

KARI MDAO System Development Plan and Status for the Hybrid eVTOL Aircraft Design

February 10, 2023

Seung-Kil Paek

Korea Aerospace Research Institute

- **Interests in UAM(Urban Air Mobility) or AAM(Advanced Air Mobility) is in full swing.**
 - The Vertical Flight Society (US) counts almost 700 entrants in the AAM industry with new ones added on a weekly basis
- **KARI(Korea Aerospace Research Institute) should provide the Korean industry with the core technology for those types of aircrafts as a Korean government funded research institute since 1989.**
 - Developing also an eVTOL aircraft a.k.a. OPPAV¹
 - Developing MDAO² technology on eVTOL aircraft through an internal project since 2019.
 - Started a new five-year-long internal project to develop a MDAO system.



¹OPPAV : Optionally Piloted Personal Air Vehicle, 2019 ~ 2023

²MDAO : Multi-disciplinary analyses and optimization



- **There have been various MDAO Frameworks, related software and digital contents:**
 - CEASIOM(CEASIOMpy), pyMDO, OpenMDAO, 3DOPT, MDOPT, CPACS, RCE, and etc.
- **DLR (Germany) started the development of the aircraft design environment from 2005.**
- **DLR created the open-sourced CPACS as a common namespace.**
 - a standard for exchanging aircraft design information among aviation-related institutions of DLR.
- **DLR developed the open-sourced RCE as a process integration framework**
 - It supports collaborative and distributed work.



- **A collaborative MDAO system**

- the design tools implemented by each discipline specialist are operated in independent software environments
- their outputs are shared by design tools in other fields

- **DLR has been conducting collaborative optimization design research in a distributed environment early on.**

- **In the AGILE project,**

- AGILE : Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts
 - from 2015 to 2018, part of the Horizon 2020 program.
- DLR has reached the level of designing an aircraft by implementing interworking between processes distributed across three continents, 19 institutions, and heterogeneous platforms



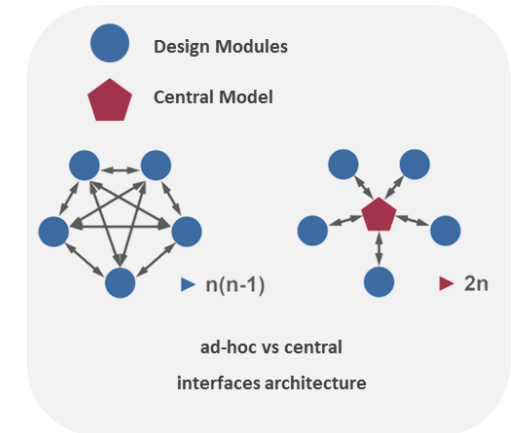
- **Software structure**

- Collaborative software structure will be used:
- The independent analysis method established by using the most familiar software tools such as Excel, script, commercial S/W, and in-house software, in the most appropriate operating environment by each discipline expert will be combined as a process.
- The concept of establishing a process is not to create a new tool, but to transform an existing tool into a form that can be shared as much as possible on the network
- In Korea, optimal design researches under such a collaborative system have hardly been conducted.
 - Hence this research will be a big challenge.



• Central Data Model

- Central data model will be used to minimize the interfaces.
- Communication between different disciplines is achieved through design parameters.
- In KARI, these design variables were independently defined and there was no common parameter pool until now.
- In this project, CPACS¹ is used to control design variables in all disciplines as much as possible
- Using CPACS, is like forcing people with different mother tongues to use only one foreign language!
 - Big challenge
- CPACS is mainly used for conventional fixed wing aircrafts with jet fan engines until now.
 - It needs much enhancements so that it can be used for UAM aircraft design



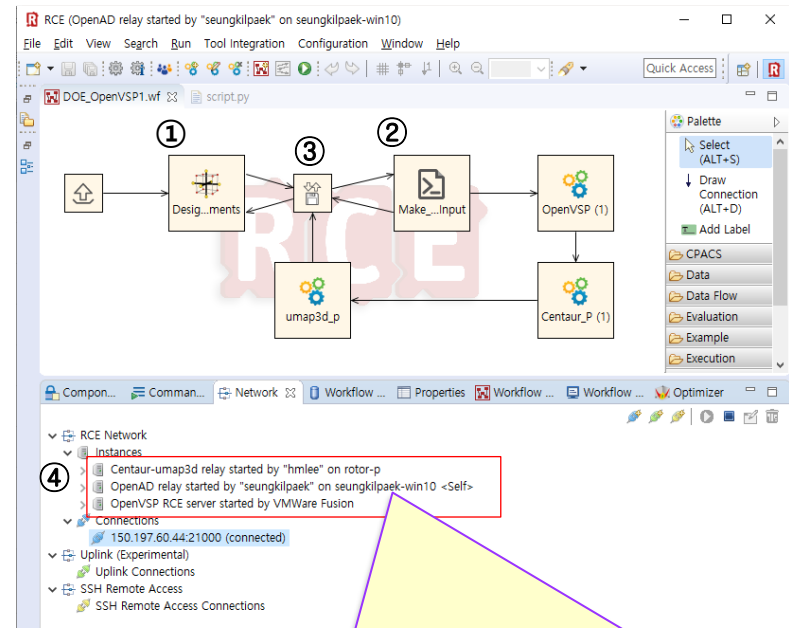
Courtesy of <https://github.com/DLR-SL/CPACS>

¹CPACS : Common Parametric Aircraft Configuration Schema. A data definition for the air transportation system. Opensource digital content hosted in github.



• Process Integration

- KARI had used ModelCenter® for the process integration
 - Kang, H.-J. “Design optimization of QTP-UAV prop-rotor blade using ModelCenter®”. *Journal of Aerospace System Engineering*, Vol. 11, No. 4, pp. 36—43, 2017.
- Recently, RCE, an open-source replacement have been tested instead.
- The user experience with RCE was satisfactory.
- RCE will be used for this project.



Three engineering tools was integrated:

- an aircraft geometry creating tool using OpenVSP and Rhino®
- a commercial mesh generation tool CENTAUR®
- an inhouse CFD solver to make geometry design study

- **2022 : MDO System Requirements & System Architecture Design**

- Define the requirements for the MDAO system
- Define the tool list to be integrated in the MDAO system
- Define the interface using the central data model.
- Define the workflow of the MDAO system as XDSM¹ format

- **2023 :**

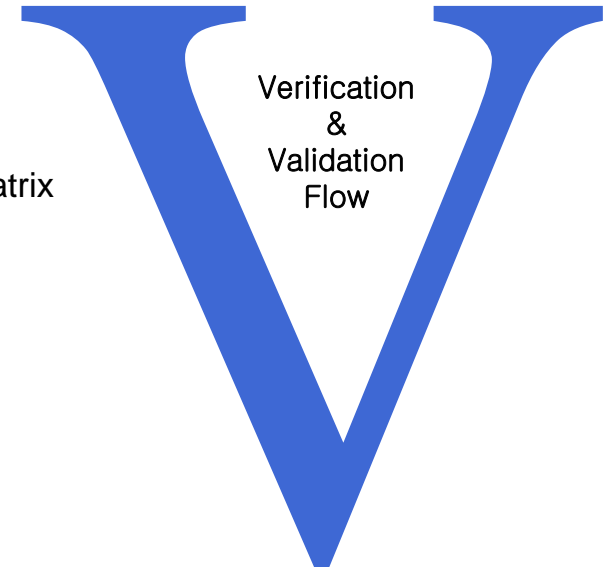
¹XDSM : Extended Design Structure Matrix

- Verification of System Architecture
- Development of tools and discipline workflows

- **2024 : Verification of each discipline workflow**

- **2025 : Verification of the integrated workflow**

- **2026 : Verification of the optimal design capability**



- **Based on system engineering methodology for software development.**

- Project management with Redmine software
- Configuration control with Git software



Concept Design Study from 2022 to 2023

- **KARI is making a concept design of a 6 seated UAM aircraft:**
 - Define the top-level requirement and analysis standard
 - the follow-up study of the OPPAV aircraft
 - Decide the design variables, analysis cases, and major performance objectives
 - Establish an optimal design problem
 - Create a concept design.

