

論文

병렬 컴퓨팅 기반 다분야통합최적설계 지원 설계 프레임워크

주민식*, 이용빈*, 이세정**, 최동훈***

Parallel Computing Based Design Framework for Multidisciplinary Design Optimization

Min Sik Chu*, Yong bin Lee*, Se Jung Lee** and Dong Hoon Choi***

ABSTRACT

A parallel computing technique was applied to large scale structure analysis or aerodynamic design and it is a essential element in reducing the huge computation time for large scale design problem. We can use a many computers for reducing the analysis time of multidisciplinary design optimization.

But previous MDO frameworks can not support a parallel design process technique so still existing which calls an analysis program continuously. In this paper, We developed a MDO framework(MLR) which supports a parallel design process to solve sequential analysis call. Finally, three sample cases are presented to show the efficiency of design time using the suggested MDO framework.

초 록

엔지니어링 분야의 병렬 컴퓨터 시스템은 일반적으로 초대형 구조해석이나 항공분야에 많이 적용되어 대형 설계문제의 긴 해석시간을 단축하였다. 슈퍼컴퓨터나 다수의 컴퓨터를 사용하여 해석시간을 단축하는 효과는 다분야통합최적설계의 설계시간을 줄이는데 사용할 수가 있다. 하지만 기존의 상용 MDO 프레임워크의 다분야통합최적화 설계 프로세스는 해석 프로그램을 순차적으로 호출하는 방식으로 동작하여 설계 해를 도출하는 방식으로 비효율적이다. 본 연구에서는 이런 문제를 해결하기 위해서 병렬 설계 프로세스를 도입하여 수행할 수 있는 MDO 프레임워크를 개발하였다. 개발된 MDO 프레임워크를 검증하기 위해서 수식 문제 및 모터설계 문제와 헬기설계 문제를 적용하여 유효성을 검증하였으며, 설계 해를 도출하기 까지 걸리는 총 설계시간을 혁신적으로 줄임으로써 기존의 MDO 프레임워크에 비해 우수성을 증명하였다.

Key Words : MDO Framework (MDO 프레임워크) , Large-scale Structural Analysis(초대형 구조 해석), Parallel Computing(병렬 컴퓨팅), Multidisciplinary Design Optimization (다분야통합최적설계)

† 2005년 1월 11일 접수 ~ 2005년 4월 15일 심사완료

* 정회원, 한양대학교 기계설계학과 대학원
연락처자, E-mail : mdoframe@ihanyang.ac.kr
서울시 성동구 해당동 17 HIT 312호

** 정회원, 서울시립대 기계설계학과

*** 정회원, 한양대학교 기계설계학과

1. 서 론

CRAY XMP, YMP, CRAY-2 C90등의 과거의 슈퍼컴퓨터는 벡터프로세싱 기법을 사용하여 고속으로 과학계산에 필요한 계산을 수행하였다. 이러한 슈퍼컴퓨터는 개발원가와 사용자의 비용

부담 및 관리비용이 큰 시스템이지만 대형 구조 문제나 유동해석의 수행에 있어서 필수적인 요소이며, 항공분야 설계에 있어선 중요한 위치에 있다. 하지만 현재는 슈퍼컴퓨터보다는 SMP[1]형 컴퓨터와 MPP[1]형 컴퓨터를 이용한 “분산병렬 처리” 컴퓨터 환경이 자리 잡고 있으며, 항공 우주비행체의 가상 시뮬레이션이나, 수백만 자유도 이상의 항공기 동체 전체구조모델에 대한 고정밀 안정성 해석을 위주로 주로 해석시간을 줄이는 역할을 담당하고 있다.

한편 지금까지의 최적설계는 주로 단일분야로 한정된 범위에서 수행되어 왔다. 그러나 고성능 다기능 고부가가치 제품을 개발하기 위해서는 제품설계와 관련된 여러 분야의 공학적 원리들(구조해석, 동역학, 열 유체유동해석, 제어, 전자기장 해석 등)을 동시에 고려하며 각 분야간 상충된 설계조건들을 통합적으로 다루는 설계방법이 필요하다. 이를 통칭하여 다분야통합최적설계(Multidisciplinary Design Optimization: MDO)라 부른다.

