

다분야 항공기 통합최적설계 기술동향

이수창, 안종기
국방과학연구소

1. 개요

다분야 통합최적설계(MDO, Multidisciplinary Design Optimization)는 여러 분야의 공학적 현상을 하나의 설계시스템 안에서 다루어야 하는 분야로서, 전체 시스템의 최적화가 궁극적인 목적인 만큼 이를 효과적으로 구현하기 위한 설계구조에 대한 연구가 중요한 주제이다.

MDO의 설계구조는 MDO프레임워크 요구조건[1]에 기술된 바와 같이 기존의 최적설계기법을 이용할 수 있어야 하며, 기존의 설계해석코드 및 상용코드를 사용할 수 있어야 하고, 해석/최적화과정이 병렬적으로 수행될 수 있어야 하는 등, 전체적으로 유연성과 확장성, 효율성을 지향하고 있다. 효과적인 설계구조를 위한 접근 방식에는 문제를 어떻게 구성(formulation)하는가에 관한 연구와 이렇게 구성된 문제를 어떠한 최적화기법(algorithm)에 의해 풀어나가느냐에 대한 연구[2]로 구분된다. 이 글은 전자의 문제구성 방식을 주된 내용으로 기술하고, 이러한 문제구성 방식이 후자의 최적 알고리즘에 미치는 영향을 기술하고자 한다.

이 글에서 기술하고자 하는 MDO의 설계구조는 다음과 같다. 첫째, 서로 얽혀 있는 여러 분야의 해석부를 하나의 통합된 해석부에서 처리하는 초기단계의 통합 해석 구조(Integrated Analysis Architecture). 둘째, 각 해석부들을 분리(Decomposition)시켜 처리하는 분산해석 구조(Distributed Analysis Architecture). 마지막으로 해석부와 최적화부를 통합된 형태로 부체계에 보다 많은 설계권한을 부여한 분산설계 구조(Distributed Design Architecture)가 그것이다

2. 통합해석 구조(Integrated Analysis Architecture)

통합해석 구조[3]는 여러 학문분야의 해석부를 하나의 통합된 해석부 속에 포함시키고 이를 최적화부와 연결시킨 기본적인 MDO의 형태이다. 일반적으로 MDF(Multidisciplinary Feasible)[2], Standard Approach[4]구조라 불리기도 한다. 통합해석 구조는 각 해석부들 중

에서 적어도 두 해석부 이상에서 연관된 변수가 있을 경우, 이 변수들은 통합 해석부 내에서 반복, 수렴과정을 거쳐 상호 동일성(Compatibility)을 확보한 다음, 최적화부로 변수를 넘겨주는 방식이다. 항공기 날개설계의 경우[12], 공력 해석부에서 계산된 주익의 형상은 구조 해석부로 넘어가서 구조해석에 이용되며, 역으로 여기에서 계산된 주익의 형상이 다시 공력 해석부로 넘어가는 과정을 반복하여 두 해석부의 주익 형상이 동일해 질 때까지 내부적인 반복과정(통상 one-point iteration)을 거치는 방식이다. 이러한 구조는 해석부의 수가 많아지거나 상호 연관된 변수가 많아질수록 계산량이 급격히 증가하여 대형시스템의 설계에는 적용하기 어렵고, 해석부의 내용을 변경하기가 쉽지 않다는 단점을 갖고 있다

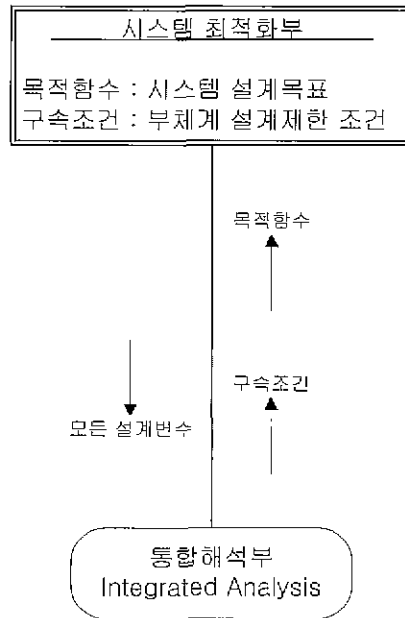


그림 1. 통합해석 구조의 구성.