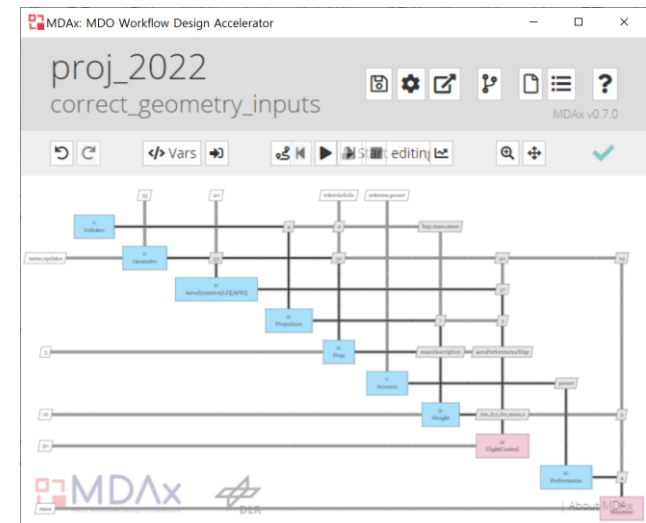


Preliminary Design Study from 2024 to 2026

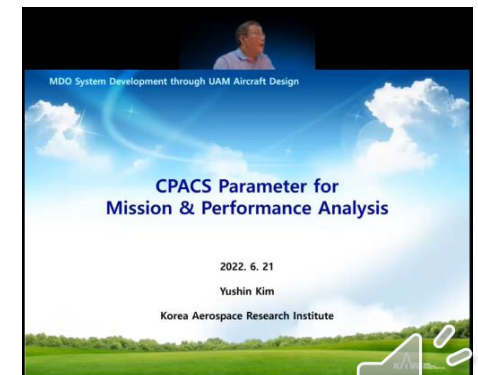
- **2024 :**
 - Build a surrogate model for each discipline workflow
 - The collected surrogate models are integrated into the total workflow.
 - The work will be used for optimal design study
 - The optimal design will be reviewed.
- **2025 :**
 - The discipline workflows will be integrated altogether.
 - DOE using the integrated workflow will be made.
 - Optimal solution using DOE results
- **2026**
 - Direct optimization with the integrated workflow



- For the design of the MDAO system, such as the input/output definition of each component and the workflow design combining each component, tools are needed:
 - KADMOS
 - An open source python package based on the data schema called CMDOWS
 - Little documentation and not easy to start
 - MDAX : MDO Workflow Design Accelerator
 - A web-based GUI application by DLR
 - An easy-to-use substitute of KADMOS
 - more promising than KADMOS.



- **KARI have been studying the usability of various DLR software tools:**
 - RCE, TiGL, TiXI, VAMPzero, CPACS and MDAX
- **KARI-DLR cooperation was proposed by KARI in 2020 and welcomed by DLR.**
 - Enhancement of DLR tools through the test and evaluation
 - Discussions were made to materialize the cooperation through 2021.
 - An online meeting was made between the aeronautics research directorate of KARI and the institute of system architecture in aeronautics of DLR in March 2022 to discuss the practical cooperation.
- **Online workshops have been held four times through the video conferencing**
 - from May 2022 to Aug 2022
 - CPACS and TiGL Tutorial and Q&A
 - Hands-on exercises using MDAX via the web “as-a-service”
 - KARI work procedures for flight control, mission analysis, and geometry generation
 - Totally 43 people from KARI side attended the workshops.



• MDO System Requirement Documentation

1.1. 목표

본 MDO 시스템(Hybrid Electric urban air mobility Aircraft Design system, 이하 HEAD)은 다음과 같은 목표를 갖는다. 이 목표는 HEAD 설계를 위한 최상위요구도(Top Level Requirement)로 간주한다. 상위요구도의 Prefix 는 TLR_ 이며 최상위 요구도이므로 이 이상의 추적성은 없다.

[TLR_001]
HEAD 는 5인승급 하이브리드 eVTOL 항공기의 기본설계에 사용할 수 있어야 한다.

[TLR_002]
HEAD 는 중앙 데이터 모델로 CPACS[AMJN20] 를 사용하여야 한다.

[TLR_003]
HEAD 의 각 시뮬레이션 도구(이하 컴포넌트)는 RCE[BFF+21] 에 통합되어야 하며, 원격 호출이 가능해야 한다.

[TLR_004]
HEAD 의 각 컴포넌트는 독자적으로 실행 가능해야 한다.

[TLR_005]
HEAD 의 각 컴포넌트를 연동하여, 최적 설계 시스템으로 작동하여야 한다.

[TLR_006]
HEAD 는 시스템공학적 방법으로 개발하여야 한다.

위 요구도에서 HEAD 는 컴포넌트와 시스템을 동시에 개발해야 하는 것임을 알 수 있다. 특히, [TLR_005]는 HEAD 가 고정된 것이 아니라 매우 유연한 시스템을 의미한다. 각 컴포넌트의 입력과 출력을 다른 컴포넌트에 연결하여 매우 다양한 최적화 문제를 풀 수 있는 것이다. 이러한 유연성으로 인해 시스템의 설계가 어렵게 된다.

7. 참고 문헌	
8. 색인	[HLR_002]는 [TLR_001]에 의한 eVTOL 항공기 유형을 제한하는 요구도이다. 모든 유형의 항공기를 설계하는 것은 너무 큰 목표이므로, 과제 규모에 맞게 제한하도록 한다. [HLR_004]는 기본설계에 사용하려면 시뮬리케임을 조정할 수 있는 장치가 필수이기 때문에 선정하였다.
4. 인터페이스 정의	으로 항공연구소
5. 내부 워크플로우	© 저작권 2022, KARI.