MDO 기술은 여러 분야의 공학적 설계원리들을 동시에 고려하여 균형 있고 유기적인 방법으로 최적설계 결과를 구하는 설계자동화기술이며, 설계자동화를 위한 필수적인 요소가 MDO 프레임워크이다.

MDO 프레임워크는 다양한 해석 프로그램을 유기적으로 연결하고 연결된 해석 프로그램을 최적 해를 도출할 때까지 자동으로 설계문제를 수행하는 시스템이다.

하지만 기존의 MDO 프레임워크는 스크립트 중심의 순차적인 설계 프로세스 실행 방식으로 설계문제를 수행한다는 문제점을 가지고 있다[2].

이러한 순차적인 해석 프로그램의 호출 방식은 MDO 해를 도출하는데 있어서 가장 큰 장애물이었다. 이를 해결하는 방법은 해석 횟수를 줄이는 알고리즘의 사용과 위에서 언급한 고성능 컴퓨터를 사용하여 해석 프로그램의 수행 시간 자체를 줄이는 방법이 있다. 하지만 최적화 수행 시 “순차적으로 해석 프로그램을 호출”한다는 문제점은 여전히 존재하게 된다. 이런 문제를 해결하기 위해서 MDO 프레임워크는 해석분야에 적용되는 분산병렬처리기술과는 다른 최적화과정이나 민감도 해석시 필요한 다중으로 해석 프로세스를 실행하는 것과 연성관계가 없는 설계 프로세스를 병렬화하는 기법을 제안하고자 한다.

본 연구에서는 최적설계 프로세스를 병렬처리하는 기법이 도입된 MDO 프레임워크를 개발하였으며 개발된 MDO 프레임워크의 신뢰성을 확

보하기 위해 수식 예제를 테스트 하였고, 실제 장시간이 소요되는 실제 상용해석 문제를 수행하여 실용성을 증명하였다.

II. 본 론

상용 MDO 프레임워크(iSIGHT, ModelCenter[2])의 설계프로세스는 단일흐름 일괄처리(Single-stream batch processing)방식으로 여러 가지 해석 프로그램을 실행한다.

하지만 최적설계 프로세스를 순차적으로 수행하지 않고 병렬로 처리한다면 총 설계시간은 크게 줄어들 것이다. 또한 다분야통합최적설계의 다단계 설계절차의 특성을 고려했을 경우 각각 설계절차의 최적화 수행단계에서도 병렬처리를 적용한다면 기존의 다분야통합최적설계의 총 설계시간 역시 획기적으로 줄일 수가 있다.

최적설계 과정에 병렬처리 기법을 도입하기 위해서는 총 세 가지의 기술이 요구된다. 첫 번째는 병렬영역의 설정할 수 있는 모델링 기법으로, 설계 프로세스에서 해석 프로그램간의 연성관계가 없는 부분을 병렬처리 영역으로 정식화 하는 기법으로 MDO 프레임워크로 하여금 병렬화 영역을 알 수 있게 하는 기술이다.

두 번째는 정식화된 병렬 영역을 원격으로 처리할 수 있는 기술로 분산 객체 기술이 필요하다. 분산 객체 기술은 병렬 프로세스로 분류되어 할당된 해당 프로세스를 로컬이 아닌 다른 원격 컴퓨터에서 수행하는 역할을 담당하는 기술이다.

세 번째는 이러한 병렬처리 기술과 분산 객체 기술을 적용할 수 있도록 개발된 MDO 프레임워크이다.

2.1 객체지향 MDO 프레임워크 MLR

상용 MDO 프레임워크는 해석 프로그램을 통합을 위주로 최적설계를 수행하는 구조를 중점으로 두고 있다. 이런 시뮬레이션 기반 프레임워크는 대부분 설계 프로세스를 상용 스크립트로 생성하고 생성된 스크립트를 단일처리 흐름으로 수행하는 형태이다. 스크립트중심의 MDO 프레임워크들의 문제점은 임의로 설계 프로세스를 정식화하기 어렵다.

