴포넌트 디자인 구조이다. EDOM은 다분야통합최적화 기법을 분석하여 나온 최하위 컴포넌트로부터 최상위 컴포넌트 집합 모델을 정의 하는 EMDIOS의 기반구조이다.

두 번째는 EMDIO-L로써 EDOM 기반으로 개발된 스크립트 형태의 언어이다. 다른 상용 프레임워크와 다른 점은 상용 언어를 기반으로 하여 개발된 것이 아니라 EDOM 구조를 기반으로 개발된 최적화 전용 스크립트 언어이다. 세 번째 비주얼 모델링은 기존 설계 프레임워크들이 가지는 그래픽 유저 인터페이스에 덧붙여 EMDIOS에서 문제를 좀더 직관적으로 표현한 편리한 그래픽 유저 인터페이스를 의미한다

### 2.1 EDOM (Engineering Design Object Model)

기존 프레임워크인 iSIGHT, ModelCenter의 기반 구조는 일반적인 최적화 및 시스템 통합이라는 관점에서 개발되었다. ModelCenter는 소프트웨어 아키텍처인 COM<sup>7)</sup> 기반의 컴포넌트 개념과 이를 조합한 스크립트 아키텍처로 기반구조를 가져 시스템 확장성이라는 측면에서 상대적으로 효율적이다. iSIGHT 역시 내부적으로 엔지니어링 기반 태스크 및 최적화 모듈 간의 유연한 아키텍처 가지고 있어 엔지니어 설계에 적합한 형태를 취하고 있다.

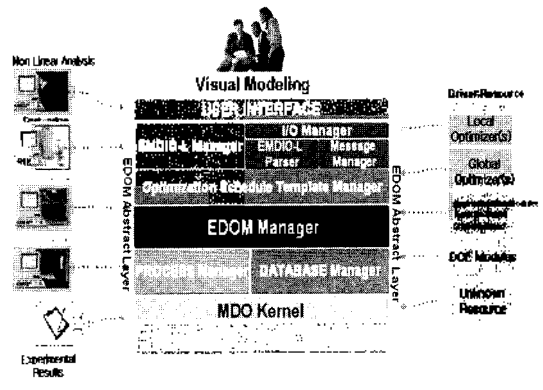


Fig. 2. EDOM Architecture.

하지만 다분야통합최적설계는 단일 최적화에 비해서 대상 문제가 매우 복잡 다단하며 이러한 복잡한 구조의 설계문제를 해결하기 위해서는 자동분해 기법의 지원 및 다중레벨로 표현할 수 있는 설계문제의 구성법, 설계관련 리소스를 제어할 수 논리 구조 등이 추가적으로 필요하게 된다.

Table 1. Object definitions EDOM

| No     | 단일/다분야                                | 최적화 방법 형태   | 객체 결합 형태   | 공학 리소스 예                     |
|--------|---------------------------------------|---|--|------------------------------|
| 1      | Single Optimization                   | Single Driver Object Model  | Driver Object +Common Object Model                       |                              |
| 2      | Single Optimization                   | OPTIMIZATION  | Optimization Object + Analysis Object or Equation Object |                              |
| 3      | Multidisciplinary Design Optimization | Multi Driver Object Model   | Single Driver Object (s) + Common Object(s)              |                              |
| 4      | Multidisciplinary Design Optimization | MDF   | Optimization Object(1) + Common Object(s)                |                              |
| 5      | Multidisciplinary Design Optimization | IDF   | Optimization Object(1) + Common Object(s)                |                              |
| 6      | Multidisciplinary Design Optimization | CO  | Optimization Object + Optimization Object Model(s)       |                              |
| 7      | Multidisciplinary Design Optimization | CSSO(concurrent subspace system optimization)                               | Optimization Object + Approximation Object Model(s)      |                              |
| 객체 설명  |                                       |   |  |                              |
| Object | Optimization Object                   | Header + Input parameter + Optimization Code+ output parameter              | EMDIOS Defined   | ADS, NLPQL, DOT...           |
|        | Approximation Object                  | Header + Input parameter + Approximation Code+ output parameter             | EMDIOS Defined   | Neural Network Code, RSM ... |
|        | Unknown Object                        | Unformatted Form  | ?  | (Control Object)             |
|        | Equation Object                       | Header + Input parameter + equation + output parameter                      | Engineering Equation                                     | Methmetical Equation         |
|        | Analysis Object                       | header + Input parameter(files) + application run + output parameter(files) | Engineering Application                                  | ADAMS, ANSYS, NASTRAN...     |

