

## FMI 표준을 활용한 관절형 로터/공력 연계시뮬레이션

백승길<sup>†</sup>, 박중용

한국항공우주연구원 회전익기연구팀

### Articulated Rotor/Aerodynamics Co-Simulation Using FMI Standard

Seung-Kil Paek<sup>†</sup>, Joongyong Park

Rotorcraft Research Team, Korea Aerospace Research Institute

**Abstract** : The purpose of this research is to develop co-simulation methodology of codes developed in different modeling and simulation environment. We develop aerodynamic FMU(Functional Mock-up Unit) meeting FMI(Functional Mock-up Interface) specification version2 utilizing Legacy FORTRAN aerodynamic code based on unsteady vortex lattice method. It is concluded that making FMU is possible utilizing Legacy code made in any language which can be compiled and linked with object in FMI API coded in C language. This paper explains QTronic's method of using FMU SDK(Software Development Kit) and suggestion for using FORTRAN properly. Finally, we make articulated rotor/aerodynamics co-simulation by integrating aerodynamics FMU and rotor FMU developed by Modelica.

**Key Words** : FMI, Functional Mock-up Interface, FMU, Functional Mock-up Unit, Co-Simulation, aerodynamics FMU, rotor FMU

## 1. 서 론

Functional Mock-up Interface(이하 FMI) 표준(<http://fmi-standard.org>)은 소스 코드가 아니라 컴파일된 모델을 서로 다른 기관/업체간에 교환하고, 다른 CAE tool과 연계하여 시뮬레이션(Co-Simulation) 하기 위한 표준인데, MODELISAR 프로젝트('08~'11) 내에서 독일 자동차 벤츠로 유명한 다일러 AG가 주도하여, 유럽에서 자동차 전장 소프트웨어 개발에 있어 표준화를 이끌어내기 위하여 개발하였다[1]. 유럽연합에서는 ITEA 2 (Information Technology for

European Advancement)라는 프로젝트가 진행되었는데 시스템 설계와 내장형 소프트웨어 설계 개선을 목적으로 하였고, MODELISAR 프로젝트는 그 한 과제로서, 자동차 소프트웨어에 대한 개방형 시스템 표준인 AUTOSAR (AUTomotive Open System Architecture) 표준을 지원하는 FMI를 개발하였다.

항공, 기계 등 기계분야는 다 마찬가지로, 자동차 분야에서도 한 시스템의 하위 컴포넌트 모델을 서로 다른 모델링 및 시뮬레이션 환경을 이용하여 개발하는 경우가 많다. 이 경우 각 프로그램은 서로 상호작용할 수 없으므로 전체 시스템을 시뮬레이션 하기 곤란하다. FMI는 이에 대한 해법인데, 상이한 환경의 시뮬레이션 모델이 서로 상호작용이 가능하게 해 주는 표준 인터페이스이다.

FMI 표준을 준수하는 한 개의 시뮬레이션 코드는

FMU(Functional Mock-up Unit)로 불리며 다양한 FMU로 구성된 큰 시스템에서 한 개의 단위기능을 수행하게 된다. 설계 단계에서 FMU는 특정단위의 시뮬레이터가 되지만, 나중에 실제 하드웨어가 사용가능할 때, FMU 대신 하드웨어를 연결하면 시뮬레이션은 자연스럽게 HILS(Hardware In the Loop Simulation)가 된다.

FMU는 단위 컴퓨터 상에서 연계 시뮬레이션 하기 위해서 사용할 수도 있지만, 분산환경에서 Client/Server 개념으로 사용할 수 있기 때문에 거의 모든 공학소프트웨어에서 활용할 수 있다.

현재는 모델리카 재단(<http://www.modelica.org>)에서 모델리카 표준과 함께 FMI 표준을 관리하고 있으며, 보쉬, 다임러, 다쏘, 지멘스, dSPACE, LMS, ANSYS, Maplesoft, QTronic, SIMPACK, IBM, DLR 등이 위원회에 참여하고 있다. 소프트웨어 CATIA, Dymola, ETAS, Matlab/Simulink, MSC ADAMS, MSC Easy5, IBM Rhapsody, LMS VirtualLab, ANSYS, SIMPACK 등에서 FMI를 지원하고 있으며, FMI를 활용하여 최단기간에 제품성능을 개량하고 설계를 최적화함으로써 개발기간을 최소화하는데 활용하려고 시도하고 있다. MSC의 경우 ADAMS와 Easy5를 연계해서 해석을 수행하는 기법을 제공하고 있다.

