다분야통합 설계 최적화(MDO) 문제의 정식화 기법에 대한 고찰

Part 2: MDO 정식화 기법의 종류와 특성

양영순, 정현승 〈서울대학교 조선해양공학과〉

1. MDO 기법의 분류

MDO 정식화 기법에 대한 분류는 Cramer 등(1993)의 논문에서 최초로 시도되었다. 이 논문에서는 정식화의 대상으로 공탄성(aeroelastic) 최적화 문제와 같이 분야간 해석의 연성만이 존재하는 문제를 다루었다. 즉 설계공간의 분해를 고려하지 않은 단일단계 MDO 기법들에 대한 분류이다. Cramer 등은 단일단계 MDO 기법을 최적점을 찾아가는 과정에서 생성되는 설계점들의 해석 만족여부(analysis feasibility)에 따라 MDF (Multi-Disciplinary Feasible) 기법, IDF (Individual Discipline Feasible) 기법 및 AAO(All At Once) 기법 등의 3가지로 분류하였다.

이후 Balling 등(1996)은 Cramer 등의 논문을 기초로 설계공간의 분해에 따른 다단계 최적화 기법들을 추가하여 6가지 기본적인 기법으로 분류하였고, 이를 구분하기 위한 간략한 표기법을 제시하였다. 이 논문에서는 각각의 기법들을 3가지 기준으로 구분하였는데, 첫째는 설계변수의 결정이 하나의 최적화모듈에서 결정되는 단일단계(single-level)기법과 설계변수가 분산된 여러 최적화모듈에서 결정되는 다단계(multilevel)기법의 구분이다. 둘째는 시스템단계에서의 SAND와 NAND의 구분으로 다분야통합해석을 수행하는 것을 NAND로, 연성변수를 도입해서 분야 각각의 해석을 독립적으로 수행하는 것을 SAND로 구분하였다. 마지막은 각 분야 해석 단계에서의 SAND와 NAND에 관한 것으로 각 분야의 상태변수를 설계변수에 포함하여 최적화모듈이 상태방정식의 해도 찾는 것을 SAND, 그렇지 않은 것을 NAND로 구분하였다. 이러한 것들을 기준으로 Single-SAND-SAND, Single-SAND-NAND, Single-NAND-NAND, Multi-SAND-SAND, Multi-SAND-NAND, Multi-NAND-NAND 의 6가지 기본적인 기법으로 분류하였다. 이 논문에서 Balling 등은 MDO의 Discipline을 좀 더 확장된 개념으로 정의하였다. 즉, 시스템 설계에 관련된 각각의 학문분야가 discipline으로 간주될 수도 있고, 전체시스템을 구성하는 개별 구성요소 자체도 discipline으로 간주될 수 있다고 하여, 기존의 다단계 구조 최적화기법들도 MDO기법에 포함시키려 하였다. 그러나, 1997년의 Balling 등의 논문에서는 MDO 기법이 세 가지 단일단계 기법들과 CO(Collaborative Optimization), CSSO(Concurrent SubSpace Optimization) 등으로 분류된 것으로 보아, 다단계 MDO 기법을 SAND와 NAND로 구분하기보다는 분해와 조정방식에 따라 CSSO와 CO로 구분하는 것이 타당하다고 여겨진다.

여기서는 단일단계 기법인 MDF, IDF, AAO와

다단계 기법인 CSSO, 수정된 CSSO 및 CO의 정식화 과정과 특성에 대하여 기술하겠다. 다음의 내용들은 Bloebaum(1991), Cramer 등(1993), Balling 등(1996), Braun(1996), Tappeta(1996), 박창규(1999), 노명일(2000) 등의 논문들을 기초로 작성되었다.

2. 일반적인 3분야 MDO 문제

여러 가지 정식화 기법들을 비교하기 위해 <그림 1>과 같은 3개의 분야를 가진 시스템을 고려하자. 3개의 분야로 구성된 시스템은 다루기에 간단하면서도 더 많은 분야들을 포함하는 시스템으로 확장하는 양식을 개발하기에 충분히 크다. 이러한 시스템으로 비행기 날개를 생각하면 분야 1, 2, 3은 각각 비행기 성능, 공기역학, 구조역학이 될 수 있다. 각 분야들은 출력에서 입력으로의 자료 교환을 통해 서로 연성되어 있다. 이제 <그림 1>에 나타난 변수와 함수들을 설명해 보자.

먼저 설계변수벡터를 살펴보자. 벡터 $\boldsymbol{x}$는 전체의 설계변수벡터를 표현하는데 사용될 것이다. $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$는 각 분야와 관련된 설계변수를 나타내는 벡터로서 전체 설계변수벡터 $\boldsymbol{x}$의 부분벡터들이다.

f_1, f_2, f_3는 각 분야의 목적함수벡터이다. 일반적으로 MDO문제는 각 분야가 목적함수를 가지는 다목적 최적화 문제로 기술될 수 있다. MDO 문제에 대한 다목적 정식화는 Balling 등(1996)의 논문과 Tappeta(1996)의 논문에 기술되어 있다. Balling 등(1996)은 다목적을 minimax formulation으로 다루었고, Tappeta(1996)는 다목적 최적화를 가중함수기법을 사용하여 단일목적 최적화 문제로 바꾸었다. 여기서는 여러 기법들의 비교를 용이하게 하기 위해 설계목적함수들이 하나의 스칼라 시스템 설계목적함수 f로 표현된다고 가정하고 단일목적 최적화 문제로 정식화하였다.

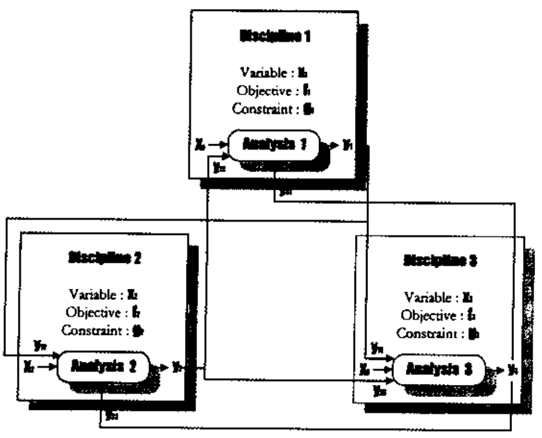

그림 1. 일반적인 3분야 연성 시스템

$$f = f(f_1, f_2, f_3)$$

벡터 $\boldsymbol{g}_1$, $\boldsymbol{g}_2$ 및 $\boldsymbol{g}_3$는 각 분야의 설계제약조건들을 담고 있다. 설계제약조건은 일반적으로 파괴 혹은 회피해야 할 거동을 방지하기 위한 것이다. 여기에서는 부등식 제약조건만을 고려하였고, 각 제약조건은 0이 허용값이 되도록 정식화되었다고 가정하여 제약함수값이 0보다 크면 제약조건을 만족한다고 가정하였다.