이러한 요구에 의하여 고안된 패턴 컴포넌트 구조인 EDOM은 다분야통합최적화 문제 및 복잡한 설계 문제를 EDOM에서 정의된 최하위 컴포넌트 객체들(설계 리소스들)로 조합하여 최상위 컴포넌트 객체들(다분야 통합 설계문제) 구성하는 객체 조합(Object Composition) 형태인 것이다.

설계 리소스의 객체화는 설계 문제의 변경 및 신규 작성을 용이하게 해주며, 외부 해석 관련 리소스(ANSYS, NASTRAN, ADAMS 등 상용 해석기)들의 입출력 구조를 랩핑하여 인터페이스가 통일된 재사용할 수 있는 마이너리 컴포넌트 구조로 만들어준다.

EMDIOS는 EDOM패턴을 이용한 재사용성을 전제로 일관된 인터페이스 구조를 가지기 때문에 재사용성에 있어서 매우 효과적인 구조로 가지게 된다. Fig. 2는 EDOM으로 구현된 EMDIOS의 기본 구조를 나타내고 있다. EDOM의 설계 기본은 세 가지이다.

첫 번째는 객체 모듈화로서 내부에 구현하는 세부 사항을 캡슐화하기 위해 정의된 시스템 인터페이스를 제공하는 것이다. 객체 모듈화는 설계의 변경 시 재설계라는 충격을 최소화 할 수 있다. 두 번째는 확장성

이며, 여기서 확장성이란 프레임워크의 확장성을 의미하는 것이 아니라 현존하는 다분야통합최적화 방법론 이외에 새로운 통합최적화 방법론이 나오더라도 기존의 하위 컴포넌트를 조합하여 만들 수 있는 기능을 의미한다. 세 번째는 컨트롤의 전환으로 프레임워크는 사용자 모듈과 프레임워크 모듈이 결합됨으로써 서로 유기적으로 동작하는 시스템이 된다. 어떤 동작을 요할 경우 일반적으로 프레임워크 모듈에서 사용자 모듈로, 사용자 모듈에서 프레임워크 모듈로 컨트롤의 포커스가 전이될 수 있으며 이러한 구조는 설계자로 하여금 다양한 기능을 설계 프레임워크에 추가할 수 있게 한다.

이런 개념의 의거하여 다분야통합최적화 방법론을 분석하여 이를 패턴에 의거하여 분류하고, 다시 객체 조합 할 수 있는 모델과 컴포넌트로 최하위부터 최상위까지 분리하고 정리하여 Table 1에 요약하였다. 테이블에서 보이듯이 EDOM은 최적화 또는 다분야통합최적설계 구조를 일반객체모델과 구동객체모델이라는 두 가지 객체모델의 조합으로 두명 하고 있다.

일반객체모델(Common Component Model)은 자체

적으로 공학 해를 제공할 수 있는 설계리소스 집합 형태를 의미한다. 즉, 설계문제의 수식이나 해석을 나타내는 코드형태이며, 일반객체모델에 속하는 수식객체는 최하위 객체가 되는 MDO 컴포넌트로서 해더와 입력 파라미터, 수치 식, 출력 파라미터로 구성되어 있다.

어플리케이션객체는 수식객체를 상속 받아 구성된 일반객체로서 단일 해석분야나 여러 개의 해석이 결합된 통합해석 분야를 나타내는 객체이다.