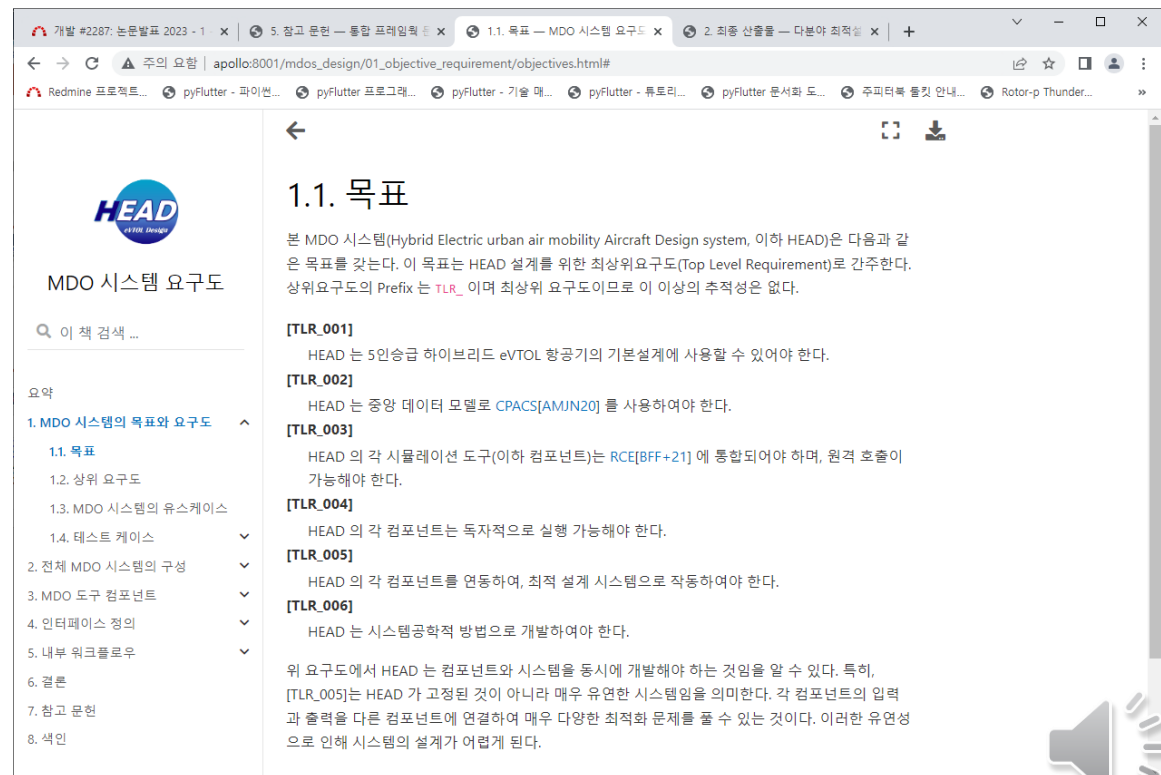
Internal Workshops

- **MDO System design needs the cooperation between the experts on physical models constituting the system.**
- **Intensive discussions and brain storming were needed in a short time.**
- **Six internal workshops were made in 2022.**
 - February : eVTOL concept design seminar
 - March : 1st optimal design problem definition
 - May : CPACS-RCE basics and 1st N2 chart build
 - July : 2nd N2 chart build
 - August : 2nd optimal design problem definition
 - September : MDO system design



• MDO System Requirement – Top Level Requirements

- Six top level requirements
 - Preliminary design for 5 seated hybrid eVTOL aircrafts
 - CPACS as central data model
 - RCE as the process integrator
 - Each Component to be able to separate execution
 - Interoperable components
 - System engineering to be followed



1.1. 목표

본 MDO 시스템(Hybrid Electric urban air mobility Aircraft Design system, 이하 HEAD)은 다음과 같은 목표를 갖는다. 이 목표는 HEAD 설계를 위한 최상위요구도(Top Level Requirement)로 간주한다. 상위요구도의 Prefix 는 TLR_ 이며 최상위 요구도이므로 이 이상의 추적성은 없다.

[TLR_001]
HEAD 는 5인승급 하이브리드 eVTOL 항공기의 기본설계에 사용할 수 있어야 한다.

[TLR_002]
HEAD 는 중앙 데이터 모델로 CPACS[AMJN20] 를 사용하여야 한다.

[TLR_003]
HEAD 의 각 시뮬레이션 도구(이하 컴포넌트)는 RCE[BFF+21] 에 통합되어야 하며, 원격 호출이 가능해야 한다.

[TLR_004]
HEAD 의 각 컴포넌트는 독자적으로 실행 가능해야 한다.

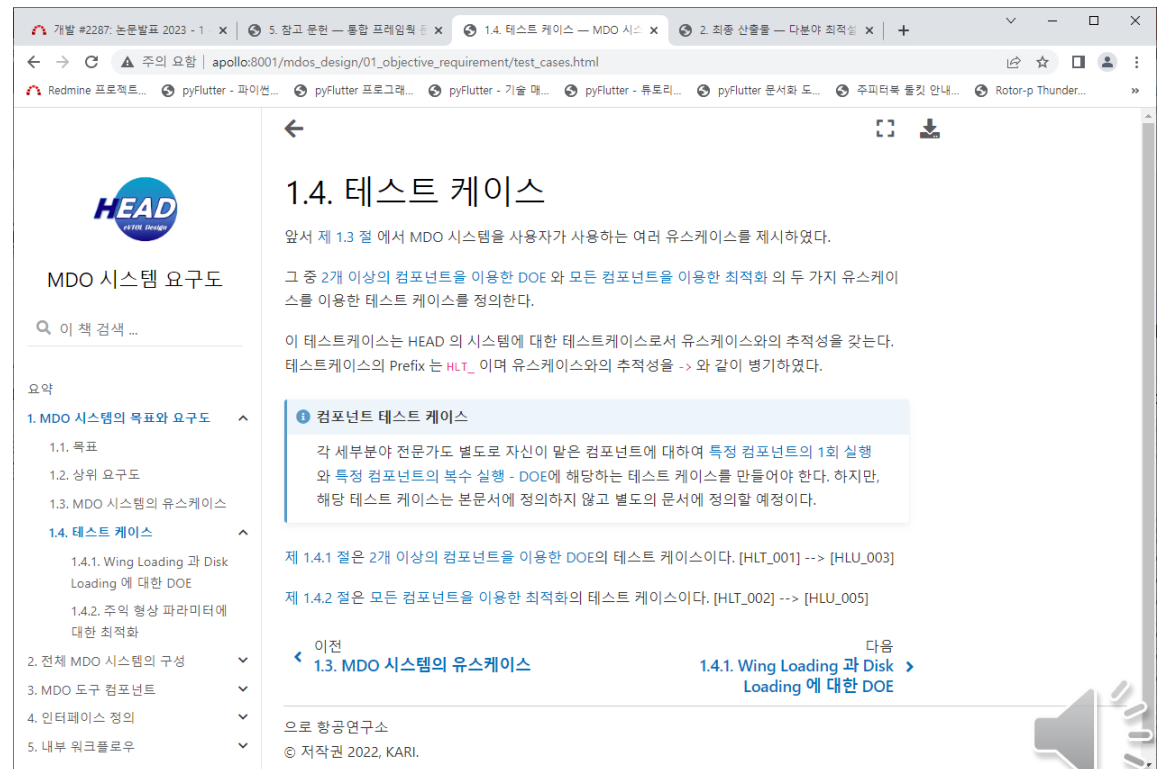
[TLR_005]
HEAD 의 각 컴포넌트를 연동하여, 최적 설계 시스템으로 작동하여야 한다.

[TLR_006]
HEAD 는 시스템공학적인 방법으로 개발하여야 한다.

위 요구도에서 HEAD 는 컴포넌트와 시스템을 동시에 개발해야 하는 것임을 알 수 있다. 특히, [TLR_005]는 HEAD 가 고정된 것이 아니라 매우 유연한 시스템임을 의미한다. 각 컴포넌트의 입력과 출력을 다른 컴포넌트에 연결하여 매우 다양한 최적화 문제를 풀 수 있는 것이다. 이러한 유연성으로 인해 시스템의 설계가 어렵게 된다.

• MDO System Requirement – Use Cases and Test Cases


- Five use cases
 - One execution of a specific component
 - Design of Experiment (Multiple executions) using a specific component
 - Design of Experiment (Multiple executions) using plural components
 - Optimization using plural components
 - Optimization using all components



개발 #2287: 논문발표 2023 - 1 x 5. 참고 문헌 — 통합 프레임워크 x 1.4. 테스트 케이스 — MDO 시스템 x 2. 최종 산출물 — 다분야 최적화 x | +

← → ↻ 주의 요함 | apollo:8001/mdos_design/01_objective_requirement/test_cases.html

Redmine 프로젝트... pyFlutter - 파이썬... pyFlutter 프로그래... pyFlutter - 기술 대... pyFlutter - 튜토리... pyFlutter 문서화 도... 주피터북 툴킷 안내... Rotor-p Thunder...

 HEAD
HEAD DESIGN

MDO 시스템 요구도

이 책 검색 ...

요약

1. MDO 시스템의 목표와 요구도 ^

- 1.1. 목표
- 1.2. 상위 요구도
- 1.3. MDO 시스템의 유스케이스
- 1.4. 테스트 케이스 ^

- 1.4.1. Wing Loading 과 Disk Loading 에 대한 DOE
- 1.4.2. 주익 형상 파라미터에 대한 최적화

2. 전체 MDO 시스템의 구성 v

3. MDO 도구 컴포넌트 v

4. 인터페이스 정의 v

5. 내부 워크플로우 v

1.4. 테스트 케이스

앞서 제 1.3 절 에서 MDO 시스템을 사용자가 사용하는 여러 유스케이스를 제시하였다.