벡터 $\boldsymbol{y}_1$, $\boldsymbol{y}_2$ 및 $\boldsymbol{y}_3$는 각 분야의 해석의 결과로 나오는 상태변수벡터들이다. 일반적으로 설계목적함수와 설계제약함수의 값들을 구하기 위해서는 각 분야의 해석이 선행되어야 하고, 해석을 통해 상태변수의 값들을 알아내야 한다. 벡터 $\boldsymbol{y}$는 모든 분야의 상태변수벡터를 하나로 묶어놓은 벡터로 전체의 상태변수들을 통칭하기 위해서 사용된다. 분야간 해석의 연성으로 어떤 분야의 상태변수가 다른 분야의 해석의 입력으로 사용되기도 하는데 이러한 것들은 $\boldsymbol{y}_{ij}$의 형태로 표현한다. 벡터 $\boldsymbol{y}_{ij}$는 분야 i의 상태변수벡터인 $\boldsymbol{y}_i$의 일부분으로 분야 j의 입력으로 사용되는 벡터를 의미한다.

벡터 $\boldsymbol{y}_{ij}^*$는 연성변수벡터로 분야간 해석의 연성을 피하기 위해 부가적으로 도입된 설계변수의 일종이다. 각 분야의 해석에 입력으로 필요한 다른 분야의 해석 결과를 설계변수로 취급하고, 그 값이 해석의 결과인 연성함수 $\boldsymbol{y}_{ij}$와 같아야 한다는 적합

성 제약조건을 부과하여 각 분야마다 독립적으로 해석을 수행할 수 있게 된다.

벡터 $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ 및 $\boldsymbol{r}_3$는 각 분야 해석의 상태방정식의 residual을 나타내는 벡터로 AAO기법을 설명하기 위해 정의된 벡터이다.

벡터 $\boldsymbol{x}_{cv}$는 조정변수벡터로 다단계 최적화 기법을 적용할 때 조정을 위해 도입된 부가적인 설계변수를 표현하기 위해 사용된다.

벡터 $\boldsymbol{x}_{sh}$는 공유설계변수벡터로 각 분야의 설계변수가 공유되는 것이 있으면 그것을 표현하기 위해 사용되는 벡터이다.

CO의 경우, 각 분야의 자율성을 높이기 위해서 시스템 설계변수 및 각 분야의 설계변수를 독립적으로 다룬다. 이때 동일한 변수가 시스템 및 각 분야에서 사용되므로, 시스템의 변수와 각 분야의 변수들을 구분하기 위해 시스템 변수는 $(\cdot)^s$와 같이 표현하기로 하자.

3. 단일단계 MDO기법

단일단계 MDO기법들에서는 각 분야가 해석만을 담당한다. 즉, 설계변수의 결정이 각 분야로 분산되어 이루어지는 것이 아니라 하나의 시스템 최적화모듈에서 이루어짐을 의미한다. 단일단계MDO기법은 최적점을 찾아가는 동안 생성되는 설계점들의 해석 만족여부에 따라 MDF, IDF, AAO의 세 가지로 구분된다.

3.1 MDF(MultiDisciplinary Feasible)

MDF는 가장 표준적인 최적화기법을 MDO문제에 적용시킨 것이다. 즉, 최적화모듈에서 설계변수를 내려보내면, 관련된 분야들을 통합하는 다분야통합해석을 통해 각 분야의 연성을 고려한 해석을 수행하고 이의 결과로부터 얻어지는 목적함수와 제약함수의 값을 최적화모듈에 되돌려주는 과정을 통해 최적해를 찾아간다(<그림 2>). MDF가 의미하는 '다분야 만족' 이라는 것은 최적해를 찾아가는 과정에서 거쳐가는 점들이 항상 다분야통합해석을 만족하는 점이라는 것을 의미한다. Balling 등(1996)은 이러한 기법을 "Single-NAND-NAND" 라고 칭하였는데, 이러한 이름의 첫 번째 부분인 "Single"은 단일단계 최적화를 의미하고, 두 번째 부분의 "NAND"는 최적화 모듈에 다분야통합해석이 중첩되었음을 의미하고, 세 번째 부분의 "NAND"는 다분야통합해석 내에 각 분야의 해석이 중첩되었음을 의미한다.

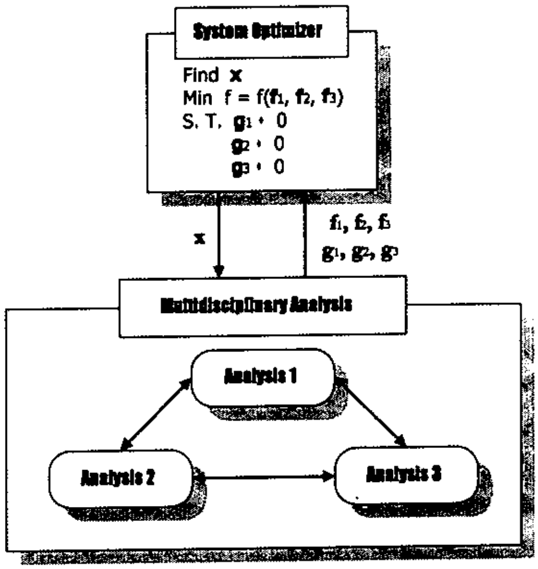

그림 2. MDF(MultiDisciplinary Feasible) 기법

MDF기법의 장점은 원래의 다른 기법들에서 채택하는 보조 설계변수들을 필요로 하지 않으므로 설계변수의 수가 증가하지 않는다는 것이다. 또한 각 분야의 해석을 black box처럼 이용하므로, 현행 해석프로그램들을 최대한 활용할 수 있다.

MDF기법의 가장 큰 단점은 다분야통합해석에 드는 비용에 있다. 분야간 해석의 연성이 복잡한 경우, 다분야통합해석을 만족하는 점을 찾기 위해 각 분야의 해석이 반복적으로 수행되어야 한다. 또한 최적화 과정에서 필요한 미분계수들을 구할 때에도 다분야통합해석이 수행되어야 한다.