이러한 개념은 다물체동역학, 구조동역학, 공기역학이 결합된 회전익기의 비선형 동역학/공력 시뮬레이션이 가능한 해석 코드의 개발에 활용할 수 있다. 특히, 모델리카 재단에서 공개표준으로 관리하고 있는, 다중물리 시스템(Multi-Physics System) 모델링을 위한 프로그래밍 언어인 모델리카와 병행해서 사용할 경우 모델 기반 시스템 엔지니어링에 큰 발전이 예상된다. 모델리카 언어를 사용하면 쉽게 FMU를 개발할 수 있다. 최근 한국항공우주연구원은 모델리카 기술과 FMI 기술 개발을 적극적으로 진행하고 있다. 진재현과 백승길[2]은 항공 분야에서는 국내 최초로 모델리카 언어를 항공기 비행역학 시뮬레이션 프로그램을 개발하는데 적용하였다. 진재현과 박봉규[3]은 인공위성 비행역학 시뮬레이션 프로그램 개발에 모델리카 언어를 적용하였다. 박중용[4]은 모델리카 언어 및 모델리카 표준 라이브러리(MSL)를 이용하여 관절형 로터 FMU를 개발하고 시뮬레이션을 수행하였다. 또한 백승길과

위성용[5]은 한국항공우주연구원이 기존에 보유 중인 와류격자법에 기초한 비정상 공력코드를 개조하여 공력 FMU를 개발하였다.

본 논문에서는 참고문헌[4]과 참고문헌[5]에서 개발된 관절형 로터 FMU와 공력 FMU를 연계하여 시뮬레이션 연구 결과를 제시하고자 한다. 이것은 국내 항공 분야에서 최초로 시도하는 것이다.

## 2. Functional Mock-Up Interface

FMI는 2개의 규격을 갖는데, 첫 번째는 FMI for Model Exchange 이며 또 다른 하나는 FMI for Co-Simulation이다[6].

두 표준 모두 연계시뮬레이션 환경에서 두 개 이상의 모델을 연계시키는 것은 같지만, FMI for Model Exchange는 각 모델이 자체적인 Solver를 가지고 있지 않고, FMI for Co-Simulation은 자체 Solver를 갖는다는 차이가 있다. 각 서브시스템 간의 데이터 교환은 정해진 통신 시각에서 이루어지는데 불연속적일 수 있다. 통신 시각 사이의 시간 동안 각 서브시스템은 서로에 대해서 독립적으로 각각의 Solver를 이용하여 다음 통신 시각에 교환할 데이터를 계산한다. 서브시스템 간의 데이터 교환과 동기화는 마스터 알고리즘이 수행해야 한다. 서브시스템 간에는 직접적인 통신은 없다. 마스터가 서브 시스템과 통신하여 데이터를 받고 다른 서브시스템에 전달한다. 연계 시스템 시뮬레이션은 계층화된 Co-Simulation 환경에서 수행할 수 있는데, 계층구조의 각 수준에서 FMI for Co-Simulation을 적용한다. 마스터 알고리즘에 대해서는 표준이 정해져 있지 않기 때문에 다양한 방식의 마스터 알고리즘을 구현할 수 있다. 현재, FMI를 지원하는 소프트웨어는 나름의 마스터 알고리즘을 사용하며, 공개 소프트웨어도 구할 수 있다.

FMU는 “.fmu”라는 확장자를 갖는 zip 파일 형식의 파일이다. 이 파일 안에는 FMU 모델 정보와 함께 데이터 활용 및 교환을 위해 필요한 FMU 내부 정보가 XML 파일(modelDescription.xml) 형태로 들어 있다. 또한 Windows 운영체제환경에서 작동하는 동적 링크 라이브러리인 DLL 파일 또는 Linux, OSX 운영체제환경에서 작동하는 \*.so 파일을 포함한다. 따라서 한 개

의 FMU 파일로 다양한 운영체제를 지원할 수 있다. 이때 32bit 버전과 64bit 버전은 따로 생성해야 한다.

FMI 규격을 만족하는 것은 FMI 규격에서 제시하고 있는 API(Application Programming Interface)규격을 만족하는 C 언어 코드를 개발하고, 컴파일하여 동적 링크 라이브러리를 만드는 것이다. Fig. 1은 FMI for Co-Simulation 2.0의 상태 머신 다이어그램을 보여주고 있다. 여기에서 접두사 fmi2를 갖는 것은 모두 FMI API에 정의된 함수명이다. 이 함수를 모두 구현하여야 FMU가 완성되는데 공개된 FMU SDK (Software Development Kit)를 활용할 수 있다.