통합 해석구조를 가지는 설계시스템은 1960년대 말, 미국에서 개발된 항공기 설계프로그램인 SYNAC을 시

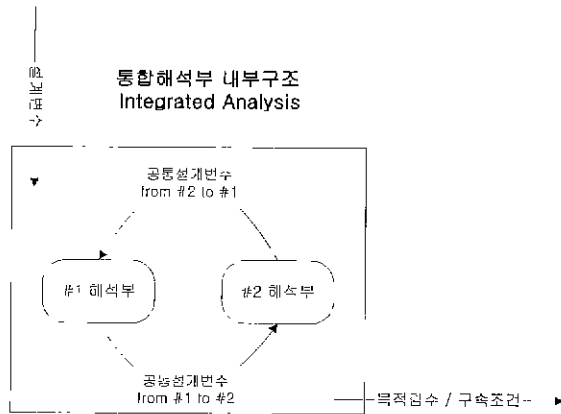


그림 2. 통합해석부의 내부구조.

작으로 ACSYNT(AirCraft SYNThesis), IPAD(Integrated Programs for Aerospace-Vehicle Design)[13], FLOPS, PASS(Program for Aircraft Synthesis Studies)[14]등이 있다. 이들 중, 항공기의 전 수명주기(Life Cycle)를 고려하는 시스템 설계를 목표로 한 IPAD와 같은 경우에는 설계 구조의 한계로 만족할만한 성과를 달성하지 못한 것으로 알려져 있다. 그러나 ACSYNT는 대부분의 해석부가 간단한 경험식을 기반으로 짜여져 있고, 객체지향 기술 등을 적용함으로써 항공기의 개념설계에 유용하게 사용되고 있다. 특히 PASS나 서울대학교에서 개발한 AAD(Advanced Aircraft Design)[15]같은 시스템은 유사순차적(QPM, Quasi Procedural Method)기법[5]과 같은 새로운 데이터베이스 관리(Database Management)기술을 적용함으로써 통합해석 구조의 대안으로 제시되고 있다.

3. 분산해석 구조(Distributed Analysis Architecture)

시스템이 대형화하고 복잡해짐에 따라 기존 통합해석구조의 한계를 인식하고 이를 극복하기 위하여 연구된 것이 분산해석 구조이다. 시스템을 학제(Disciplinary)나 설계변수사이의 연관성을 감안하여 여러 개의 하부 해석부로 분리(Decomposition)시킨 다음, 이를 최적화부

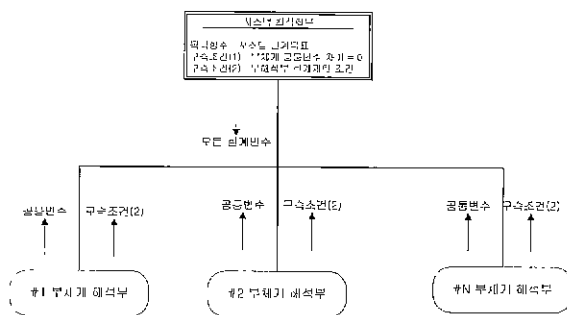


그림 3. 분산해석 구조의 구성.

와 연결시킨 구조로서 전공에 따라 공력, 구조, 제어 등으로 인적자원을 분류한 실제 산업현장의 설계조직 구성과 유사한 면이 있다.

분산해석 구조는 IDF(Individual Discipline Feasible)[2,6], All-at-once Optimization[4], OBD(Optimization Based Decomposition) 등으로 불려지고 있다.