때문에 DSM(Design Structure Matrix)을 구성하지 못하여 시스템 분해기법을 도입하지 못한다는 것이다. 또한 병렬로 정식화 역시 불가능하기 때문에 설계시간이 비효율적이라는 문제점이 있다. 본 연구에서는 기존에 선행연구로 개발된 “분산 환경 기반 MDO 프레임워크”[3]를 기초로 본 연구에서는 XML[4]을 이용한 설계문제를 객체지향 개념을

```
<?xml version="1.0" encoding="utf-8" standalone="no" ?>
(ContainerControlObject Root[ID="MLR" MLRVersion="0.1" version="0.1" xmlns:space="preserve"])
  (Object Classtype = "CDesign_Optimization_ADSClass" name = "MLR")
    (Operators)
      (Operator name="OBJ" order="1" SCRIPT="VBSCRIPT">F5-F6-F7-F8</Operator>
      (Operator name="G1" order="1" SCRIPT="VBSCRIPT">J1-0.0001</Operator>
    (Operators)
  (Properties)
    (Property name="Z1" TYPE = "DESVAR" LB="460" UB="560">460</Property>
    (Property name="Z2" TYPE = "DESVAR" LB="1.8" UB="2.6">2.7</Property>
    (Property name="Z3" TYPE = "DESVAR" LB="0.03" UB="0.05">0.04</Property>
  (Properties)
</Object>

(Object name="SYSTEM_ANAL" Classtype="ApplicationObject")
  (Properties)
    (Property name="X1">460</Property>
    (Property name="X2">2.7</Property>
  (Properties)
</Properties>
  (Operators)
</Operators>
</Object>

(Object name="SUB_ANAL" Classtype="ApplicationObject")
  (Properties)
    (Property name="description" User Driver</Property>
    (Property name="VT">460</Property>
  (Properties)
</Properties>
  (Operators)
    (Operator name="R1" input="X1 X2" output = "A1"</Operator>
  (Operators)
</Operators>
</Object>

(DoDataExchange)
  (LinkTo From="mir" FROM_ID="VT" To="init" TO_ID="VT">0.00</LinkTo>
  (LinkTo From="noise" FROM_ID="EPNL" To="mir" TO_ID="EPNL">0.00</LinkTo>
  (LinkTo From="docs" FROM_ID="DOCS" To="mir" TO_ID="DOCS">0.00</LinkTo>
  (LinkTo From="hndqit" FROM_ID="TAUTO" To="mir" TO_ID="TAUTO">0.00</LinkTo>
</DoDataExchange>
</ContainerControlObject>
```

그림 1. XML로 정식화된 최적설계문제

이용하여 정식화하는 MDO 프레임워크인 MLR (Multidisciplinary Language Runtime)을 개발하였다. MLR은 그림 1과 같이 설계문제를 XML로 표현하여 설계문제를 인식한다. 설계문제를 XML로 표현한다는 것은 표준의 메타 모델링언어로 표현한다는 장점이 있지만, 해석프로그램, 최적화 모듈과 같은 설계 리소스 전부를 논리적인 객체로의 표현이 매우 자유롭다는 장점이 있다.

그림 1은 하나의 해석 프로그램과 최적화 모듈과 민감도 해석 단계를 객체로 모두 표현한 하나의 예이다. 즉 MLR에서는 최적화, 해석 프로그램, 민감도 해석 등을 모두 객체로 인식하며, 이를 XML로 정의한다. 다분야통합최적설계문제에 관련된 설계자원

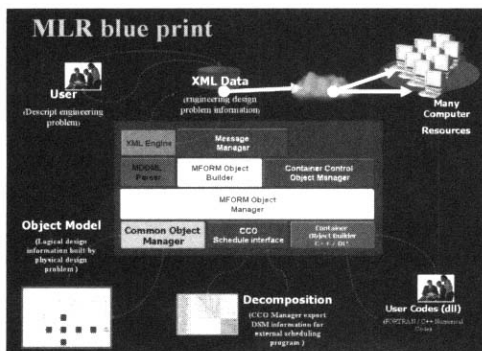


그림 2. MLR 청사진

은 모두 객체로 표현이 되며 그림 1과같이 Object와 Object의 Operators, Properties로 표현하며 객체간 변수 교환을 DoDataExchange로 표현한다.

그림 1에서 XML로 객체화되어 표현된 객체와 객체간의 상호관계를 가지고 그림 2에서 XML Data를 XML Engine과 MDOML Parser 에 의해서 설계문제를 분석한 뒤 이를 Object Model Manager이라는 객체 관리자에 의해 DSM 구조로 설계 프로세스를 변환한다. 얻어낸 DSM 정보를 이용하여 외부분해 기법[5]과 연동하여 후방연성이 최소화된 구조로 설계 프로세스를 MLR은 가질 수 있다.