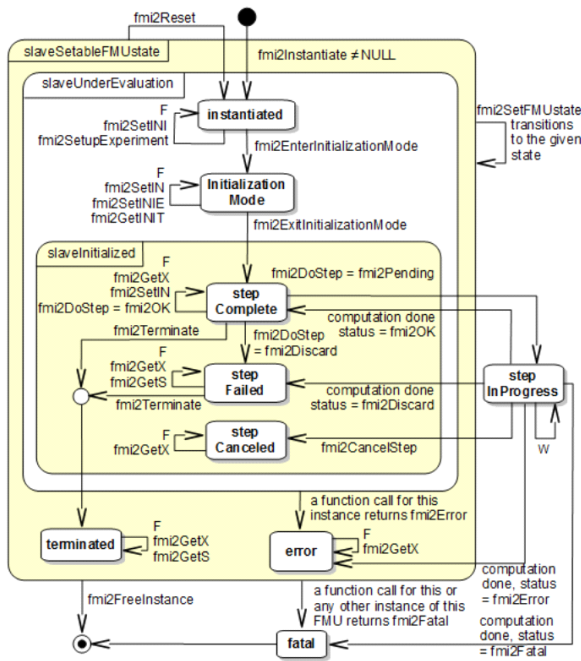


Fig. 1 FMI Co-Simulation State Machine

독일 QTronic社는 FMI for Model Exchange와 FMI for Co-Simulation에 대한 FMI 규격 1.0 및 2.0에 대해 FMU의 기본적인 사용방법을 시연하는 SDK를 공개했다. 이 FMU SDK는 다양한 이산형 및 연속형 FMU 모델에 대한 C 언어 소스파일을 제공하고 있어, FMU를 생성하거나 사용하는 응용프로그램을 개발하는데 시작점으로 활용가능하다[7].

Table 1에 제공되는 FMU 예제 별로 입력 변수, 출력 변수, 패러미터 변수, 지역 변수에서 실수형(real), 정수형(int), 문자열(string) 변수의 개수를 보여 주고

있다.

앞서 언급한 바와 같이, FMU를 사용하는 마스터 프로그램은 FMU의 입력, 출력, 패러미터 변수, 지역 변수를 알 수 있어야 하기 때문에 modelDescription.xml에 변수 정의를 한다.

Table 1 Variables Used in Each Example

name	inputs	outputs	parameters	local
bouncingBall	-	-	real=2	real=4
dq	-	-	real=1	real=2
inc	-	int=1	-	-
values	int=1 bool=1 string=1	int=1 bool=1 string=1	-	real=2
vanDerPol	-	-	real=1	real=4

Table 1에 있는 예제를 참고하여 자신이 개발한 FMU에 대한 modelDescription.xml를 작성할 수 있다. 특히, values 예제는 입력과 출력변수를 갖는 FMU로서 잘 이해하여야 한다. 한편, FMI규격에서는 Array변수에 대한 특별한 배려가 없기 때문에 모든 Array변수의 요소를 정의해야 한다. Array변수를 사용할 경우 수작업으로 modelDescription.xml를 작성하는 것은 시간이 많이 걸리고 오류발생 가능성이 높기 때문에 개별적인 작성 프로그램이 필요하다.

FMI 규격에서 요구하는 모든 C 함수가 구현되어 있는 fmuTemplate.c 파일은 가장 중요한 파일이다. FMU SDK의 모든 예제는 각 예제별로 차이가 있는 부분만 따로 모아서 한 개의 C 소스 파일과 헤더 파일에 저장해 놓았다. C언어 함수 구현에 있어 반드시 이해해야 하는 함수는 fmi2Instantiate와 fmi2DoStep인데, 각각 루프 실행 전 초기화를 하는 함수 및 각 루프에서 매번 실행하는 함수이다. 특히 fmi2DoStep에서는 미분계수를 이용한 오일러 적분 기법을 사용하여 시간적분을 수행하는 부분이 있다. 자신만의 시간적분을 구현하려면 이 부분을 제거하여야 한다.

### 3. 관절형 로터 FMU의 개발

#### 3.1 관절형 로터 모델 설계

공력 FMU와의 연계해석을 위하여 관절형 로터 FMU를 개발하였다[4]. 관절형 로터의 적절한 모델링

을 위하여 기존 상용해석코드에서 구성체계를 어떻게 해 놓았는지 알 필요가 있다. 이를 위하여 통상적인 회전익기에 대한 상용해석도구인 미국 ART(Advanced Rotorcraft Technology)사의 FLIGHTLAB®의 로터 시스템 모델링 체계를 Xanalysis[8]라는 tool을 통해 다소 어렵지만 분석이 가능하다. FLIGHTLAB® 소프트웨어가 제공하는 관절형 로터에 대한 기본모델을 분석하여 관절형 로터를 설계하였다[4].