통합해석 구조와 비교되는 점은 시스템 설계자가 해석부들 사이의 상호 관련성을 이해하고, 해석부 사이의 변수들을 직접 연결하는 작업이 크게 줄었다는 점이다. 시스템 설계자는 각 해석부의 입력력 관계만을 파악하고, 해석부 사이에 동일한 의미를 가지는 공통 설계변수(Interdisciplinary Variables)는 구속조건(1)에서 하나의 이름으로 치환하는 작업만을 하면 되기 때문이다.

분산해석구조는 각각의 해석부를 분산 전산환경에서 처리할 수 있기 때문에 계산시간을 줄일 수 있다는 장점이 있다. 날개설계를 예로 들면, 통합해석구조에서는 구조해석부에서 날개의 유한요소(Finite Element)에 분포하는 구조하중을 계산하기 위해서는 공력해석부로 부터 날개 위의 압력분포를 계산해줄 때까지 기다려야 한다. 그러나, 분산해석구조에서는 각 해석부는 시스템이 내려 준 변수들을 가지고 동시에 계산을 수행할 수 있다. 물론, 이 경우 시스템의 최적화부는 각 해석부에서 올라온 변수들 중 동일한 의미를 가지는 변수들끼리 묶어 이들의 차이를 등식 구속조건(Equality Constraints)으로 설정함으로써 각 해석부들끼리의 동일성을 만족시키는 역할을 한다.

그러나 통합해석구조와 마찬가지로 분산해석구조의 해석부는 여전히 단순한 해석만을 수행함으로써 설계 변수나 제한조건의 수가 많은 시스템, 또는 해석부가 서로 밀접하게 연결되어 있는 시스템의 설계는 최적화부에 큰 부담이 될 수 있다. 그럼에도 불구하고 분산해석구조는 현재로서는 가장 보편화되고, 신뢰성이 검증된 MDO구조로 인정받고 있다. 대표적인 MDO프레임워크는 Rockwell사의 Design Sheet[16], GenIE, NASA의 FIDO(Framework for Interdisciplinary Design Optimization)와 General Electric사의 자체 설계용 프로그램에서 상용으로 발전한 Engineous사의 iSIGHT[7,17]가 있다. 특히 iSIGHT는 어떠한 원시코드나 실행코드도 쉽게 연결이 가능한 뛰어난 확장성을 가지고 있다. GenIE의 각 해석부는 유사순차적 구조를 가지고 있으며, 병렬처리 프로그램인 PVM(Parallel Virtual Machine)을 이용하여 연산을 수행하는 구조이다.

4. 분산 설계구조(Distributed Design Architecture)

앞서의 분산해석구조에서 각 해석부는 단순히 계산만을 수행하고 전체 시스템의 최적화는 하나의 최적화부에서 수행하는 방식이다. 이러한 방식은 한명의 설계자(Chief Design Engineer)가 전체 시스템의 설계하는 고

전적인 항공기설계 방식과 유사한데, 시스템이 복잡해지고 대형화한 현대의 항공기 설계에는 김당할 수 없는 방식이다. 이러한 문제를 해결하고자 대두된 것이 분산설계 구조[3], 다층최적화(Multilevel Optimization)[2,6]라 부르는 구조이다.

분산설계구조는 시스템을 몇 개의 하부 해석부로 분리한 다음, 각 해석부에 최적화부를 포함시켜 해석부가 단순히 해석만을 수행하는 것이 아니라 시스템의 설계도 일정부분 담당하게 만든 구조이다. 분산설계구조에서 각각의 부체계(해석부와 최적화부의 조합)는 시스템이 내려준 설계조건과 내부의 구속조건만을 만족하는 역할을 수행하며, 시스템은 적어도 2개 이상의 부체계에 연관된 공통변수들의 동일성을 만족시키는 역할을 수행한다. 따라서 분산설계구조는 분산해석구조에 비해 부체계(해석부와 최적화부의 조합)의 설계권한이 확대되었으며, 부체계가 증가하더라도 전체 시스템에는 큰 변화가 없는 구조이다.