MDF기법을 실제문제에 적용하는데 있어서 다분야통합해석에 드는 비용을 줄이기 위해 근사기법들이 사용되기도 한다. Hajela등(1990), Sobieski와Bloebaum 등(1991)의 논문에서는GSE(Global Sensitivity Equation)를 이용하여 각 분야 해석의 상태방정식을 선형적으로 근사하였고 이를 최적화 과정에서 사용하였다. 또한 각 분야의 제약함수를 다루는데 있어 KS함수를 이용한 누적제약함수를 사용하여 제약함수의 수를 감소시켰다.

GSE를 간략하게 설명하기 위해 다음과 같이 연성된 두 개의 해석모듈을 고려하자.

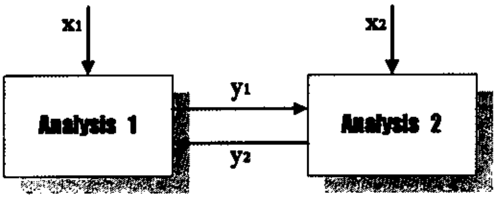

그림 3. 분야간 해석의 연성

각각의 해석의 상태방정식은 일반적으로 다음과 같은 음함수의 형태로 표현된다.

$$A_1(x_1, y_2, y_1) = 0$$
$$A_2(x_2, y_1, y_2) = 0$$

해석의 연성으로 인해 해석 1의 결과인 y_1은 x_1과 y_2의 함수이고, 해석 2의 결과인 y_2는 x_2과 y_1의 함수이다. 즉, 다음과 같이 표현된다.

$$y_1 = y_1\ (x_1,\ y_2)$$
$$y_2 = y_2\ (x_2,\ y_1)$$

위의 식을 각각 x_1과 x_2로 미분하면 다음과 같은 식을 얻을 수 있다.

$$\frac{dy_1}{dx_1} = \frac{\partial y_1}{\partial x_1} + \frac{\partial y_1}{\partial y_2}\frac{dy_2}{dx_1}$$
$$\frac{dy_2}{dx_1} = \frac{\partial y_2}{\partial y_1}\frac{dy_1}{dx_1}$$
$$\frac{dy_1}{dx_2} = \frac{\partial y_1}{\partial y_2}\frac{dy_2}{dx_2}$$
$$\frac{dy_2}{dx_2} = \frac{\partial y_2}{\partial x_2} + \frac{\partial y_2}{\partial y_1}\frac{dy_1}{dx_2}$$

위의 식들을 정리하면 다음과 같은 GSE가 만들어진다.

$$\begin{bmatrix} I & -\dfrac{\partial y_1}{\partial y_2} \\ -\dfrac{\partial y_2}{\partial y_1} & I \end{bmatrix} \begin{Bmatrix} \dfrac{dy_1}{dx_1} \\ \dfrac{dy_2}{dx_1} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial y_1}{\partial x_1} \\ 0 \end{Bmatrix}$$
$$\begin{bmatrix} I & -\dfrac{\partial y_1}{\partial y_2} \\ -\dfrac{\partial y_2}{\partial y_1} & I \end{bmatrix} \begin{Bmatrix} \dfrac{dy_1}{dx_2} \\ \dfrac{dy_2}{dx_2} \end{Bmatrix} = \begin{Bmatrix} 0 \\ \dfrac{\partial y_2}{\partial x_2} \end{Bmatrix}$$

이러한 GSE는 편미분값들로부터 전미분값을 계산하는 연립방정식이다. GSE를 풀어서 전미분값을 구하면 각 분야의 해석들은 다음과 같이 선형적인 양함수로 근사된다.

$$y_1 = y_1(x^0) + \frac{dy_1}{dx}(x - x^0)$$
$$y_2 = y_2(x^0) + \frac{dy_2}{dx}(x - x^0)$$

이러한 근사식은 최적화과정에서 해석의 대용으로 사용되어 해석에 드는 비용을 감소시킬 수 있다.

GSE에 기반한 MDF기법의 전체적인 과정은 <그림 4>와 같다. 먼저 초기의 값에 대하여 다분야통합해석을 수행하고, 편미분값들을 구하기 위해 분야별로 민감도 해석을 동시에 수행한다. 민감도 해석이 완료되면 GSE를 구성하고 이것을 풀어서 전미분값들을 구한다. 이제 각 해석의 상태방정식들이 선형적으로 근사되고 최적화를 통해 설계변수가 갱신된다. 상태방정식이 선형적으로 근사되었으므로 적절한 이동범위제한(move limit)내에서 최적화를 수행해야 한다. 이상의 과정을 수렴할 때까지 반복하여 최종적인 해를 구한다.

3.2 IDF(Individual Discipline Feasible)

IDF는 연성변수와 적합성 제약조건을 도입하여 각 분야의 해석을 독립적으로 수행하려는 방법이다. 이렇게 함으로써 다분야통합해석의 반복계산을 피할 수 있고 각 분야의 해석이 분산환경에서

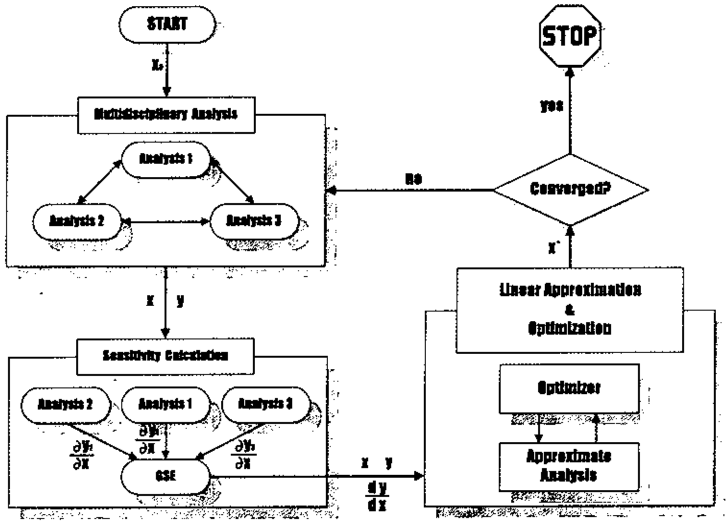

그림 4. GSE에 기반한 MDF의 흐름도

동시에 수행될 수 있다. IDF의 경우, 최적점을 찾아가면서 거쳐가는 점들이 다분야통합해석을 만족하지는 않는다. 즉, 최적화가 완료되는 시점에서 설계변수뿐 아니라 적합성 제약조건을 만족하는 연성변수를 찾음으로써 비로소 다분야통합해석을 만족하게 되는 것이다. 다만, 최적화가 진행되는 과정에서 각 분야마다 해석이 수행되고 있으므로, 개별 분야의 해석은 만족하고 있다고 할 수 있다. IDF 즉 '개별 분야 만족' 이라는 것은 이러한 특성을 지칭하는 것이다. Balling 등(1996)은 이러한 기법을 "Single-SAND-NAND"라고 칭하였는데, 이러한 이름의 첫 번째 부분인 "Single"은 단일단계 최적화를 의미하고, 두 번째 부분의 "SAND"는 연성변수와 적합성 제약조건을 도입하여 분야간 해석의 연성을 최적화를 통해 풀어냄을 의미하고, 세 번째 부분의 "NAND"는 각 분야의 해석이 최적화 모듈에 중첩되었음을 의미한다.