2.2 병렬 설계 프로세스-1

위에서 언급했듯이 본 연구에서 개발한 MDO 프레임워크는 DSM기반으로 동작하는 시스템이며 외부 분해기법을 이용하여 최적화된 설계 프로세스를 얻을 수가 있다.

하지만 DSM 기반의 동작방식도 병렬화를 고려하지 않는 기법이기 때문에 병렬처리기술을 적용하는데 문제가 있다.

그림 3은 헬기설계문제[5]로 17(FINAL은 제외)개의 해석프로그램으로 연성되어 있으며 이를 DSM 구조로 변환 후 분해기법을 적용하면 그림 4

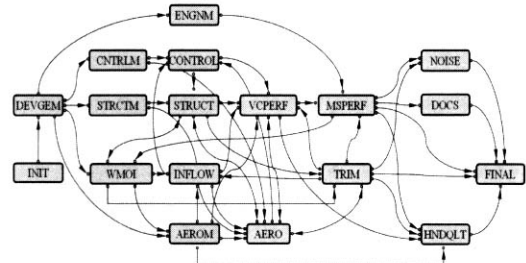


그림 3. 헬기설계 문제의 해석연성관계

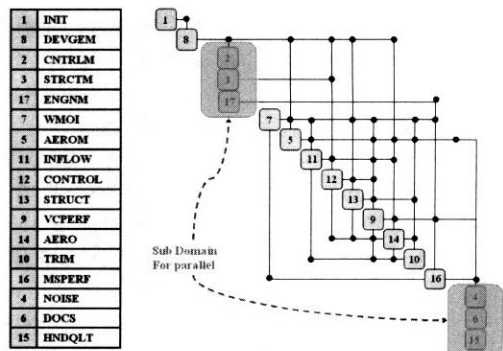


그림 4. 재구성된 설계문제

와 같은 형태로 설계 프로세스가 정해진다.

분해기법을 통해 나온 설계 프로세스 중 연성 관계가 없는 영역은 하부 영역화 하여 또 하나의 자체적인 프로세스를 가지는 영역으로 정의한다 면 13개의 해석영역으로 다시 재정의 할 수 있다. 그림4의 형태를 MLR은 총 13개의 객체로 정의한다 즉 하위 도메인으로 분류된 영역도 하나로 객체로 인식을 하고, 그 객체 내부는 외부와는 다른 병렬형태의 설계 프로세스를 가지고 있는 구조이다.

이러한 설계 프로세스를 DSM 기반으로 수행 하면서 병렬화도 수행 할 수가 있으므로 기존의 MDO 프레임워크의 수행방식보다 매우 효과적이다.

다분야통합최적설계문제의 CO(Collaborative Optimization)나 IDF(Individual Discipline Feasible) 같은 문제 역시 병렬화가 필요한 부분은 위의 그림 4와 같이 하부 도메인 영역으로 정식화하여 실행한다면 총 설계 시간을 단축할 수가 있다.

2.3 병렬 설계 프로세스-2

하부 도메인화 된 영역은 주 도메인과 분리된 설계 프로세스를 가지는 객체로 구현이 된다고 2.2절에서 설명하였다. 주 도메인과 분리하는 이유는 이질적인 프로세스를 관리를 용이하게 해준다.

본 절에서는 2.2절과는 다른 설계 프로세스가 아닌 최적화의 중간 단계에서 요청하는 민감도 해석이나, RSM 기법에서 요구하는 실험 점들을 병렬로 하고자할 경우에 MDO 프레임워크에서 어떻게 이를 수행하느냐는 것이다. 이를 해결하기 위해서 하부 도메인에 해당하는 객체에 해당하는 클래스를 상속받아 새로운 클래스로 구현하여 수행하는 방법을 채택하였다.

그림 5는 MLR 일부를 리버스 엔지니어링[5]한 것으로 MLR의 병렬처리 관련 클래스를 상속 받아 ADS[6]라는 최적화 코드를 내장한 예이다.