이러한 최적화 구조는 분산해석구조에 비하여 최적부의 부담을 비약적으로 감소시켜준다. 항공기 날개를 설계할 경우, 구조해석부의 날개 내부구조를 결정하는 수많은 설계변수(예, 날개 길이방향 요소용력, 각 부재 무게 등)나 공력해석부의 날개단면 양력계수(Sectional Lift Coefficient) 제한치 등은 각각의 해석부내에서 처리하며, 시스템 최적화부에서는 공력해석과 구조해석에 입력 또는 출력변수로 동시에 이용되는 날개형상, 면적, 양항비, 성능변수 등만을 고려하면 되는 것이다.

그러나 이 구조는 공통설계변수의 수가 많아질수록, 즉 각 해석부가 밀접하게 연결되어 있을수록 수렴특성이 떨어지는 단점이 있다. 또한, 시스템 최적화부에서 내려보낸 공통설계변수가 부체계 최적화부의 제한조건을 만족하지 못하는 값일 경우, 해석부에서 다시 올려 보내는 공통설계변수는 시스템 최적화부의 수렴을 저해하게 된다. 따라서 분산해석 구조의 수렴성을 증대시키기 위해서는 공통설계변수가 전체 해석부의 해집합 범위 내에 미리 존재해야 한다는 전제가 필요한데, 공통설계변수의 수가 많아질 경우, 합당한 해집합 범위 내에서 공통설계변수의 초기치를 결정하기란 쉽지 않다.

다음은 분산해석구조의 두 가지 방식인 부체계 동시 최적화기법과 공동 최적화에 대한 내용이다.

가. 부체계 동시 최적화(CSSO, Concurrent Subspace Optimization)

부체계 동시최적화[1,7]는 각 부체계의 최적화부(Subspace Optimizer)가 타 부체계의 설계제한 조건을 선형적으로 근사화(GSE, Global Sensitivity Equation)하여 사용함으로써 수렴성을 증대 시킨 구조로서 각 부체계가 모든 타 부체계의 구속조건을 민감도를 계산하여 이를 시스템에 전파하는 비계층적(Non-hierarchical)구조를 가지는 형태이다. 이 구조는 공통 설계변수가 증가하거나

해석부가 밀접하게 연관되어 있는 시스템에는 한계를 가진다. 상용MDO시스템에는 SysOpt가 있다.

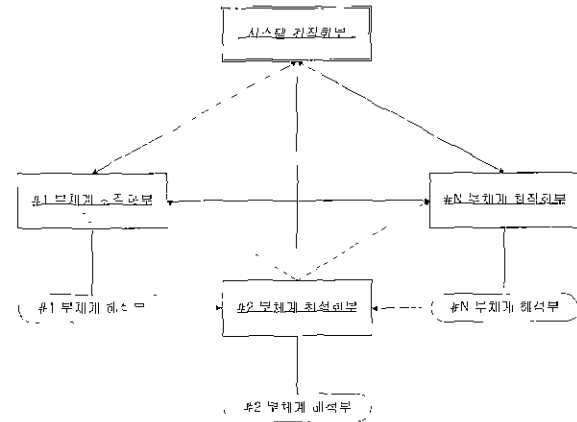


그림 4. 부체계 동시 최적화 구조.

나. 공동최적화(Collaborative Optimization)

공동최적화구조[4,6,9,11]는 비계층 구조인 CSSO와는 달리 계층적(Hierarchical) 구조를 가진다. 부체계간의 상호 동일성을 유지하면서 각 부체계에 최대한의 설계권한을 주는 것이 특징인데, 각 부체계는 시스템에서 내려 보내준 공통설계변수와 내부설계변수를 임의로 수정할 수 있는 권한을 가진다.