IDF기법의 특징을 살펴보자. 가장 큰 특징은 다분야통합해석이 사라지고, 각 분야의 해석이 자율적으로 이루어진다는 것이다. 분야간 해석의 연성이 사라짐으로 각 분야의 해석이 병렬적으로 수행될 수 있다. MDF와 마찬가지로 각 분야 고유의 해석프로그램들을 최대한 활용할 수 있다. 다분야통합해석이 사라지는 대신에 연성변수가 설계변수로 추가되었다. 또한 적합성 제약조건이라는 등가제약조건들이 추가되었다. 풀려고 하는 문제가 해석간의 연성 폭이 적고, 등가제약조건을 잘 다룰 수 있는 최적화 소프트웨어가 있다면 IDF는 MDF의 효율적인 대안이 될 것이다.

3.3 AAO(All At Once)

AAO는 분야간 해석의 연성을 해결하는 것을 최적화모듈에 맡김과 동시에 각 분야 해석의 상태방정식을 만족하는 해를 찾는 것도 최적화모듈이 담당토록 하는 방법이다. 분야간 해석의 연성을 피하기 위해 최적화 문제에 연성변수와 적합성 제약조건이 추가되고, 각 분야의 해석의 상태방정식을 만족하는 해를 찾기 위해 상태변수와 상태방정식의 residual이 0이 되어야 한다는 제약조건이 추가된다. 이러한 방식으로 최적화 모듈에 다분야통합해석이 중첩되는 것과 각 분야의 해석이 중첩되는 것을 피할 수 있다. AAO는 최적점을 찾아가면서 거

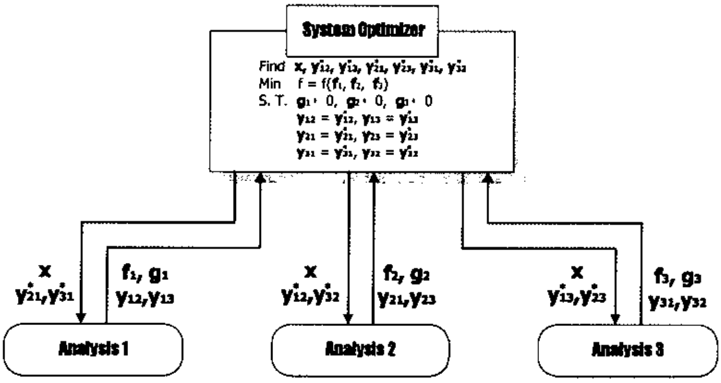

그림 5. IDF(Individual Discipline Feasible) 기법

쳐가는 점들이 다분야통합해석을 만족하지 않을 뿐더러, 각 분야의 해석도 만족하지 않는다. 다만, 최적화가 이루어지는 시점에서 설계변수와 함께 적합성 제약조건을 만족하는 연성변수와 상태방정식을 만족하는 상태변수를 찾음으로써 최적점은 각 분야의 해석뿐 아니라 다분야통합해석을 만족하게 되는 것이다. AAO 즉, '모두 한꺼번에' 가 의미하는 것은 최적화가 종료되는 시점에서 설계변수, 연성변수 및 상태변수가 모두 한꺼번에 결정되는 특성을 나타내는 것이다. Balling 등(1996)은 이러한 기법을 "Single-SAND-SAND"라 칭하였는데, 이러한 이름의 첫 번째 부분인 "Single"은 단일단계 최적화를 의미하고, 두 번째 부분의 "SAND"는 연성변수와 적합성 제약조건을 도입하여 분야간 해석의 연성을 최적화를 통해 풀어냄을 의미하고, 세 번째 부분의 "SAND"는 상태변수와 상태방정식의 residual이 0이 되어야 한다는 제약조건을 도입하여 각 분야의 해석 결과인 상태변수를 최적화 모듈에서 구하게 됨을 의미한다. AAO에서의 각 분야 해석 모듈의 기능은 이전의 해석 모듈의 기능과는 구별되는데, 이전의 해석모듈에서는 상태방정식을 만족하는 상태변수를 찾아서 목적함수와 제약함수의 값을 구했지만, 여기서는 최적화 모듈에서 할당하는 상태변수에 대하여 목적함수와 제약함수 및 상태방정식의 residual을 계산하는 단순한 "function evaluation"의 기능만을 수행한다.

AAO기법의 특징을 살펴보자. IDF와 마찬가지로 AAO에서는 다분야통합해석이 필요치 않다. 또한 각 분야의 해석도 그 분야의 상태방정식을 풀 필요가 없다. 최적화모듈에서 설계변수 뿐 아니라 다분야통합해석을 만족하는 연성변수와 각 분야 해석의 상태방정식을 만족하는 상태변수를 구하게 된다. 이러한 방식은 각 분야의 해석프로그램들을 black box로 이용하는 것이 아니라 최적화프로그램과 code 수준에서 융합됨을 의미한다. 각 해석프로그램들과 최적화프로그램의 code 수준에서 융합은 가장 효율적인 MDO 기법을 창출할 수 있지만, code 수준의 융합에 드는 비용을 생각할 때 그다지 현실적이지 못하다. 또한 각 분야 고유의 해석방식을 하나의 최적화 기법으로 정식화하는 것이 불가능할 수도 있다. 각 분야의 해석이 간단하고 최적화모듈과 code 수준에서의 융합이 용이한 경우, AAO는 다른 MDO기법들보다 가장 효율적인 방법이 될 것이다.