CObject 클래스는 MLR에서 사용되는 하나의 객체단위를 표현하는 기본 클래스이다. 이를 상속 받은 CContainControlObject는 수치해석코드와 연성관계가 있는 하위객체(해석프로그램이나 수식)를 관리하고 수행하며 동작을 관리하는 설계 프로세스 관리 클래스이며, DSM기반으로 하위객체들을 동작시킨다.

CparallelCtnerCtrlObject 클래스는 ContainerControl 클래스를 상속 받아 하위객체들을 수행할 경우 DSM 기반이 아닌 병렬로 수행하는 기능이 구현

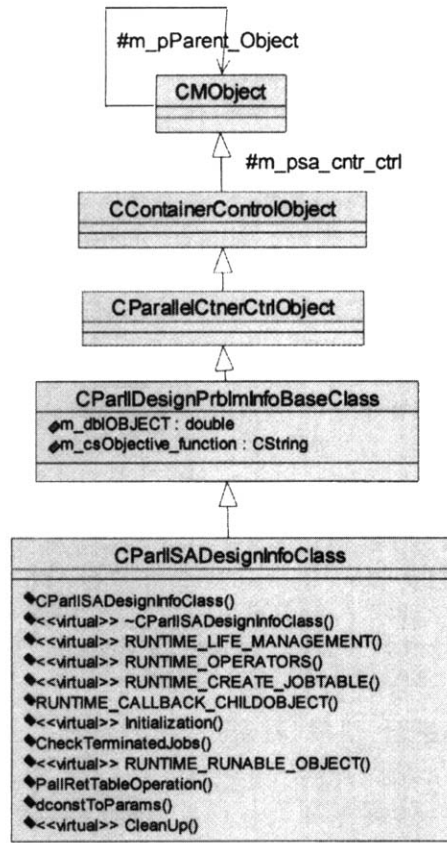


그림 5. 병렬처리 관련 코드의 역 엔지니어링 화면

되었다.

CparallelDesignPblmInfoBaseClass는 CparallelCtnerCtrlObject을 상속 받아 설계문제 관리 기능을 추가한 것으로 최종적으로 사용자는 CparallelDesignPblmInfoBaseClass를 상속 받아 병렬기반 설계프로세스 관리 클래스를 이용하여 최적화 모듈을 MLR에 구현할 수가 있다. 본 연구에서는 CparallelDesignPblmInfoBaseClass를 상속 받아 ADS와 PQRS[7] 코드를 내장하였다. 병렬처리관련 클래스가 담당하는 하부 도메인영역은 최적화 수행 시 설계자의 추가적인 코딩 없이 해당 영역을 병렬로 해석 프로세스를 실행하고 관리하며 자동으로 로드 밸런싱을 수행하며 병렬로 수행되는 객체를 멀티스레드[8]를 통해서 생성하고 분담하는 기능을 가지게 된다.

2.4 병렬 프로세스 처리

본 연구에서는 병렬화를 위해 하부 도메인으로 설정된 영역은 멀티스레딩과 병렬 프로세스

기능을 자체적으로 제공하도록 개발되었다.

멀티스레딩은 스케줄링 및 프로세스가 되는 스레드라고 불리는 경량형 메커니즘으로 구현된 것으로 사용자의 요청을 관리한다. 병렬 처리는 대규모 사용자 작업을 하위 작업으로 분할시켜 모든 가용 자원상의 수행이 분산되도록 하는 것이다.

이 두 기술은 모든 시스템 자원이 가장 효율적으로 사용되도록 하고자 하는데 목적이 있다. 예로써 실험 계획법에 의하여 실험점의 개수가 10개가 생성 되었을 때 MDO 프레임워크는 10개의 수행 작업을 생성 하고 설정된 가용 컴퓨터 정보를 얻어 온다. 원격의 가용 컴퓨터의 개수가 총 5개의 경우에는 멀티스레드를 5개를 동시에 생성하여 동시에 수행을 하며, 이 멀티스레드는 분산 객체에게 수행할 정보를 넘기며 해당 테스트가 끝날 때까지 기다린다. 이때 외부에 등록된 컴퓨터는 모두 성능이 다르므로 같은 시간에 종료되지 않으며, 종료된 컴퓨터의 숫자만큼 다시 멀티스레드 코드를 생성하여 분산객체를 활성화 하는 로드 밸런싱 작업을 수행한다.