그 중 2개 이상의 컴포넌트를 이용한 DOE 와 모든 컴포넌트를 이용한 최적화 의 두 가지 유스케이스를 이용한 테스트 케이스를 정의한다.

이 테스트케이스는 HEAD 의 시스템에 대한 테스트케이스로서 유스케이스와의 추적성을 갖는다. 테스트케이스의 Prefix 는 HLT_ 이며 유스케이스와의 추적성을 -> 와 같이 병기하였다.

1.4.1. 컴포넌트 테스트 케이스

각 세부분야 전문가도 별도로 자신이 맡은 컴포넌트에 대하여 특정 컴포넌트의 1회 실행 와 특정 컴포넌트의 복수 실행 - DOE 에 해당하는 테스트 케이스를 만들어야 한다. 하지만, 해당 테스트 케이스는 본문서에 정의하지 않고 별도의 문서에 정의할 예정이다.

제 1.4.1 절은 2개 이상의 컴포넌트를 이용한 DOE 의 테스트 케이스이다. [HLT_001] --> [HLU_003]

제 1.4.2 절은 모든 컴포넌트를 이용한 최적화의 테스트 케이스이다. [HLT_002] --> [HLU_005]

이전 < 1.3. MDO 시스템의 유스케이스 > 다음 1.4.1. Wing Loading 과 Disk Loading 에 대한 DOE >

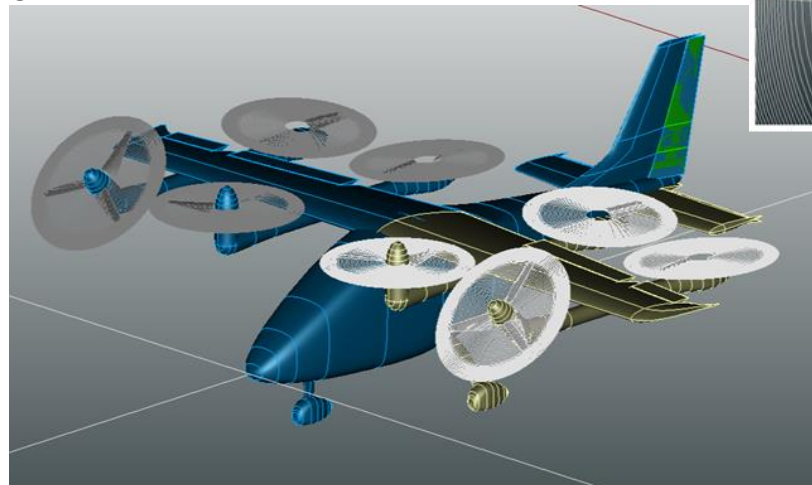
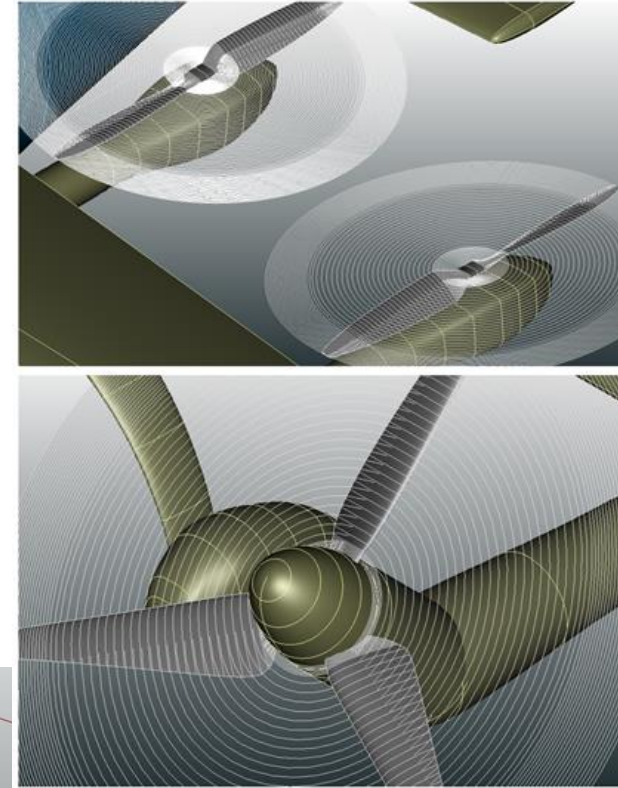
오로 항공연구소
© 저작권 2022, KARI.

MDAO System Design Status

• Major tools to be integrated into the MDAO system:

- Geometry : inhouse software using CPACS, TiXI, TiGL, Python
- Aerodynamics* :
 - ◆ Athena Vortex Lattice (AVL) code
 - ◆ Inhouse source doublet panel method + actuator disk theory
 - ◆ OpenFOAM + actuator disk model
- Flight control : inhouse flight model, MATLAB, CONDUIT
- Mission performance and Prop* : inhouse software, CAMRAD II
- Noise* : inhouse software
- Structure* : inhouse FE modeler, MSC/NASTRAN, Hypersizer
- Hybrid electric propulsion* : inhouse software in MATLAB and SIMULINK