4. 다단계 MDO기법

다단계 MDO기법은 MDO기법에 다단계 최적화기법을 적용한 것을 말하는 것으로 각 분야가 해석뿐 아니라 설계변수의 결정까지 담당하는 것을

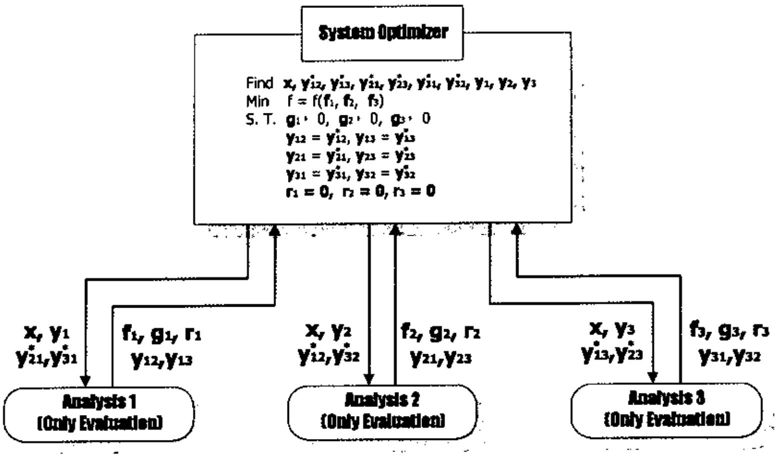

그림 6. AAO(All At Once) 기법

의미한다. 즉, 전체의 설계변수가 하나의 최적화모듈에서 결정되는 것이 아니라, 각각의 분야가 자신의 최적화모듈을 가지고서 자신에게 할당된 설계변수들을 자체적으로 결정하는 것이다. 이러한 결정권의 분산은 각각의 결정들을 전체로 융합하기 위한 조정과정을 필요로 하는데 이러한 것은 시스템단계에서의 최적화로 표현된다. CSSO기법은 해석의 연성을 고려하면서 설계변수의 결정에 있어 다단계 최적화 기법을 도입하여 각 분야마다 자신의 설계변수를 결정하는 기법이다. 즉, 분야간의 연성을 고려한 다분야통합해석이 수행되고 난 후, 설계변수의 결정이 하나의 시스템 최적화모듈에서 일어나는 것이 아니라 각각의 분야에 분산되는 것이다. CO기법은 해석도 비연성화되고, 의사결정도 분산된 기법이다. 따라서 CO기법은 다른 어떤 기법들보다 각 분야의 자율성을 강조된 기법이라 할 수 있다.

4.1 CSSO(Concurrent SubSpace Optimization)

OLD와 같은 기존의 다단계 최적화 기법들은 수직적 계층구조에만 적용될 수 있었고, 이러한 한계를 극복하기 위해 Sobieski는 비수직적 계층구조를 갖는 문제에 적용할 수 있는 다단계 최적화 기법으로 CSSO를 고안하였다(Sobieski, 1988). 이 방법은 그 이름이 말해주듯이 분해된 부공간(subspace, 하위시스템과 같은 개념)에서 최적화가 동시에 이루어지고, 전체의 조정이 시스템 단계의 최적화를 통해 달성되는 기법이다.

CSSO는 GSE에 기반한 MDF 방법에 다단계 최적화 기법을 접목한 것으로 다분야통합해석과 GSE를 푸는 과정까지는 GSE에 기반한 MDF 방법과 동일하다. 이 후의 최적화 과정이 최적화모듈에서 모든 설계변수를 결정하는 단일단계 최적화와는 다르게 각각의 하위 공간의 최적화모듈에서 각각의 설계변수를 동시에 결정하고 이후에 조정과정을 거치는 다단계 최적화 기법을 적용한다는 것이 중요한 차이점이라고 할 수 있다. 다단계 최적화 기법을 적용하기 위해서 설계변수의 분해가 필요하다. CSSO에서는 전체시스템의 설계변수가 여러 분야의 설계변수로 나누어질 때, 각 부공간의 설계변수들은 서로 공통된 부분이 없도록 분해된다. 일반적으로 하나의 설계변수가 여러 분야에 공유될 수 있지만 CSSO에서는 이러한 변수의 결정을 하나의 분야에서만 담당하고 다른 분야에서는 그 결정에 따르도록 하는 것이다. Sobieski는 과

거의 설계경험이나 직관을 바탕으로 설계변수를 분해할 것을 제안하였다. 그러나 설계변수의 할당이 적절치 못한 경우, 수렴에 나쁜 영향을 끼칠 수 있으므로 Bloebaum은 설계변수의 할당에 영향계수(effectiveness coefficient)를 도입하여 이동범위 제한 전략을 강화할 것을 제안하였다(Bloebaum, 1991).

이제 CSSO 기법의 전체적인 흐름을 간략히 소개해 보자. 초기의 설계변수에 대하여 다분야통합 해석을 수행하고, GSE를 구성하기 위한 민감도 해석이 분야별로 독립적으로 동시에 수행된다. 이후 GSE를 풀어서 전미분값들을 구하는 과정까지는 GSE에 기반한 MDF방법과 동일하다. 이후의 최적화 과정에서 단일단계 최적화 기법을 사용하는 대신 다단계 최적화 기법을 사용한다는 것이 차이점이다. 다단계 최적화 기법을 적용하기 위해 설계공간의 분해가 필요한데, 직관적으로 혹은 영향계수를 바탕으로 설계변수가 각각의 부공간으로 분해된다. 이렇게 분해된 설계공간은 일반적으로 비수직적 계층구조를 갖게되어 OLD에서 사용되는 다단계 최적화 기법과는 다른 방식을 적용해야 한다.

먼저 각 부공간의 목적함수를 살펴보자. 각 부공간의 목적함수는 시스템의 목적함수와 동일하지만, 각 부공간의 설계변수만이 변수로 사용되고, 다른 부공간의 설계변수들은 그 변수들의 현재 값으로 고정된다.

부공간 1의 목적함수 : $f = f(x^1, x^{20}, x^{30}) = f(x^1)$

부공간 2의 목적함수 : $f = f(x^{10}, x^2, x^{30}) = f(x^2)$

부공간 3의 목적함수 : $f = f(x^{10}, x^{20}, x^3) = f(x^3)$

각각의 부공간에서 만족해야 할 제약조건들은 자신의 부공간의 제약조건만이 아니다. 부공간들 사이의 연성이 존재하므로 어떤 부공간의 설계변수 변화에 의해 다른 부공간의 제약조건이 위배되기도 한다. 따라서, 각 부공간의 최적화 과정에서 다른 부공간의 제약조건도 고려되어야 한다. GSE를 사용한 MDF 기법과 마찬가지로 각각의 부공간의 제약조건들은 KS함수를 사용하여 하나의 누적제약함수(C_i)로 표현된다. 각 부공간의 설계변수 변화에 의한 제약함수값의 변화는 그 부공간의 해석을 통해 계산될 수 있다. 다른 부공간의 제약함수들은 GSE를 통해 구한 전미분값을 사용하여 근사된다.

$$C_j = C_j^0 + \frac{dC_j}{dx_i} \Delta x_i, \qquad \text{for } i \neq j$$

이제 각각의 부공간은 자기자신의 목적함수과 자신의 제약조건 및 다른 부공간의 근사된 제약조건 하에서 동시에 최적화를 수행한다. 다른 부공간의 제약조건들이 선형적으로 근사되므로, 설계변수에 이동범위제한(move limit)이 부과된다.