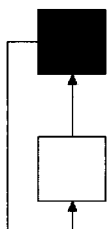
2.5 분산 객체 기술

분산객체 기술은 객체지향 개념을 도입해 구현한 객체들 간의 통신을 원활하게 만들어 주는 일종의 '소프트웨어 버스'와 같은 것으로 클라이언트와 서버 객체 간의 규약이며 이렇게 객체를 분산하여 실행하는 기술로는 DCE/RPC, CORBA, DCOM, JAVA RMI[9]가 존재한다.

본 연구에서는 RPC[9] 규약에 의한 코딩으로 구현되어 있다. RPC에 의한 코딩 방식은 프로그래머의 코딩 부담 비중이 크지만 향후 MS WINDOWS[10]환경이 아닌 다른 OS 플랫폼과의 연동을 고려할시 연동이 손쉽게 가능하다.

이러한 분산객체 기술을 이용하여 본 연구에

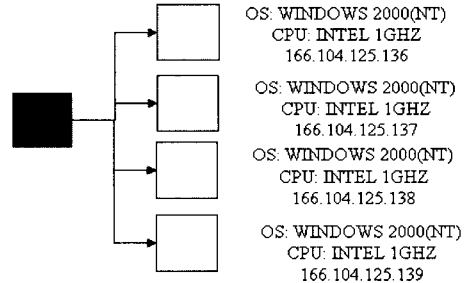
Serial System Design Configuration



OS: WINDOWS 2000(NT)
CPU: INTEL 4.3 GHZ
166.104.125.119

그림 6. 일반적인 최적화를 위한 구성 환경

Parallel System Design Configuration



OS: WINDOWS 2000(NT)
CPU: INTEL 1GHZ
166.104.125.136
OS: WINDOWS 2000(NT)
CPU: INTEL 1GHZ
166.104.125.137
OS: WINDOWS 2000(NT)
CPU: INTEL 1GHZ
166.104.125.138
OS: WINDOWS 2000(NT)
CPU: INTEL 1GHZ
166.104.125.139

그림 7. 최적설계를 위한 분산 병렬컴퓨팅 구성 환경

서는 클라이언트/서버 구조가 아닌 Peer To Peer(P2P) 구조의 시스템으로 구현되었으며 P2P 구조의 분산객체 구현방식은 클라이언트/서버의 역할이 가변적으로 변할 수 있다는 장점이 있다.

일반적인 분산 컴퓨팅 기술을 이용하여 설계 문제를 구현한다면 그림 6 과 같은 형태로 구현이 된다. 하지만 효율적인 분산 병렬컴퓨팅 환경은 그림 7과 같이 구성이 되어야한다 다음 3절에 나오는 각각의 예제들은 그림 7과 같이 각 상황에 따라서 5대의 컴퓨터에서 6대의 컴퓨터를 이용하여 최적설계를 수행하였다.

2.6 예제

예제1 수식 문제

시스템의 신뢰성을 확보하기 위해서 첫 예제로는 참고문헌[11]의 QQR-T1-6 의 수식 예제를 선택하였으며 최적화 기법은 ADS의 MMFD (Modified Method of Feasible Direction)와 SLP(Sequential Linear Programming)를 사용하였다. 수행된 결과는 표 1과 같다.

$$f(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$
$$g_1(x) = -x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8 \geq 0$$
$$g_2(x) = -x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 9 \geq 0$$
$$g_3(x) = -2x_1^2 - x_2^2 - x_3^2 - 2x_4^2 + x_2 + x_4 + 5 \geq 0$$

그림 9. QQR-T1-6 문제

표 1의 SOLUTION은 실제 참고문헌에서 제시하는 최적해이며 MLR이 도출한 최적 해는 MMFD의 지역 최적 값이다. SLP의 최적화 결과

표 1. QQR-T1-6 수식 예제결과

	SOLUTION	MLR
Objective Function	-44.000000	-44.122931
X1	0.000000	-0.041588
X2	1.000000	0.848440
X3	2.000000	2.031090
X4	-1.000000	-1.082980

역시 이와 유사하다. 병렬처리의 효율성을 가지 적으로 평가하기 위해서 해석영역에 임의의 가장 시간을 10초씩 추가하여 수행한 결과는 표 2와 같다.