*Need to be developed or upgraded



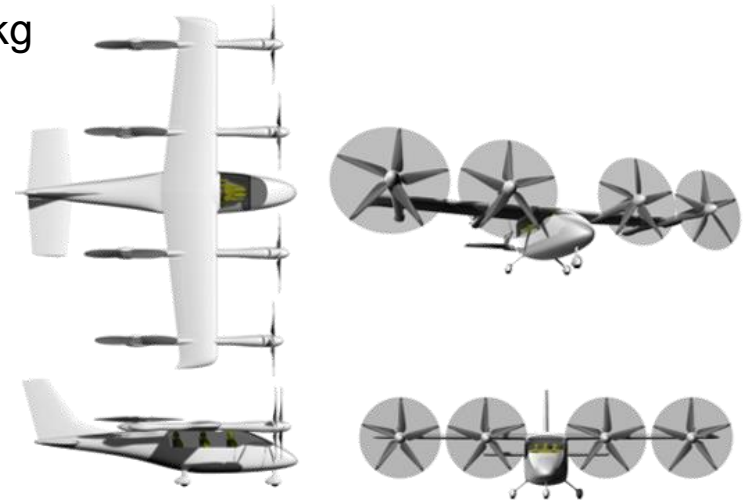
Requirements for a UAM Design

• TLAR

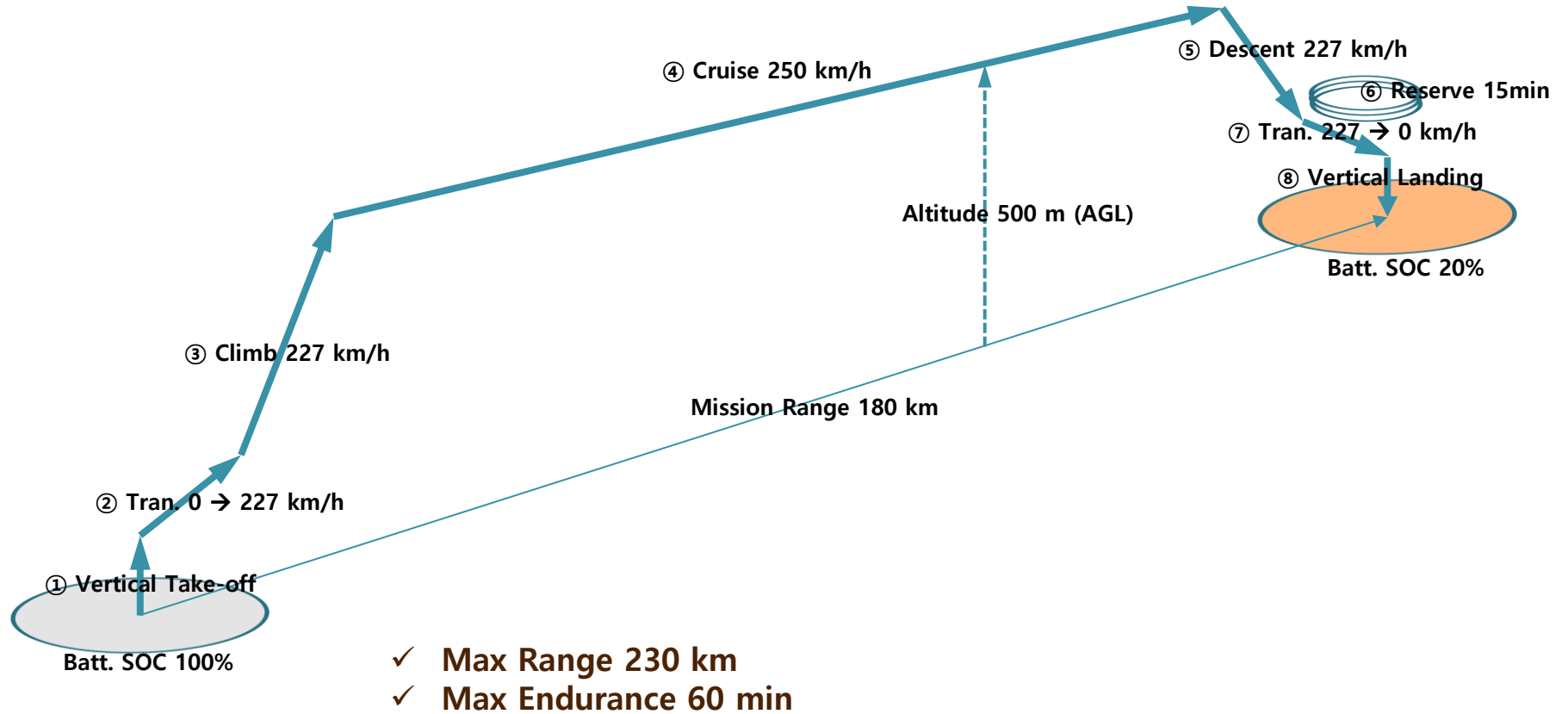
- Payload : more than 550 kg (1 pilot and 5 passengers)
- Cruise speed : more than 250 km/h
- Maximum speed : more than 300 km/h
- Maximum range : more than 200 km
- Maximum endurance : more than 60 minutes
- Maximum altitude : 2 km
- Noise : less than 65dBA at T/O and Landing (100 m distant and at 100 m altitude)

• Assumption

- Battery Pack Energy Density : more than 250 Wh/kg



Mission Profile

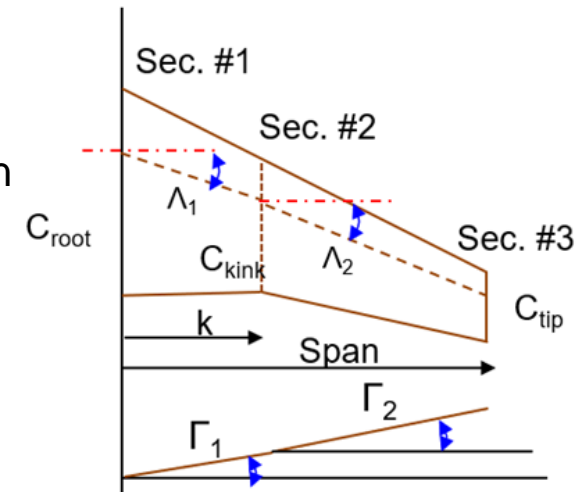


• Design Parameters

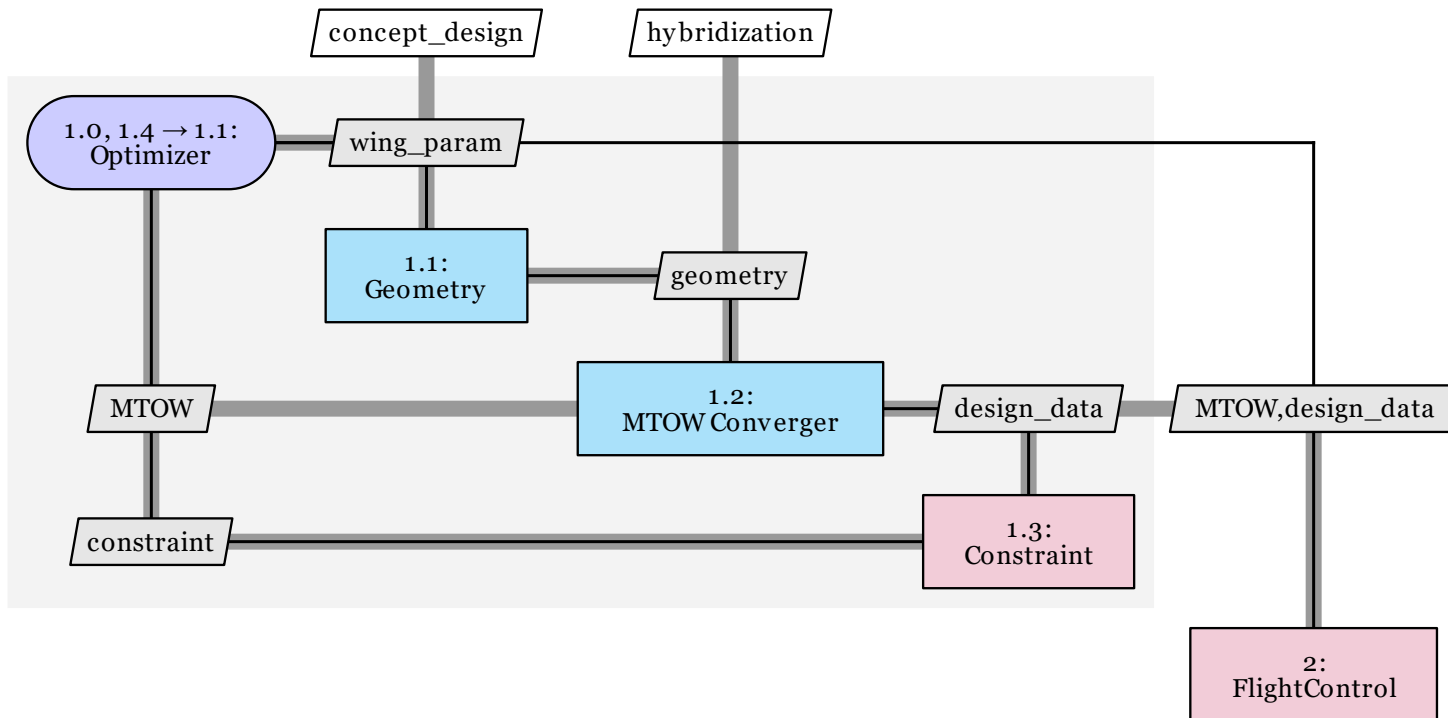
- k, b : the kink position and span of the main wing
- $C_{root}, C_{kink}, C_{tip}$: the chord lengths of the root, kink, and tip sections of the main wing
- Λ_1, Λ_2 : the sweep angles of the root and kink sections of the main wing
- θ_1, θ_2 : the twist angles of the kink and tip sections of the main wing
- Γ_1, Γ_2 : the dihedral angles of the root and kink sections of the main wing
- Hybridization factor