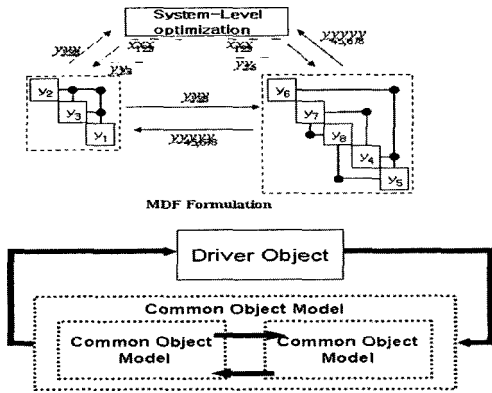


Fig. 3. MDF Configuration with EDOM.

구동객체모델(Driver Object Model)은 단일 형태로는 동작할 수가 없는 형태로 기초적인 최적화 문제나 다분야통합최적화 문제들은 이러한 구동객체모델의 조합으로 표현이 된다.

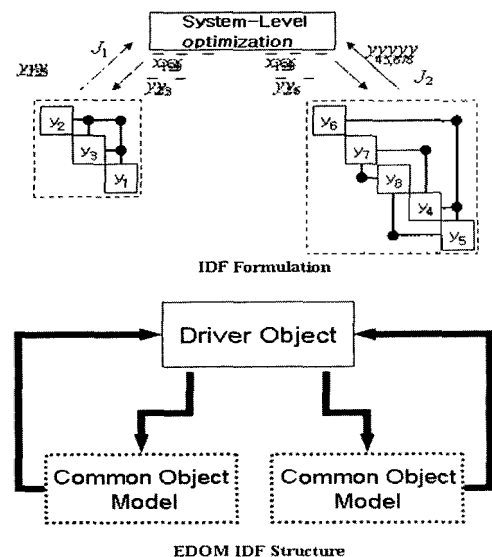


Fig. 4. IDF Configuration with EDOM.

구동객체를 이용하여 모델을 정의 한다는 것은 다른 일반객체나 다른 구동객체와의 조합을 의미하며 최적화 객체, 근사화 객체가 이러한 구동객체 모델에 속하는 객체이다. 기초적인 최적설계 문제는 구동객체모델 중 하나의 최적화객체와 일반객체의 조합으로 표현이 가능하다. 이들 객체를 이용하면 다분야통합 최적설계 기법인 MDF, IDF, CO방법론들도 객체 조합으로 표현가능하며, 다음과 같은 구조를 가진다.

Fig. 3의 MDF는 일반적인 최적화 방법론을 표현하는 구조와 동일하며 구동객체 + 일반객체의 구조로 표현이 된다. Fig. 4의 IDF는 MDF와 비슷한 형태이지만 하위해석영역이 단일객체가 아니고 서로 독립된 복수의 일반객체로 구성되는 차이점이 있다. 이는 하위 영역의 객체들은 병렬처리할 수 있음을 의미한다. Fig. 5에서 보이는 CO 구조는 MDF나 IDF보다 복잡한 구조를 가지며, EDOM으로 표현했을 때 구동객체 + (구동객체 + 일반객체)의 구조를 가지게 된다. 여기서도 하위 영역은 병렬화되어 수행될 수 있다.

다분야통합최적화 문제를 EDOM으로 표현했을 때 해석 영역과 최적화 영역간의 명확한 분리가 가능하며, 병렬처리를 위해서는 위상 관계가 동등한 배열 구

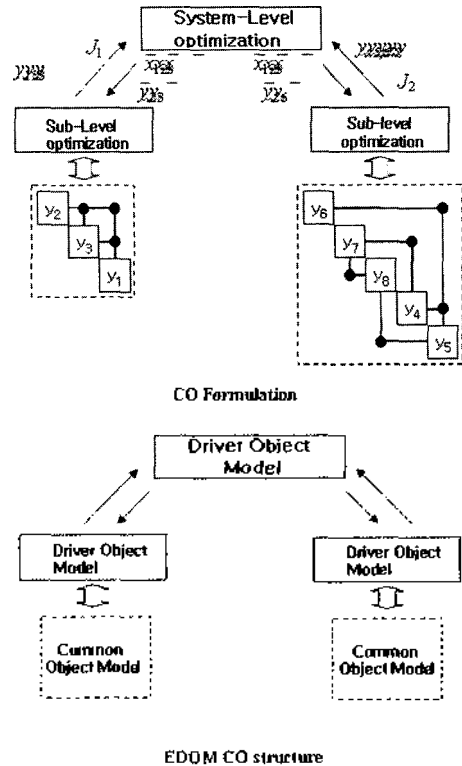


Fig. 5. CO Configuration with EDOM.