표 2. QQR-T1-6의 병렬컴퓨팅을 적용한 결과

최적화 방법	함수 호출 횟수	수행 시간(sec)
MMFD(S1)	76	836
SLP(S1)	75	825
MMFD(P3)	56(76)	616
MMFD(P5)	45(76)	495
SLP(P5)	30(75)	330

표 2는 병렬로 설계 프로세스를 수행하면서 두 가지 최적화 기법을 적용 시 나온 결과이며 S1, P3의 의미는 S는 단일 컴퓨터 1대를 적용이라는 의미이며 P는 병렬로 처리된 3대의 분산된 컴퓨터를 적용 했다는 의미이다. 본 문제는 MMFD, SLP 방법을 단일 컴퓨터로 적용 시 걸린 시간과 3대, 5대 컴퓨터를 추가하여 분산 병렬 컴퓨팅 기반으로 수행했을 시 걸리는 시간을 측정하였다. 표 2의 괄호 안에 숫자는 원래 함수 호출 횟수이다.

예제2 모터 설계 문제

본 예제는 Switched reluctant motor 설계문제 로써 스위치드 릴럭턴스 모터는 기존의 모터들에 비해서 낮은 비용과 높은 토크밀도 등 여러 가지 장점을 가지고 있다.

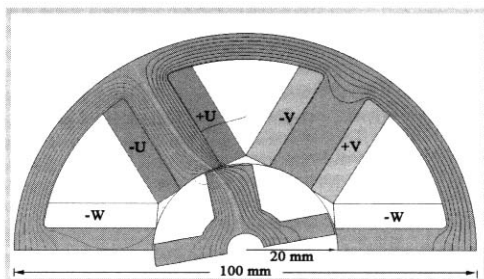


그림 8. SRM 해석프로그램

표 3. SRM 최적화 문제 정보

Software	2D Finite Element Model Switched Reluctant Motor / BLDC Designer
Analysis run time	20min(Linear), 2hour(Non linear)
Objective	torque(Maximize)
Constraints	shape dimensions, torque ripple
Design Variables	switch on, off angle, rotation angle

그러나 자체적인 토크 리플 등에 의한 문제로 인해 실제 산업체에서 사용하기 힘든 문제점이 있으며 이러한 문제점을 해결하기 위해서 최적설계를 수행하였다. 설계문제는 표 3과 같으며 최적화 방법은 PQRSM[9]기법을 사용하여 설계 해를 도출 하였다.

표 4. SRM 설계문제 최적화결과

PQRSM	초기값	최적값
objective function	-1.81E+00	-3.08E-01
X1	25.2	15.49104
X2	45.1	49.92188
X3	30,0	46.28756
G1		-1.98E-04
G2		-2.73E-01

표 4는 PQRSM기법을 적용 시 나온 결과이며 이 최적 해를 얻기 위해서 걸리는 시간은 선형해석 프로그램을 사용할시 24시간이 소요되었으며, 또한 같은 모델로 비선형 해석을 적용한 경우 6일 정도가 소요되었다.

표 5. SRM 해석을 병렬 컴퓨팅 적용한 최적화 결과

최적화 방법	함수 호출횟수	해석시간 (분)	해석시간 비율
PQRSM(S1)	71	1420	100%
PQRSM(P6)	21	420	30%

표 5는 선형 해석을 적용하여 최적 해를 낸 결과이다. 최적화를 수행했을 경우에는 병렬처리를 한 결과 7시간 만에 결과를 얻었으며, 비선형 해석을 적용하였을 경우에 병렬 컴퓨팅을 도입할 경우 2.5일 정도 소요되었다. 이렇듯 병렬 컴퓨팅 기법을 도입 시에는 기존의 순차적인 방법에 비해 최적 해를 얻는데 혁신적인 단축 효과를 얻을 수 있다.

예제3 헬기설계문제

앞의 2.1절에서 언급한 헬기설계 문제에 대한 최적화 설계문제는 표 6과 같다. 총 17개의 해석

표 6. 헬기중량 최소화를 위한 최적설계문제

VT : tip speed DL : main rotor disk loading SGMA : main rotor solidity ratio THETA : main rotor blade twist GW : gross weight HPREQ : shaft power EPNL : noise DOCS : operating cost ALFUSBR : attack angle TAUTO : autorotation entry time	
Objective	GW
Constraints	HPREQ < 95
	EPNL < 90
	DOCS < 14
	-0.15 < ALFUSBR < 0.035
	TAUTO > 0.15
Design Variables	VT , DL , SGMA , THETA