• Objective

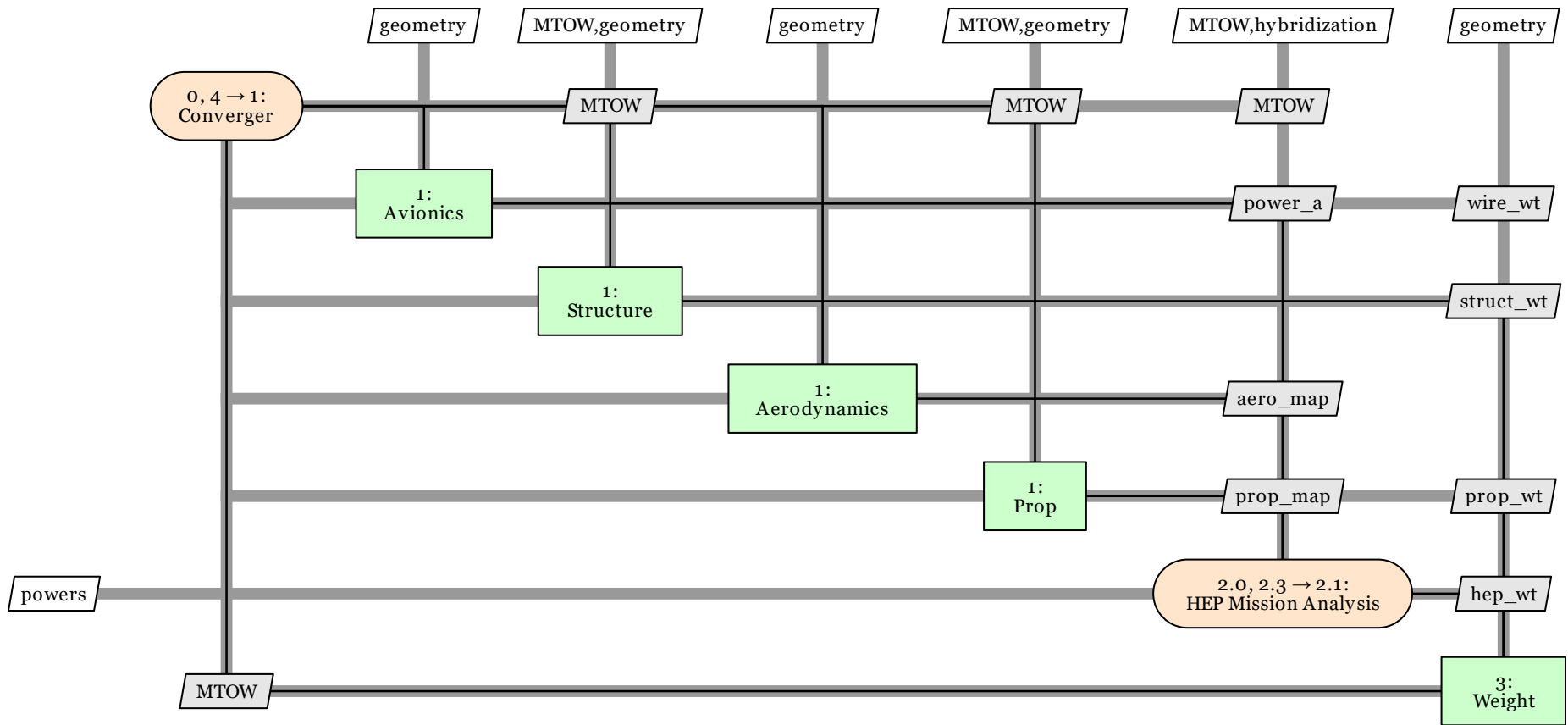
- Minimize the maximum take-off weight



Optimization Problem

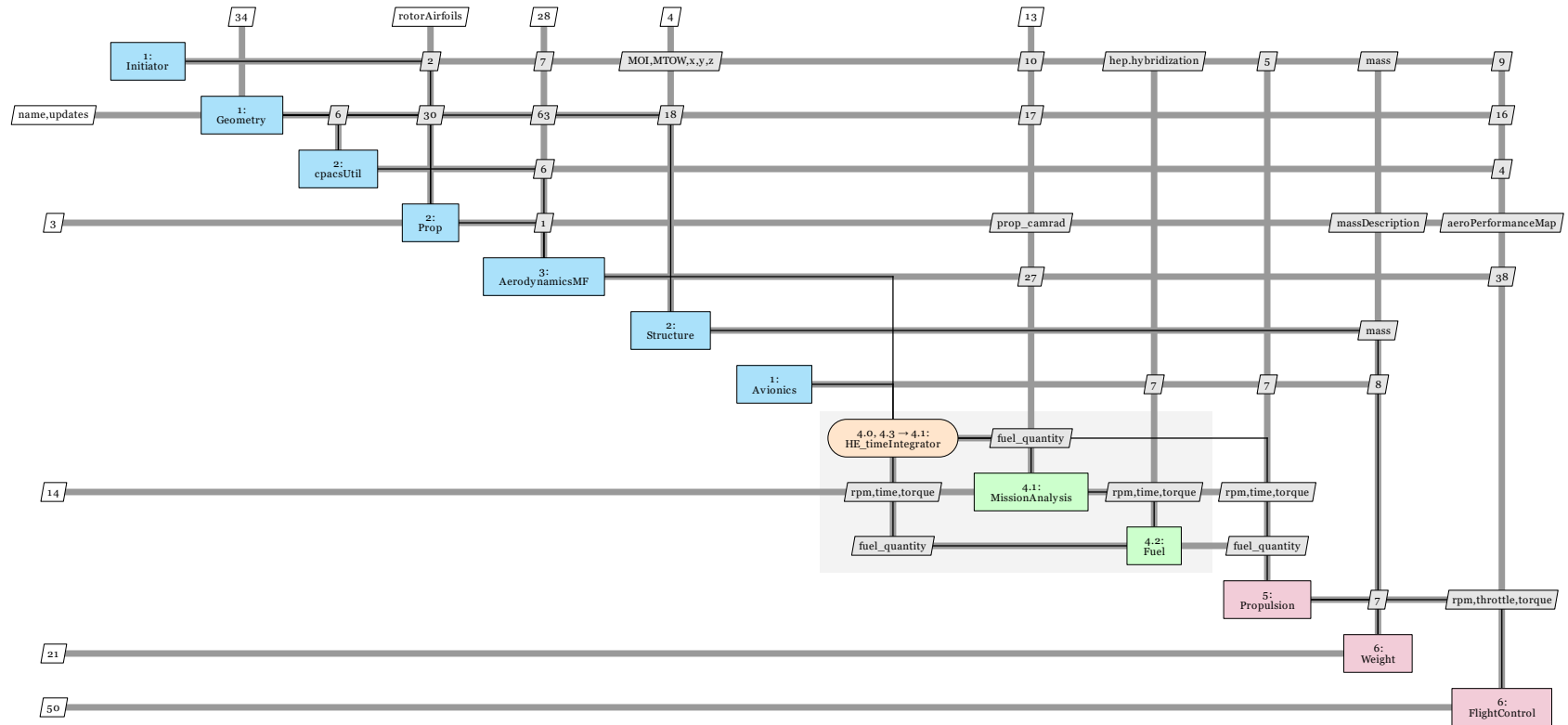


Optimization Problem – MTOW Converger



MDAO System Design Status - XDSM

- Draft XDSM chart



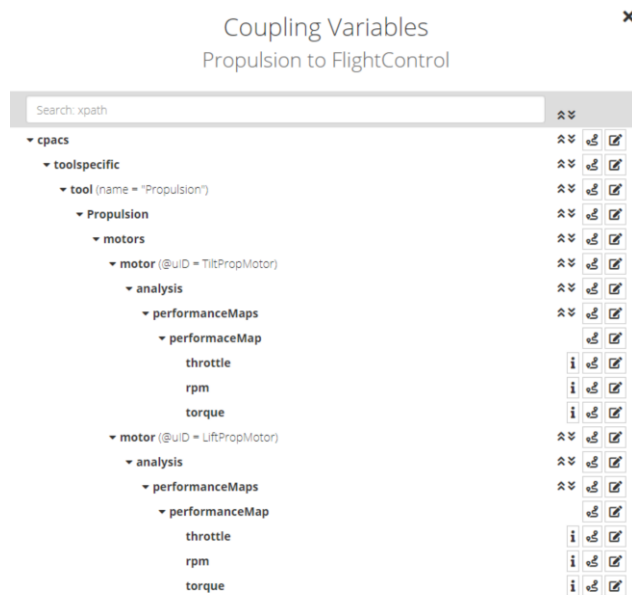
cpacsUtil : from CPACS to higher level parameter