CSSO의 조정과정은 제약조건의 위반의 책임을 각 부공간에 분산시키는 책임계수(responsibility coefficient)와 어떤 부공간의 목적함수 감소에 의해 발생하는 제약조건의 위반을 다른 부공간에서 보상하기 위한 교환계수(trade-off coefficient)를 통해 이루어진다. 이러한 조정과정의 도입으로 각 부공간은 다음과 같은 최적화를 수행한다. 즉, 제약조건의 위배가 있으면 그 위배량에 대한 자신의 책임부분을 감소시키면서 목적함수의 증가를 최소화하고, 제약조건이 이미 만족되었다면, 목적함수를 최소화하도록 한다. 조정변수인 책임계수와 교환계수를 결정하는 과정은 조정최적화과정(coordination optimization procedure)으로 불리는 시스템단계의 최적화과정이다. 조정최적화과정을 위해서 각각의 부공간의 최적화가 끝나면, 조정변수들에 대한 최적민감도해석(optimum sensitivity analysis)을 수행한다. 최적민감도해석을 통해서 시스템목적함수가 조정변수의 함수로 근사되면, 조정변수들을 결정하기 위한 조정최적화가 수행된다. 이러한 일련의 과정들이 수렴될 때까지 반복된다.

4.2 수정된 CSSO 기법

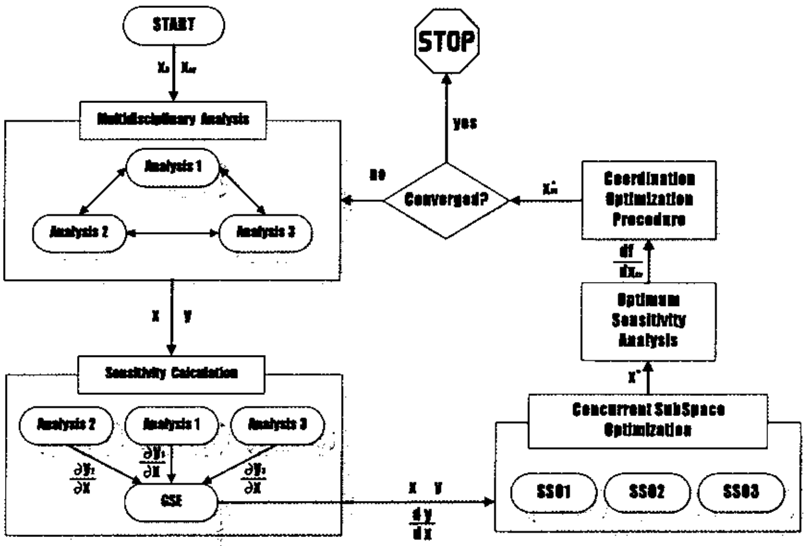

그림 7. CSSO기법의 흐름도

Renaud는 수정된 CSSO기법을 개발하였다(Renaud, 1993). 이러한 방법의 장점은 CSSO에서 사용되는 직관적인 조정과정 대신에 응답면에 기반한 조정과정을 사용했다는 점이다. 또한 부공간들 사이의 설계변수 공유도 가능하다.

수정된 CSSO 기법의 전체적인 흐름을 살펴보자. 먼저 주어진 설계변수에 대한 다분야통합해석이 이루어진다. 다분야통합해석이 완료되면 각각의 분야별로 편미분을 계산하기 위한 민감도 해석을 수행하고, GSE를 구성하여 전미분값들을 구한다. 이후 설계변수가 각 부공간으로 분해되어 할당되고, 각 부공간별로 할당된 설계변수들을 결정하기 위한 최적화 작업이 수행된다. 이때, 각각의 부공간이 만족시켜야 할 제약조건들은 그 분야 고유의 제약조건과 선형으로 근사된 다른 분야의 제약조건들이다. 부공간의 최적화는 설계변수의 이동범위 제한내에서 각각 동시에 수행된다. 각각의 설계과정에서 생성된 자료들은 전체 설계 데이터베이스에 축적된다. 부공간의 최적화가 끝나면 데이터베이스에 축적된 자료를 바탕으로 현재의 설계점 근처에서 시스템 전체의 근사 문제가 형성된다. 즉, 데이터베이스를 바탕으로 목적함수와 제약함수들에 대한 이차의 응답면이 만들어진다. 이러한 근사문제에 대한 최적화를 통해 전체시스템의 조정이 이루어지는 것이다. 전체 시스템의 근사 최적화도 설계변수에 대한 이동범위의 제한내에서 수행된다. 이러한 전체적인 과정이 수렴할 때까지 반복된다.

수정된 CSSO기법은 다음과 같은 단점들이 지적되고 있다. 첫째, 부공간의 최적화가 단지 시스템 전체의 근사를 위한 자료를 생성하기 위해 수행된다. 둘째, 거대한 설계문제의 경우, 자료를 저장하고 전체시스템의 근사를 형성하는데 많은 비용이 들 수 있다. 셋째, 전체 시스템 근사의 정확도가 각 부공간에서 사용되는 이동범위 제한 전략에 크게 좌우된다.