조를 찾아 병렬 수행하면 된다. 각각의 방법론들 간의 전환을 위해서는 객체 조합된 모델을 기준으로 MDF에서 IDF, MDF에서 CO로 변환될 수 있다.

**2.2 분해 기법(Decomposition)**

일반적 설계 프레임워크인 iSIGHT나 ModelCenter는 사용자가 정의한 문제를 실행시키기 위하여 자체적인 스케줄러를 내장하고 있다. 일반적인 스케줄링의 형태는 설계구조행렬(Design Structure Matrix, DSM)을 구성한 후 순차적으로 프로세스를 실행 하는 것이다. 이러한 방식에서는 자체적인 규칙에 따라서 수렴의 여부를 판단한 후 종결하는 것이다. 이는 단일 최적화 모듈과의 조합의 경우에 있어서 순서순 해결 방법이며 기존 설계 프레임워크들도 이와 유사한 방식을 가지고 있다.

다분야통합최적화 방법론들은 다층 최적화 구조이며 기존과는 다른 최적화 코드와 해석 코드가 혼재되어 있는 복잡한 형태이므로 분해 기법을 적용하는데 문제점이 발생한다. 또한 외부 분해 기법의 적용의 용이성 및 복잡하게 설정된 설계문제에 분해 기법을 적용하여 형성한 여러 개의 해석그룹에 대한 병렬화 자동 지원은 기존의 프레임워크들로는 적용이 불가능하였다. 기존 시스템들은 분해 기법이 적용될 수 있는 독립 컴포넌트 단위와 이러한 독립 단위를 이용한 상위객체 조합 개념이 없는 스케줄링의 방식이기 때문이다.

EDOM으로 구성된 모델링 정보는 객체 조합 구조로 정의되어 있으며, 모델 마나 일반적인 분해 기법이 적용 되는 DSM으로 일대일 변환이 된다. 또한 Fig.

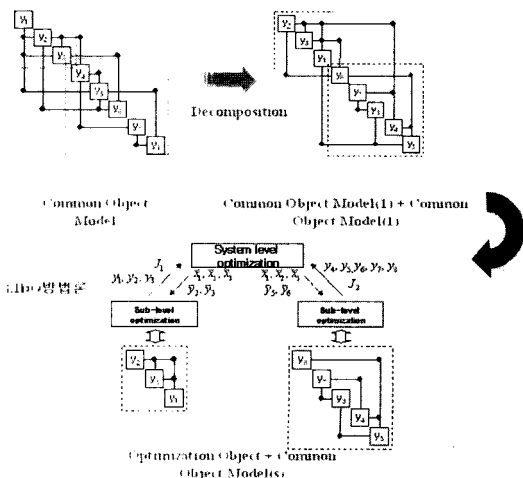


Fig. 6. Decomposition with EDOM.

6과 같은 MDO문제의 변환 시 EDOM은 매우 편리한 구조를 제공할 수 있다. 일반적인 설계문제에서 분해 기법을 이용하여 그룹화한 후, 다분야통합최적화 문제로의 변환에 이르기까지 변환과정은 해석관련 리소스와 기타 리소스들간의 객체 조합으로 표현할 수가 있다. Fig. 6과 같이 COM에서 COM(1)+COM(1)의 구조로 변경이 된 후 다중 구동객체 모델(IDF or CO)로 표현이 자연스럽게 전환된다.