MDAO System Design Status - CPACS

- Enhancement being devised

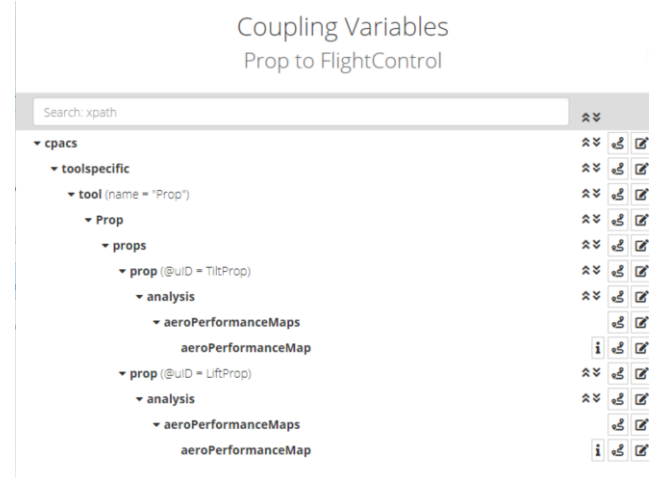
- Elements for UAM specific systems
- Performance Map for Propellers and Motors



Coupling Variables
Propulsion to FlightControl

Search: xpath

- ▼ cpacs
 - ▼ toolspecific
 - ▼ tool (name = "Propulsion")
 - ▼ Propulsion
 - ▼ motors
 - ▼ motor (@uiD = TiltPropMotor)
 - ▼ analysis
 - ▼ performanceMaps
 - ▼ performanceMap
 - throttle
 - rpm
 - torque
 - ▼ motor (@uiD = LiftPropMotor)
 - ▼ analysis
 - ▼ performanceMaps
 - ▼ performanceMap
 - throttle
 - rpm
 - torque



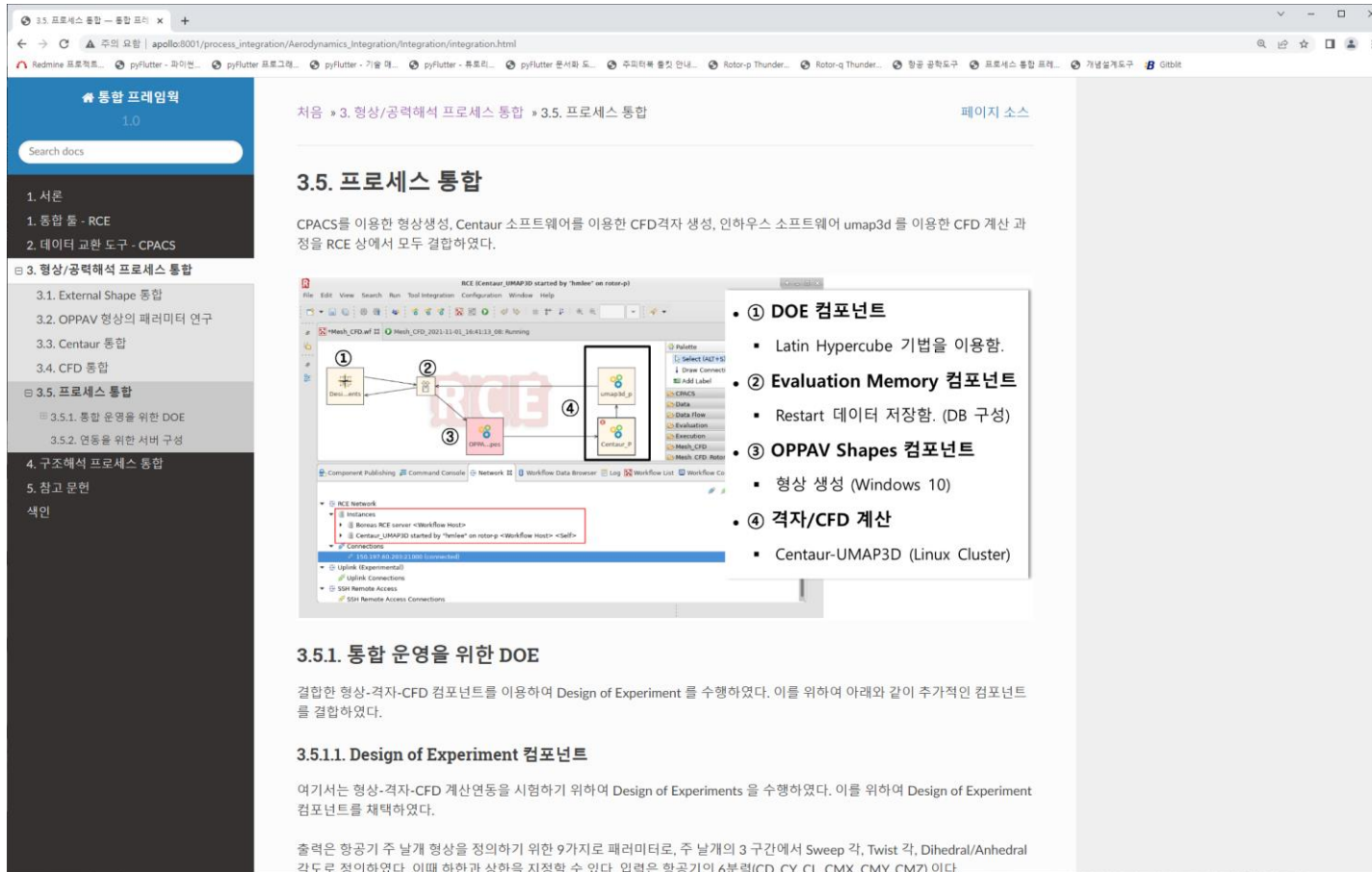
Coupling Variables
Prop to FlightControl

Search: xpath

- ▼ cpacs
 - ▼ toolspecific
 - ▼ tool (name = "Prop")
 - ▼ Prop
 - ▼ props
 - ▼ prop (@uiD = TiltProp)
 - ▼ analysis
 - ▼ aeroPerformanceMaps
 - aeroPerformanceMap
 - ▼ prop (@uiD = LiftProp)
 - ▼ analysis
 - ▼ aeroPerformanceMaps
 - aeroPerformanceMap



- User manual is being made



통합 프레임워크 1.0

1. 서론

1. 통합 툴 - RCE

2. 데이터 교환 도구 - CPACS

3. 형상/공력해석 프로세스 통합

3.1. External Shape 통합

3.2. OPNAV 형상의 파라미터 연구

3.3. Centaur 통합

3.4. CFD 통합

3.5. 프로세스 통합

3.5.1. 통합 운영을 위한 DOE

3.5.2. 연동을 위한 서버 구성

4. 구조해석 프로세스 통합

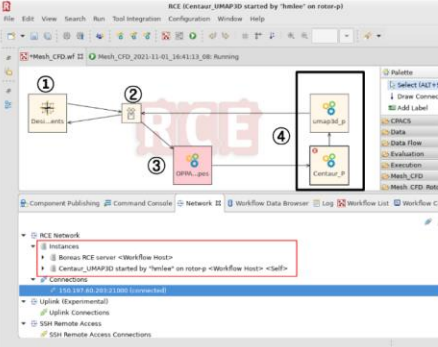
5. 참고 문헌

색인

처음 » 3. 형상/공력해석 프로세스 통합 » 3.5. 프로세스 통합 페이지 소스

3.5. 프로세스 통합

CPACS를 이용한 형상생성, Centaur 소프트웨어를 이용한 CFD격자 생성, 인하우스 소프트웨어 umap3d 를 이용한 CFD 계산 과정을 RCE 상에서 모두 결합하였다.



- ① DOE 컴포넌트
 - Latin Hypercube 기법을 이용함.
- ② Evaluation Memory 컴포넌트
 - Restart 데이터 저장함. (DB 구성)
- ③ OPNAV Shapes 컴포넌트
 - 형상 생성 (Windows 10)
- ④ 격자/CFD 계산
 - Centaur-UMAP3D (Linux Cluster)

3.5.1. 통합 운영을 위한 DOE

결합한 형상-격자-CFD 컴포넌트를 이용하여 Design of Experiment 을 수행하였다. 이를 위하여 아래와 같이 추가적인 컴포넌트를 결합하였다.