4.3 CO(Collaborative Optimization)

CO는 MDO기법들 중에서 가장 최근에 개발된 것으로 Kroo와 Braun에 의해 제안되었다. CO에서는 각 분야의 자율성 증진에 초점을 맞추고 있다. CO에서는 연성변수를 도입하여 다분야통합해

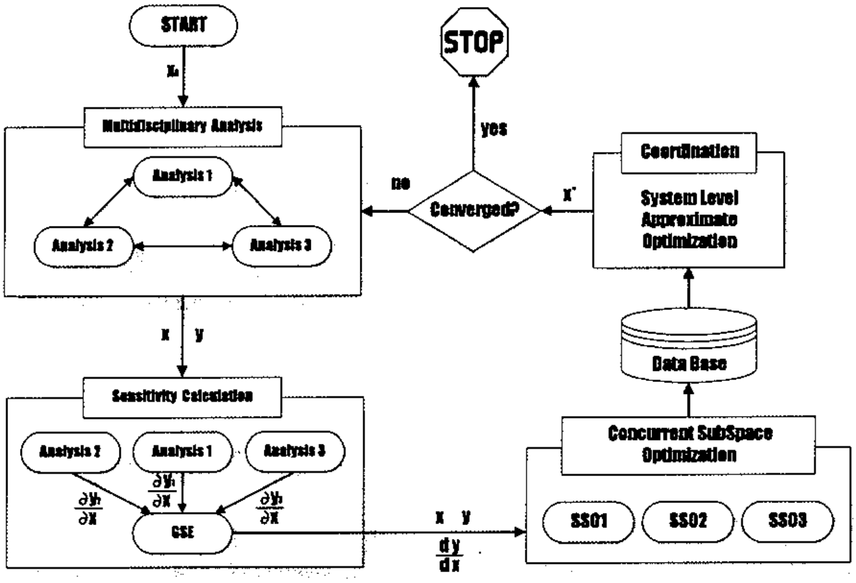

그림 8. 수정된 CSSO기법

석을 피하는 SAND기법을 적용할 뿐 아니라, 각 분야에서 설계변수를 결정하는 다단계 최적화 기법을 적용하고 있다. 또한 설계공간의 분해시 각 부공간들 사이의 변수 공유도 가능하다. 조정기법으로는 부공간에서 그 분야 고유의 제약조건을 만족하면서 전체시스템과의 불일치를 최소로 하는 방식을 채택하여, 각 부공간에서 생성되는 설계점들이 항상 그 분야 고유의 제약조건을 만족하는 "Discipline Constraints Feasible"의 특성을 갖는다. CSSO와의 가장 큰 차이점은 한 분야의 최적화를 수행할 때 다른 분야의 제약조건은 고려하지 않고 그 분야 고유의 제약조건만을 만족하는 설계점들을 찾는다는 점이다. <그림 9>는 CO의 기본적인 개념을 나타낸 것이다.

<그림 9>에서 보듯이 각 부공간 최적화모듈의 목적은 분야간의 공유 변수에 대하여 다른 부공간들과 의견을 일치시키도록 하는 것이다. 시스템의 최적화모듈은 전체의 목적함수를 최소화시키면서, 부공간들의 의견들을 조정하는 것을 담당한다. 각 부공간의 목적이 다른 그룹과 의견의 일치를 달성하는 것이고 시스템의 최적화모듈이 이러한 프로세스를 조정하는 것이므로, CO 즉 "협동 최적화"라 불리게 되었다.

CO의 시스템 단계에서의 최적화과정은 각 분야의 연성과 관련된 변수들을 결정하는 것이다. 즉, 공유설계변수, 조정변수 및 연성변수가 시스템 단계의 설계변수이다. 공유설계변수와 조정변수는 각 하위시스템들에도 존재하므로, 그들 사이의 구분을 위해서 시스템 단계의 공유설계변수 및 조정변수는 $(\cdot)^s$의 기호를 사용하자. 시스템 단계의 최적화과정에서 제약조건으로 사용되는 것은 각 분야와 시스템의 연성과 관련된 변수들에 대한 일관성 혹은 적합성 제약조건들이다. 시스템 단계에서의 설계변수를 제외한 나머지변수들은 시스템 단계의 최적화과정에서 고정된 파라미터로 취급한다. 따라서, 시스템 단계의 최적화과정은 하위시스템의 최적화에서 생성된 값들을 고정된 파라미터로 취급하여, 시스템 목적함수를 최소로 하고 각 하위시스템과의 일관성 혹은 적합성 제약조건을 만족하는 시스템의 공유설계변수, 조정변수 및 연성변

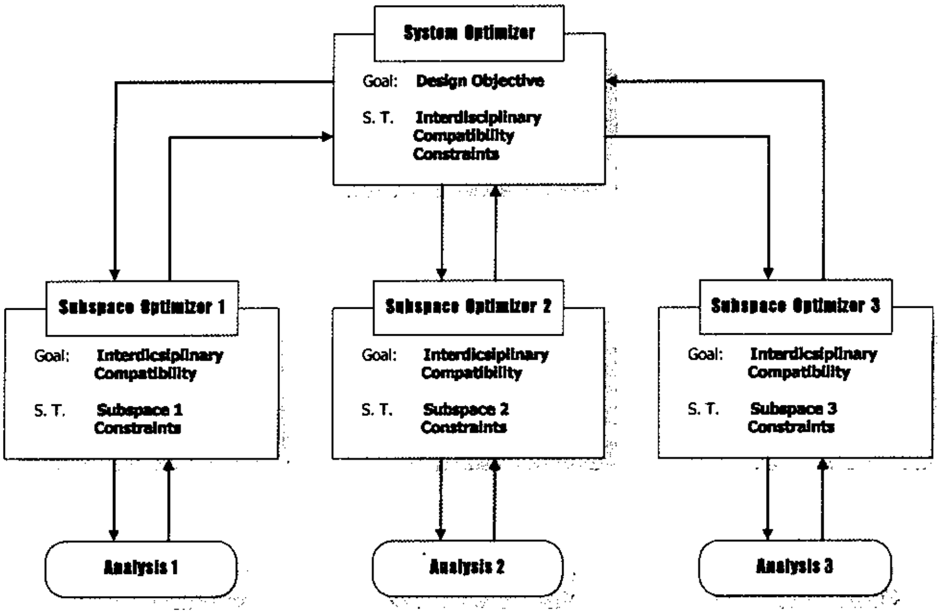

그림 9. CO의 기본구조

수를 찾는 것이다.

시스템 단계에 대한 최적화 문제는 다음과 같이 기술된다.

Find: $(x_{sh})^s$, $(x_{cv})^s$, y_{ij}^*

Minimize: $f((x_{sh})^s, (x_{cv})^s, y_{ij}^*)$

Satisfy: $d_1^s = 0$, $d_2^s = 0$, $d_3^s = 0$

여기서, 시스템 단계의 불일치함수는 다음과 같이 정의된다.

$$d_i^s = d_i^s\,((x_{sh})_i^s, (x_{cv})_i^s, y_{ij}^*)$$
$$= \|(x_{sh})_i^s - (x_{sh})_i\|^2 + \|(x_{cv})_i^s - (x_{cv})_i\|^2 + \|y_{ij}^* - y_{ij}\|^2$$

하위시스템 단계에서의 최적화는 각 분야 고유의 제약조건을 만족하면서 시스템 단계의 변수들과의 불일치를 최소로하는 각 분야의 설계변수 및 조정변수를 찾는 것이다.