**2.3 EMDIO-L**

일반적으로 설계 프레임워크로 해석 리소스를 통합하는 과정에서 설계자는 응용 프로그램 제어할 할 수

```

MDO MODELING LANGUAGE DESCRIPTION FILE
PROJECT_NAME = HEARTDIPOLE_IDF
MDOMETHOD = IDF
START_POINT = SYSTEMLEVEL

IMPLEMENT_DOMINFO
{
    DOM_NAME      .SUBSYSTEM2
    MODEL         .ANALYSIS_MODEL

    INPUT
    SYSPARA.0,x2,1.000301
    SYSPARA.1,x3,-0.000063

    OUTPUT
    DEQU.0,LINK44,0.000000
    DEQU.1,LINK66,0.000000
}

IMPLEMENT_DOMINFO
{
    DOM_NAME      .SYSTEMLEVEL
    MODEL         .OPTIMIZATION_MODEL

    INPUT
    SYSPARA.0,h4,0.000198
    SYSPARA.1,h3,-0.000266
    SYSPARA.2,h2,-0.000328

    OUTPUT
    DESVAR.0,x1,1.010000
    DESVAR.1,x2,1.010000
    DESVAR.2,x3,1.010000
}

BEGIN_EXCHANGE_MAP()
DATAEXCHANGE(SUBSYSTEM1.LINK11,Equation.LINK11)
DATAEXCHANGE(SUBSYSTEM1.LINK88,Equation.LINK88)
DATAEXCHANGE(SYSTEMLEVEL.x2,SUBSYSTEM1.x2)
DATAEXCHANGE(SYSTEMLEVEL.x3,SUBSYSTEM1.x3)
DATAEXCHANGE(Equation.f8,SYSTEMLEVEL.f8)
END_EXCHANGE_MAP()
    
```

Fig. 7. An example of EMDIO-L.

있는 내부 구조에 대한 이해가 요구되었다. 기존 설계 프레임워크는 이런 내부 구조의 이해를 돕는 방법으로 사용자 기반의 스크립트 언어를 지원하고 있다.

본 논문에서 제안하는 EMDIO-L은 ModelCenter나 iSIGHT에서 사용되는 스크립트 언어와 같이 설계문제를 정의 한다는 관점에서는 동일한 의미를 가진다. 그러나 차이점은 개발 초기부터 다분야통합최적화를 고려한 스크립트형 언어라는 점이다. Fig. 7은 EMDIO-L을 이용하여 IDF문제로 설정한 예의 일부를 보여주고 있다.

ModelCenter나 iSIGHT는 같은 문제를 설정하기 위해서는 비주얼 베이직 및 기타 상용 스크립트를 제공하는데, 이는 일반 사용자가 다분야통합최적화 문제를 표현하는 데 일정 수준의 스크립트 언어에 대한 지식이 필요하다는 것을 의미한다. 일반적인 상용언어를 사용한다는 것은 강력한 언어 기반으로 설계자가 제어를 할 수 있다는 장점이 있지만, 최적화 문제를 정의하는 전용언어가 아니므로 설계자는 높은 수준의 상용언어 제어 능력이 요구된다.

또 다른 문제점은 이들 상용 스크립트의 복적이 전산과학 측면에서 개발된 것이라는 점이다. 따라서 설계문제나 다분야통합최적화 문제의 해석 프로세스의 병렬화, 분산화, 최적화, 분해 기능, 근사화 기법 등 공학적인 측면의 기능을 표현하는 것이 매우 까다로우며, 지원을 못하는 일이 발생하기 때문이다.

위의 Fig. 7의 예제에서는 주어진 문제를 IDF 방법을 이용하여 다분야통합최적화 방법론으로 표현하였다. 스크립트가 최적화 전용언어로 구성되었기에 최적화 문제의 표현 이외에 불필요한 부분이 없으므로 표현이 간결하게 된다.

다분야통합최적화 방법론을 표현하기에 적합한 구조 외에도 설계자들이 모델링 언어를 배우는데 드는 시간과 비용이 적어야 설계비용도 줄어 드는 것이다. 일반적인 상용언어는 그 나름대로의 언어 표현 형식을 가지고 있지만 EMDIO-L의 문법은 NASTRAN 입/출력 방식을 따르고 있다. 이는 설계자들이 의미를 파악하기 쉽다는 점과, 설계 문제의 설정에 있어서 기존의 스크립트언어에 비해서 매우 간결하면서 직관적이라는 장점을 제공한다.