3.5.1.1. Design of Experiment 컴포넌트

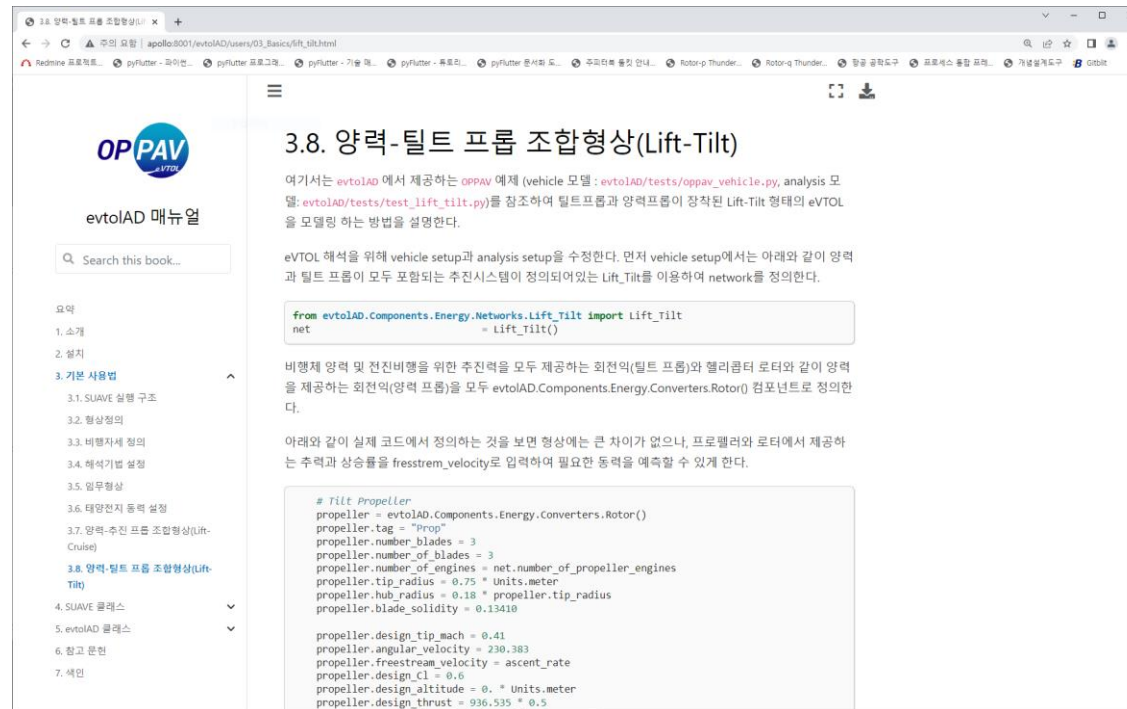
여기서는 형상-격자-CFD 계산연동을 시험하기 위하여 Design of Experiments 을 수행하였다. 이를 위하여 Design of Experiment 컴포넌트를 채택하였다.

출력은 항공기 주 날개 형상을 정의하기 위한 9가지로 파라미터로, 주 날개의 3 구간에서 Sweep 각, Twist 각, Dihedral/Anhedra 각도로 정의하였다. 이때 하한과 상한을 지정할 수 있다. 입력은 항공기의 6분력(CD, CY, CL, CMX, CMY, CMZ) 이다.



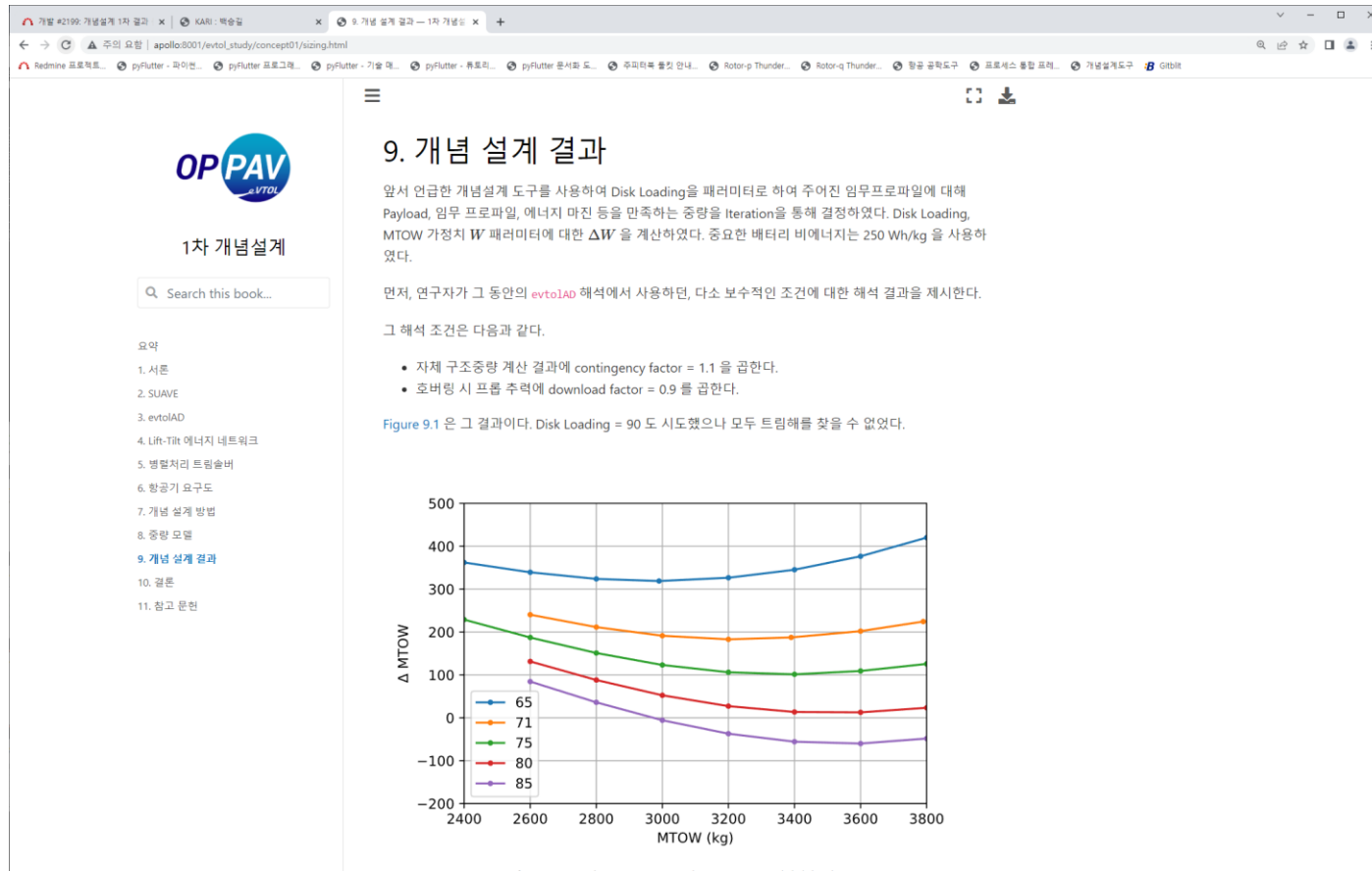
- **Concept Design Tool**

- Based on SUAVE(an open source concept design tool by Stanford Univ)
- Devised a component for the Lift-Tilt configuration



• The first sizing

- Based on OPPAV, one-seated vehicle, the sizing was made with the six-seater requirement.



- **The KARI plan, from 2022 to 2026, on the overall MDAO system development was explained**
- **KARI-DLR cooperations until now were explained.**
- **MDAO system design and concept design in 2022 were explained**
 - Requirements
 - System design
 - Interfaces
 - Component workflows