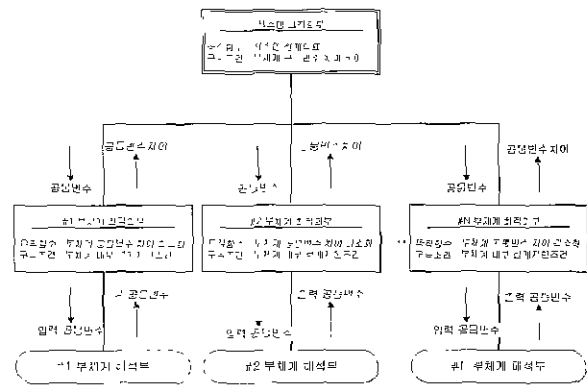


그림 5. 공동 최적화 구조.

그림 5.는 공동 최적화 구조를 개념적으로 도시한 것이다. 시스템 최적화부는 설계목표를 목적함수로 하고, 각 부체계의 목적함수 값을 등가의 구속조건으로 설정하는 구조를 가진다. 즉, 시스템의 구속조건은 시스템에서 부체계에 내려보낸 공통설계변수와 각 부체계에서 설정한 공통변수의 차이가 없어질 때 만족된다.

시스템에서 공통변수의 초기값을 각 부체제로 내려 보내면, 부체계는 시스템에서부터 내려온 이 값을 목표

치로 보고 이를 만족시키기 위하여 내부적으로 변수를 설정한 다음, 해석을 수행하고, 결과변수를 부체계 최적화부에 제시하여 목표치를 만족할 때까지 반복계산을 수행한다. 부체계가 시스템에서 내려보낸 목표치를 만족하지 못할 때, 즉, 시스템에서 내려보낸 변수들이 설계불가능(Infesible)영역에 존재할 경우, 부체계는 시스템에 0이 아닌 값을 되돌려 주게 된다. 이 때, 시스템은 0이 아닌 제한조건으로부터 민감도를 계산하여 보다 나은 변수집합을 다시 부체계로 내려 보내는 과정을 반복하여 최종적으로 해집합을 도출하게 된다.

부체계 최적화부는 시스템에서 설정한 공통변수와 해석부의 계산결과 또는 내부설계변수의 차이를 최소제곱(Least Squared)의 형태로서 최소화하는 목적함수를 가짐으로써 독립적인 최적화부를 형성한다. 독립적인 최적화부를 가지게 된다는 것은 부체계가 보다 많은 설계권한(autonomy)을 가짐으로써 특정 부체계를 설계 시스템에 매우 쉽게 연결할 수 있다는 것을 의미한다. 예를 들어, 공력형상 최적화에 주로 사용되는 Adjoint 기법[18]으로 미리 개발한 날개최적설계 프로그램이 있다면, 동시최적화구조의 날개설계 프로그램에, 별다른 수정 없이 쉽게 적용할 수 있다는 것이다. 이러한 장점은 기존 설계/해석 코드의 재 사용율을 높임으로써 개발기간 및 비용을 단축시키는데 기여한다.

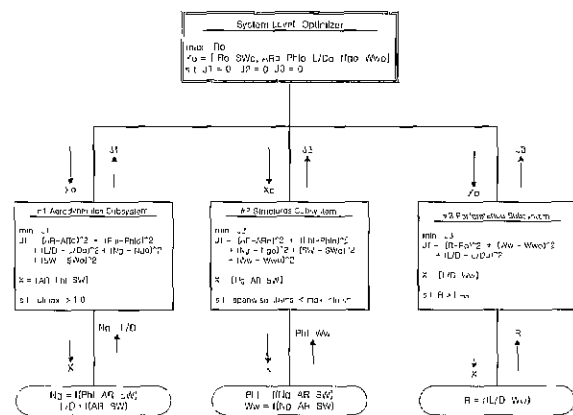


그림 6. 공동 최적화 구조의 항공기 개념설계 적용.