기존의 상용언어에 익숙한 사용자들은 표현의 다양함이라는 매우 강력한 도구를 가지지만, 실질적으로 이러한 상용 언어들은 공학설계를 위한 것이 아니므로 최적화 문제의 설정에 있어서 많은 문제점을 가지고 있었으며, 다분야통합최적화문제의 표현에 있어서는 더욱 어색한 구조를 가지는 한계성을 지니고 있

다. EMDIOS-L은 설계문제의 직관적인 표현 외에 다분야통합최적설계를 모델링 하는데 있어서 매우 적합한 언어로 개발되었다.

### 3. 적용 예제

개발된 프레임워크의 유효성을 보이기 위해서 여러 가지 검증된 최적화 문제를 수행하였으며, 두 가지 문제를 선택하여 EMDIOS와 상용 프레임워크로 수행한 결과를 본 논문에 제시한다.

#### 3.1 Switched Reluctance Motor<sup>(8)</sup>

SRM 모터는 기존의 모터들에 비해서 낮은 비용과 높은 토크밀도 등 여러 가지 장점을 가지고 있다. 그러나 자체적인 토크 리플 때문에 실제 산업체에 사용하기 힘든 문제점이 있었는데, 이러한 문제점을 해결하기 위한 최적설계문제를 개발된 EMDIOS로 해결하였다.

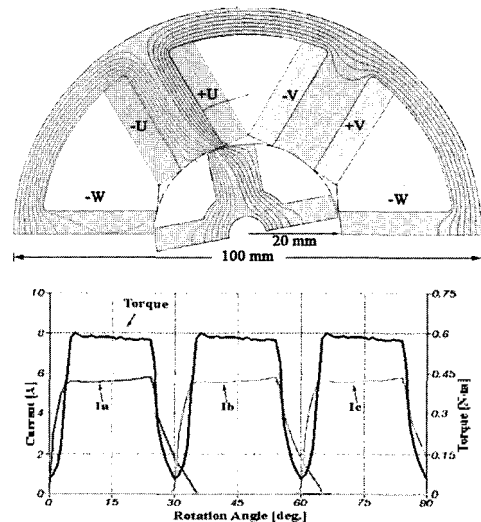


Fig. 8. Switched Reluctance Motor.

Table 2. Optimization Problem of SRM System

|                   |  |
|-------------------|--|
| Software          | 2D Finite Element Model - Switched Reluctant Motor / BLDC Designer |
| Analysis run time | 2 hour (Non linear)  |
| Objective         | torque ripple (Minimize)   |
| Constraints       | shape dimensions   |
| Design Variables  | switch on, off angle, rotation angle                               |

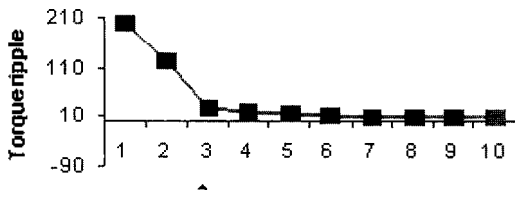


Fig. 9. Optimum result.

Fig. 9의 결과에서 보이는 것 처럼 모터의 토크리플이 10 이하로 줄었으며 이러한 최적화로 문제가 되던 모터의 토크리플이 줄어 실제 시스템 제작에 사용되었으며 이러한 최적화 문제의 해결을 통하여 시스템의 신뢰성을 확보하였다.

**3.2 Heart Dipole Problem**

개발된 시스템의 다분야통합최적화 문제 해결 능력과 상용 프레임워크를 비교하기 위한 예제로써 NASA MODOB의 Heart Dipole<sup>TM</sup>을 선정하였다. 비교를 위하여 ModelCenter, iSIGHT를 사용하였으며 최적화 문제는 아래와 같다.