하위시스템 단계에서의 최적화 문제는 부재 i에 대하여 다음과 같이 기술된다.

Find: x_i, $(x_{cv})_i$

Minimize:

$$d_i = d_i\,((x_{sh})_i, (x_{cv})_i)$$
$$= \|(x_{sh})_i - (x_{sh})_i^s\|^2 + \|(x_{cv})_i - (x_{cv})_i^s\|^2 + \|y_{ij} - y_{ij}^*\|^2$$

Satisfy: $g_i \geq 0$

CO는 다른 어떤 기법보다 각 분야의 자율성이 강조된 기법이다. 즉, 연성변수를 도입하여 해석들 사이의 연성을 해결하였고, 다단계 최적화기법을 도입하여 각 분야에 설계변수의 결정권한을 부여하였다. 또한 분야들 사이의 설계변수공유도 가능하다. 이러한 각 분야의 자율성을 강조하는 대가로 다른 기법들에 비해 보다 많은 보조설계변수를 도입해야 한다. 따라서 분야간의 연성의 폭이 큰 문제를 다루는 경우, CO는 효율적이지 못하다.

5. 맺음말

이상에서 MDO 기법들을 소개하고 각 기법의 특성을 이론적으로 설명하였다. 지금까지 MDO 기법들이 다양한 설계문제에 적용되어왔고, 기존의

설계방법들과 비교되어 왔다. 그러나, MDO 기법들 간의 비교는 Balling 등(1997)과 Alexandrov 등(1998)에 의해 시도되고 있지만 아직 시작단계이다. 따라서 다양한 예제를 통해 MDO 기법들을 비교하여 그들의 성능을 검증하고, 그 결과를 바탕으로 MDO 기법 선택을 위한 가이드라인을 생성할 필요가 있다. 이와 더불어 MDO 기법을 적용하려는 설계 대상의 도메인 특성을 파악하려는 노력이 요구된다.

참고문헌

[1] Alexandrov, N. M., and Kodiyalam, S., "Initial Results of an MDO Method Evaluation Study," AIAA Paper 98-4884, 1998.

[2] Balling, R. J., and Sobieszcznski -Sobieski, J., "Optimization of Coupled Systems: A Critical Overview of Approaches," AIAA Journal, Vol. 34, No. 1, pp. 6-17, 1996.

[3] Balling, R. J., and Wilkinson, C. A., "Execution of Multidisciplinary Design Optimization Approaches on Common Test Problems," AIAA Journal, Vol. 35, No. 1, pp. 178-186, 1997.

[4] Balling, R. J., and Gale, D. L., "Collaborative Optimization of Systems Involving Discrete Design at the Discipline Level," ASME Journal of Mechanical Design, Vol. 120, No. 1, pp. 32-39, 1998.

[5] Bloebaum, C. L., Formal and Heuristic System Decomposition Methods in Multidisciplinary Synthesis, Ph.D. Dissertation, University of Florida, 1991.

[6] Braun R. D., Collaborative Optimization: An Architecture for Large-Scale Distributed Design, Ph.D. Dissertation, Stanford University, 1996.

[7] Cramer, E. j., Dennis J. E., Jr., Frank, P. D., Lewis, R. M., and Shubin, G. R., " Problem Formulation for Multidisciplinary Optimization," Center for Research on Parallel Computation, Rice Univ., Rept. CRPC-TR93334, Houston, 1993.

[8] Hajela, P., Bloebaum, C. L. and Sobieszcznski-Sobieski, J., "Application of Global Sensitivity Equations in Multidisciplinary Aircraft Synthesis," Journal of Aircraft, Vol. 27, No. 12, pp. 1002-1010, 1990.

[9] Renaud, J. E., and Gabriele, G. A., "Improved Coordination in Nonhierarchic System Optimization," AIAA Journal, Vol. 31, No. 12, pp. 2367-2373, 1993.

[10] Renaud, J. E., and Gabriele, G. A., "Approximation in Nonhierarchic System Optimization," AIAA Journal, Vol. 32, No. 1, pp. 198-205, 1994.

[11] Sobieszcznski-Sobieski, J., "Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems," Recent Advances in Multidisciplinary Analaysis and Optimization, NASA CP-3301, Pt. 1, 1988.

[12] Sobieszcznski-Sobieski, J., "Sensitivity of Complex, Internally Coupled Systems," AIAA Journal, Vol. 28, No. 1, pp. 153-160, 1990.

[13] Sobieszcznski-Sobieski, J., "Sensitivity Analysis and Multidisciplinary Optimization for Aircraft Design : Recent Advances and Results," Journal of Aircraft, Vol. 27, No. 12, pp. 993-1001, 1990.

[14] Sobieszcznski-Sobieski, J., Bolebaum, C. L., and Hajela, P., "Sensitivity of Control-Augmented Structure Obtained by a System

Decomposition Method," AIAA Journal, Vol. 29, No. 2, pp. 264-270, 1991.

[15] Sobieszcznski-Sobieski, J., and Haftka, R. T., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," AIAA Paper 96-0711, 1996.

[16] Tappeta, R. V., An Investigation of Alternative Problem Formulations for Multidisciplinary Optimization, Master' s Thesis, University of Notre Dame, 1996.

[17] Wujek, B. A., Renaud, J. E., and Batill, S. M., "Concurrent Subspace Optimization Using Design Variable Sharing in a Distributed Computing Environment," Concurrent Engineering: Research and Applications, Vol. 4, No. 4, pp. 361-377, 1996.

[18] 노명일, A Study on the Multidisciplinary Design Optimization(MDO) using Collaborative Optimization Method, 공학석사학위논문, 서울대학교, 2000.

[19] 박창규, 자동미분에 의한 다분야통합설계문제의 분리시스템동시최적화기법 개선, 공학석사학위논문, 연세대학교, 1999.

양 영 순

- 1951년 1월 3일생
- 1979년 서울대학교 공학박사
- 현 재: 서울대학교 조선해양공학과 교수
- 관심분야: 선체구조 신뢰성 해석 및 인공지능분야
- 전 화: 02-880-7330
- E-mail: ysyang@gong.snu.ac.kr

정 현 승

- 1972년 6월 12일생
- 1997년 서울대 조선해양공학과 석사
- 현재 서울대 조선해양공학과 박사과정
- 관심분야: MDO, 구조충돌해석
- 전 화: 02-880-7338
- E-mail: jhs@insdel.snu.ac.kr