- R = 순항거리, Cruise Range
- AR = 날개의 가로세로비, Aspect Ratio
- Phi = 날개의 비틀림 각, Twist Angle
- SW = 날개의 뒤젓힘 각, Sweepback Angle
- L/D = 기체의 양항비, Lift to Drag Ratio
- Ng = 하중계수, Load Factor
- Ww = 날개중량, Wing Weight
- Clmax = 날개면 양력계수, Section Lift Coefficient

그림 6.은 공동최적화 구조로 구현한 항공기의 날개

설계 문제로서 문헌[4]의 내용을 재구성한 것이다. 최적화 문제의 목적함수는 항공기의 항속거리, R을 최대화하는 것으로 설정되어 있으며, 부체계는 항공기 설계에 있어 가장 기본이 되는 공력과 구조. 그리고 Brequet방정식을 이용하여 항속거리를 계산하는 세 부분으로 분해되어 있다. 시스템은 항속거리(R), 날개형상관련 변수(AR, SW, Phi), 공력성능변수(L/D, Ng), 날개중량(Ww) 등 적어도 2개 이상의 부체계에서 사용되는 공통변수 들을 설계변수로 사용하고 있다. 시스템에서 설계변수의 초기값(Xo)을 설정하여 부체계로 내려보내면, 각 부체계에서는 이를 목표치로 설정(J)하고, 해석부의 입력변수(공력의 경우, AR, SW, Phi)를 설정하여 해석부를 구동시킨다. 해석부에서는 입력변수로부터 공통 출력변수(공력의 경우, Ng, L/D)를 계산하여 부체계의 목적함수 J를 계산하게 된다. 부체계 최적화부는 J가 가장 작은 값을 가질 때까지 이러한 과정을 반복하여 J값을 시스템에 돌려주게 된다. 시스템은 부체계로부터 올라온 J값과 목적함수 R의 민감도를 계산하여 다시 설계변수들을 설정하고 이를 부체계로 내려보내는 과정을 반복하게 된다.

부체계 내에서 내부적으로 존재하는 설계제한 조건(공력의 경우, Clmax 하한설정)은 부체계의 목적함수를 계산할 때 반영되기 때문에 시스템에서 별도로 고려하지 않아도 최적화 과정에서 내재적(Implicit)으로 타 부 체계에 영향을 미친다. 따라서 각 부체계가 모든 타 부 체계의 제한조건 민감도를 계산하여 이를 시스템에 전파하는 부체계 동시최적화(CSSO)에 비해 시스템과 부 체계 사이의 자료이동을 현저히 감소시킨다. 이러한 특징들로 인하여 동시최적화는 병렬처리(Pauallel Processing)에 적합하다. 즉, 시스템부와 각 부체계를 독립적인 PC나 WORKSTATION에 탑재하고 이를 연결하여 동시에 계산을 수행하게 함으로써 최적화시간을 감소시킬 수 있는 것이다.

그러나 공동최적화 구조는 부체계의 내부변수는 많 이 있으면서 부체계 사이가 느슨하게 연결된 시스템의 설계에 적합한 구조로서, 각 부체계가 서로 밀접하게 연결된(공통변수가 많은) 시스템에서는 느린 수렴특성 [19]을 가지는 것으로 알려져 있다.

5. 결론

지금까지 MDO의 최적설계구조를 기초적인 MDO구조인 통합해석 구조, 더 나아가 분산해석 구조, 가장 발전된 형태인 분산설계 구조로 나누어 기술하였다. 현재 개발되고 있는 대부분의 MDO프레임워크가 선택하고 있는 설계구조는 분산해석 구조이지만 가까운 미래에는 학술연구의 수준에 머물러 있는 분산설계 구조가 주류를 차지할 것으로 예측되고 있다. 그러나 분산설계 구조는 시스템이 내부적으로 밀접하게 연결되어 있을 경우 발생하는 느린 수렴성 등이 해결해야 할 과제

남아 있다.

전술한 세 가지의 설계구조는 각각 고유한 장단점을 가지고 있어 어떤 것이 우수하다고 단언할 수는 없다. 경험식 등을 주로 사용하고 설계자가 전체 설계흐름을 충분히 파악할 수 있는 초기 개념설계단계에서는 통합 해석 구조가 유리할 수 있으며, 시스템이 대형화하고, 개념설계뿐만 아니라 기본설계, 전 수명주기를 고려한 설계를 원할 경우, 분산설계 구조를 택하는 것이 유리할 수 있다. 그러나 현재 가장 보편화되고, 검증된 설계구조로서 많은 MDO프레임워크가 기본적으로 채택하고 있는 구조는 분산해석 구조이다. 따라서 설계자는 자신에게 주어진 설계환경에 가장 적합한 설계구조를 선택하는 것이 중요하다.