$$\begin{aligned}
 f_1 &= x_1 + x_2 \cdot d_{mx} = 0 \\
 f_2 &= x_3 + x_4 - d_{my} = 0 \\
 f_3 &= x_5x_1 + x_6x_2 - x_7x_3 - x_8x_4 - d_A = 0 \\
 f_4 &= x_7x_1 + x_8x_2 + x_5x_3 + x_6x_4 - d_B = 0 \\
 f_5 &= x_1(x_5^2 - x_7^2) - 2x_3x_5x_7 + x_2(x_6^2 - x_8^2) - 2x_4x_6x_8 - d_C = 0 \\
 f_6 &= x_1(x_5^2 - x_7^2) + 2x_1x_5x_7 + x_4(x_6^2 - x_8^2) - 2x_2x_6x_8 - d_D = 0 \\
 f_7 &= x_1x_5(x_5^2 - 3x_7^2) + x_3x_7(x_7^2 - 3x_5^2) + x_2x_6(x_6^2 - 3x_8^2) \\
 &\quad + x_4x_8(x_8^2 - 2x_6^2) - d_E = 0 \\
 f_8 &= x_3x_5(x_5^2 - 3x_7^2) + x_1x_7(x_7^2 - 3x_5^2) + x_4x_6(x_6^2 - 3x_8^2) \\
 &\quad - x_2x_8(x_8^2 - 2x_6^2) - d_F = 0 \\
 \text{Find} &\quad x_1, x_4, x_6, x_7 \\
 \text{to Minimize} &\quad f_5 + f_6 + f_7 + f_8
 \end{aligned}$$

$$\begin{aligned}
 \text{Subject to} &\quad f_j \geq 0, \quad j = 5, 6, 7, 8 \\
 \text{Analysis 1} &\quad x_2, x_5 \rightarrow \\
 \text{Analysis 2} &\quad f_5 = 0 \\
 &\quad \leftarrow x_3, x_8 \quad f_4 = 0
 \end{aligned}$$

여기서는 방법론 중 MDF, IDF, CO를 각각 모델링하여 최적화를 수행 하였으며 이에 대한 결과를 아래 표에 요약하였다.

테이블에 나타난 최적화 결과는 프레임워크들 간에 약간 상이한 값을 나타냈으나, 이는 내장된 최적화 모듈이나 최적화 방법 및 내부 파라미터가 다르기 때문이다. 이 중 EMDIOS는 MDO 문제의 해석 영역을 병렬로 수행하여 다른 두 소프트웨어 비해 MDO 프레임워크로서의 높은 효율성을 지니고 있는 것으로 나타난다.

**4. 결 론**

본 연구에서는 다분야통합최적설계 문제들을 효과적으로 표현, 설정하고, 해결하기 위한 프레임워크를 개발하였으며, 기존 프레임워크의 문제점을 해결하기 위해서 패턴 이키택처인 EDOM과 EMDIOS-L을 개발 하였다.

EDOM이라는 개념으로 모든 실제 리소스를 패턴 컴포넌트화하여 조합하여 MDF, IDF, CO로의 상호 전환 및 설정이 가능하며 다분야통합 최적설계문제론 분해 및 재구성 하였으며 설정된 정보를 이용하여 병렬처리가 가능하게 되었다. 또한 설정된 모델링정보를 이용하여 DSM 구조로 변환되어 자동 분해 기법을 손쉽게 적용이 가능하였다.

모델링관점에 있어서 많은 비용을 소모하는 iSIGHT와 ModelCenter의 모델링 기법은 단일 최적설계와 리소스 통합화를 중심으로 하는 시스템이므로

Table 3. Optimum Result of Heart Dipole problem

|              |  | Parallel Support  | Heart Dipole Problem |              |               |
|--------------|--|-------------------|----------------------|--------------|---------------|
|              |  |                   | MDF                  | IDF          | CO            |
| Model Center | Final Design Objective                 | Serial Analysis   | -0.00060             | 0.00059      | 0.00461       |
|              | Final Design Max. Constraint Violation |                   | 0.00031 (f7)         | 0.00010 (f7) | -0.00159 (f7) |
| iSIGHT       | Final Design Objective                 | Serial Analysis   | 0.03481              | 0.00217      | -2.28592      |
|              | Final Design Max. Constraint Violation |                   | -0.02417 (f8)        | 0.00087 (f7) | 0.54783 (f7)  |
| EMDIOS       | Final Design Objective                 | Parallel Analysis | 0.00087              | 0.00074      | 0.00503       |
|              | Final Design Max. Constraint Violation |                   | 0.00024 (f7)         | 0.00012 (f7) | 0.00021 (f7)  |