모듈이 서로 연성되어 있으며, 공력, 구조 등의 연성 외에도 로터블레이드의 회전과 탄성 거동의 영향으로 인한 후류 및 동적 특성들이 복잡하게 연관되어 있다. 따라서 최적화 문제에 직접 적용하기에는 과도한 계산시간이 드는 단점이 있으므로 이를 외부 분해 기법으로 설계 프로세스를 최적화 한 후 그림 4와 같이 병렬화 할 수 있는 부분을 정식화 하여 최적화 하였으며 이에 대한 결과는 아래 표 7과 같다.

표 7. 병렬분산 컴퓨팅을 헬기설계에 이용한 결과

최적화 방법	해석시간(분)	해석시간 비율
SQP(S1)	65	100%
SQP(P3)	35	45%

III. 결 론

엔지니어링 해석 분야에 사용되는 병렬 컴퓨팅과 최적설계에서 필요한 병렬 컴퓨팅 기술에 대해서 기술하였다. 다분야통합최적설계에서 일반적으로 사용하는 순차적 설계 프로세스 진행방식이 아닌 병렬로 설계 프로세스를 정식화하고 실행하는 기술을 도입하여 MDO 프레임워크를 개발하였다. 이를 검증하기 위해서 수식 예제를 적용하여 신뢰성을 증명하였으며, 실제 해석시간이 긴 상용 해석 문제를 최적화하여 결과를 도출한 후 순차적으로 해석 프로그램을 호출하여 최

적 해를 도출하는 방식과 병렬 컴퓨터 기반으로 해석 프로그램을 호출하여 도출된 최적 값과 총 설계시간을 비교 하였다. 실제 모터설계 문제의 경우 최적 해를 얻는데 기존의 순차적인 방식에 비교하여 5대의 컴퓨터를 추가로 적용하여 70%의 총 설계시간이 줄어드는 효과를 보였다. 이 결과로 인하여 매우 효율적인 시스템임을 증명하였으며, 향후 해석 시간이 긴 최적설계문제나 다분야통합최적설계 문제의 경우 매우 유용한 시스템으로 적용되어질 것으로 사료된다.

후 기

이 연구는 한국과학재단 지정 최적설계신기술 연구센터의 연구비 지원으로 수행되었습니다.

참고문헌

- 1) Cluster Computing White Paper Version2.0, Edited by M. Baker, University of Portsmouth, UK, Sept, 2000, <http://www.csm.port.ac.uk/~mab/Personal/pubs.html>.
- 2) Andrew Thomas Scott, John Olds, An evaluation of three commercially available integrated design framework packages for use in the Space System Design Lab, April 26, 2001.
- 3) 주민식, 최병렬, 이세정, 최동훈, 다분야 통합최적설계 지원 분산환경 프레임워크 개발, 항공우주학회. Proceedings of the KSAS Fall Annual Meeting 2001, pp372.
- 4) Beginning XML ,David Hunter의 5인 공저, Wrox Press LTD.
- 5) 박형욱, 최동훈, 안병호, "다분야통합최적설계를 위한 적응분해기법", 한국항공우주학회지 제 31권 5호 2003, pp. 18-24
- 6) G. N. Vanderplaats, "ADS- A FORTRAN PROGRAM FOR AUTOMATED DESIGN SYNTHESIS", January, 1987.
- 7) Kyung-Jin Hong, 2001, "Efficient Approximation Method for Constructing Quadratic Response Surface Model", KSME international Journal, Vol.15, No.7, pp. 876~888.
- 8) Multithread, www.msdn.microsoft.com/library/kor/vccore/html/_core_multithreading_with_c.2b2b_and_mfc.asp
- 9) Jim Farley, "JAVA Distributed Computing",

O' REILLY. January 1998.

10) www.microsoft.com/windows.

11) Basile, A., Drexl, A., Dawid, H., Inderfurth, K., Kürsten, W., Schittko, U. "Lecture Notes in Economics and Mathematical Systems", Editors-in-chief: Fandel, G., Trockel, W.

12) Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design Pattern", Addison-Wesley, 2002.

13) Vanderplaats, G. N., Numerical Optimization Techniques for Engineering Design, 3rd Ed. VR&D, Inc., 1999.