[참고문헌]

- [1] A.O. Salas, J.C. Townsend, Framework Requirements for MDO Application Development, AIAA-98-4740, 1998
- [2] N.M. Alexandrov, R.M. Lewis, Comparative Properties of Collaborative Optimization and Other Approaches to MDO, NASA-CR-2999-209354, 1999
- [3] Ilan Kroo, Multidisciplinary Optimization Applications in Preliminary Design-Status and Direction, AIAA-00-0000
- [4] Robert D. Braun, Collaborative Optimization : An Architecture for Large-Scale Distributed Design, Stanford Univ. Ph.D. Thesis, 1996
- [5] Masami Takai, A New Architecture and Expert System for Aircraft Design Synthesis, Stanford Univ. Ph. D. Thesis, 1990
- [6] N.M. Alexandrov, Initial Results of An MDO Method Evaluation Study, AIAA-98-4884
- [7] S.L. Padula et al, Multidisciplinary Optimization Branch Experience Using iSIGHT Software, NASA-TM-209714, 1999
- [8] Sobieski, Sensitivity Analysis and Multidisciplinary Optimization for Aircraft Design :Recent Advances and Results, J. Aircraft Dec. 1990, pp 993
- [9] Ilan Kroo. et al, Multidisciplinary Optimization Methods for Aircraft Preliminary Design, Aircraft Aerodynamics and Design Group, Stanford Univ. AIAA 94-4325, 1994
- [10] Srinivas Kodiyalam, Evaluation of Methods for Multidisciplinary Design Optimization(MDO), Phase I, NASA-CR-1988-208716, 1988
- [11] Ilan Kroo, MDO for Large-Scale Design, Multidisciplinary Design Optimization - State of the Art, SIAM, ISBN 0-89871-359-5, 1997
- [12] Wakayama, S., Ilan Kroo, Subsonic Wing Design Using Multidisciplinary Optimization, AIAA-94-4449, 1994
- [13] D. D. Meyer, Development of Integrated Programs for Aerospace-Vehicle Design (IPAD) – Reference Design Process, NASA-CR-2981, 1979
- [14] Ilan Kroo, An Introductory System for Aircraft Design Optimization, AIAA-92-1190, 1992
- [15] 전성진, 항공기 개념설계 단계에서의 시스템 최적화에 관한 연구, 석사학위논문, 서울대, 1997
- [16] M, J. Buckley. et al., Design Sheet : An Environment for Facilitating Flexible Trade Studies During Conceptual Design, AIAA-92-1191, 1992
- [17] R. Krishnan, Evaluation of Frameworks For HSCT Design Optimization, NACA-CR-1998-208731, NASA LARC.
- [18] R.M. Lewis, Numerical Computation of Sensitivity and the Adjoint Method, NASA-CR-1997-206247
- [19] Rohl et al., A Collaborative Optimization Environment for Turbine Engine Development, AIAA-98-4734. 1998

이 수 창

1990년 서울대학교 공과대학 항공공학과(학사), 1992년 서울대학교 공과대학 항공우주공학과(석사), 1997년 서울대학교 공과대학 항공우주공학과(박사). 1997년 2월 - 현재국방과학연구소 선임연구원. TEL) 042-821-3611, FAX) 042-821-2224
E-mail) suchang@chollian.net

안 중 기

1992년 한국항공대학교 항공기제공학과(학사), 1994년 한국항공대학교 항공공학과(석사), 1994년 3월 - 현재국방과학연구소 선임연구원
TEL) 042-821-3482, FAX) 042-821-2224.
E-mail) jkahn69@lycos.co.kr