다분야통합최적화 문제의 정의에는 적합하지 않다. 또한 다분야 통합최적화 문제 및 복잡한 설계 최적화 문제를 표현하기 위해서는 내부 구조를 이해하여야 하며 이를 바탕으로 상용 스크립트를 설계자가 작성해야 한다. 반면 EMDIOS는 다분야통합최적화 전용언어 이므로 필요 이상의 코드의 생성 없이 대상 문제를 설정하고 수정하는데 있어서 매우 효율적이다.

이를 근간으로 구성된 스크립트형 언어를 이용하여 일반적인 최적설계 문제 및 다분야통합최적화 방법론을 구성한 후 다분야 통합 최적설계문제를 통하여 기존 상용 프레임워크들과 비교 분석하여 유용성을 보 있으며, 실제 산업체 문제에 적용하여 현실적으로 매우 효과적인 시스템임을 증명하였다.

## 후 기

이 연구는 한국과학재단 지정 최적설계신기술연구센터의 연구비 지원으로 수행되었습니다.

## 참고문헌

1. Salas, A. O. and Townsend, J. C., "Framework Requirements for MDO Application Development", 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, Missouri, AIAA 98-4740, September 2-4, 1998.
2. ModelCenter Technical White Paper, Improving The Engineering Process With Software Integration, Phoenix Integration Inc.
3. iSIGHT Developer's Guide Volume 2:MDOL -Version 5.5-, Egincoous Software Inc.
4. Andrew Thomas Scott, "An Evaluation of three commercially available integrated design framework packages for Use in the Space Systems Design Lab".
5. Philip A. Wilsey, "WEB-BASED ANALYSIS AND DISTRIBUTED IP", Proceedings of the 1999 Winter Simulation Conference.
6. Alexandrov, N. M. and Lewis, R. M., "Comparative Properties of Collaborative Optimization and Other Approaches to MDO", Proceedings of the First ASMO UK/ISSMO Conference on Engineering Design Optimization, July 8-9, 1999. R. P. Weston, J.
7. Inside COM Microsoft's Component Object Model, Dale Rogerson, Microsoft Press.
8. "Optimal Design of Switched Reluctance Motor Using Two-dimensional Finite Element Method", Sol Kim, JOURNAL OF APPLIED PHYSICS.
9. Natalia Alexandrov and Srinivas Kodiyalam, "Initial Results of an MDO Method Evaluation Study. Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, Missouri, AIAA 98-4884, September 2-4, 1998.
10. An Open Computing Infrastructure That Facilitates Integrated Product And Process Development From A Decision-Based Perspective, Mark A Hale, Georgia Institute of Technology.
11. DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis, Sandia National Laboratories Albuquerque, New Mexico 87185.
12. 홍은자, 이세정, 이재호, 김승민, 다분야통합최적설계를 위한 데이터서버 중심의 컴퓨팅 기반구조, 한국CAD/CAM학회논문집, 제8권, 제4호, 2003. 12, pp. 231-242.

## 주 민 석

1994년~1998년 군산대학교 기계설계 학사  
 1998년~2000년 한양대학교 기계설계 석사  
 2000년~2005년 한양대학교 기계설계 박사과정

## 이 세 정

1976년 3월~1980년 2월 서울대학교 기계공학  
 1980년 3월~1982년 2월 한국과학기술원 기계공학  
 1986년 8월~1989년 12월 Pennsylvania State University 기계공학  
 1993년 1월~現 서울시립대 조교수

## 최 동 훈

1971년~1975년 서울대학교 기계공학 기계설계학과  
 1975년~1977년 한국과학기술원 기계공학  
 1981년~1986년 Univ. of Wisconsin-Madison 기계공학  
 1986년 3월~現 한양대학교 기계설계학과, 조교수, 부교수, 교수