### 3.2 로터 FMU의 작성

프랑스 Dassault社의 상용 모델리카 모델링 S/W인 Dymola[9]를 사용하되, 대부분 모델리카 표준 라이브러리를 활용하여 설계안에 따라 관절형 로터의 모델을 개발하였다[4]. Dymola는 공력 FMU를 사용할 수 있는 마스터 S/W로도 활용하였다.

참고문헌 4에서 작성한 관절형 로터 FMU를 nblades 패러미터를 이용하여 간단히 블레이드 개수를 조절할 수 있도록 하였고, 간단한 2엽 로터인 Caradonna & Tung[10] 모델(Fig 2 참조)과 동일한 블레이드 형상을 갖도록 수정하였다(Fig 3 참조). 컬렉티브 각( $\theta_0$ ), 사이클릭 각( $\beta_{1c}$ ,  $\beta_{1s}$ )과 로터 토오크를 입력으로 받아, 로터의 회전속도( $\omega$ ), 각 블레이드의 Azimuth 각, 피치각과 플랩각을 출력으로 생성한다. 2엽 블레이드에 대해 총 7개의 신호를 생성한다.

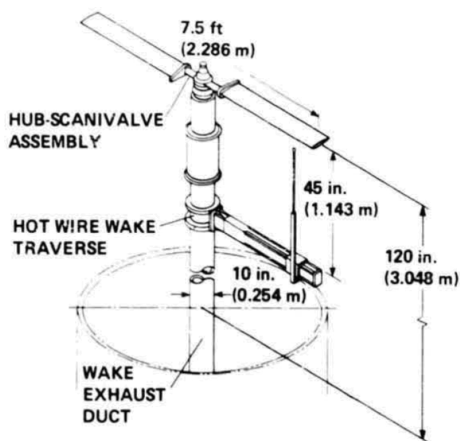


Fig 2 Caradonna and Tung's rotor test stand[10]

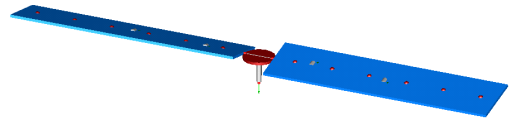


Fig. 3 Rotor FMU Shape

## 4. 공력 FMU의 작성

### 4.1 자유후류 공력 코드

앞서 언급한 바와 같이 백승길과 위성용은 참고문헌 5에서 헬리콥터 공력 성능 해석을 위해 개발된 Legacy 공력코드[11, 12]를 활용하여 비정상 공력 FMU를 개발하였다. 이 코드는 시간의 흐름에 따라 후류의 거동을 표현하는 비정상 후류 해석 기법을 사용하는데 다수의 문제 해석을 통하여 알고리즘의 신뢰도가 검증된 코드이다. 동 코드는 자체적인 시간적분 Solver를 가지고 있기 때문에 FMI for Model Exchange 가 아니라 FMI for Co-Simulation 규격을 적용하여야 한다.

동 코드는 비정상 와류격자법(Unsteady Vortex Lattice Method)을 기반으로 하고 있으며, 블레이드와 후류 부분을 구분하여 와류격자모델을 설정한다. 블레이드는 와류격자로 이루어진 양력면으로 모사하고, 후류는 trailed, shed 자유와선으로 모사된 와선(Vorticity Line)들의 다발로 모사한다. 시간에 대하여 수치적분을 적용하여 시간에 대한 와류의 움직임과 유동속도를 실시간으로 계산한다. 계산된 유동의 유도속도를 이용하여 물체에 가해지는 하중을 계산한다.

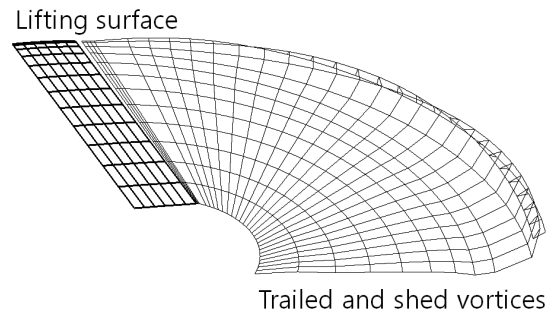


Fig. 4 Free Wake Structure using Vortex Lattice

### 4.2 포트란 코드와 C 코드의 결합

동 공력코드는 모두 포트란 90언어로 작성되어 있다. 앞서 언급한 것과 같이, FMI API 규격은 C 언어 기준이다. 연구 초기에, 포트란 90언어의 소스 코드를 컴파일 및 링크하고, FMI API에 맞추어 작성한 C 언어 코드가 포트란 함수 또는 서브루틴을 호출할 수 있을 것으로 예상하였고 연구 결과, 이것을 확인하였다. 따라서 FMI API 2.0 규격에 맞추어 작성된 C 언어 코드를 일종의 Wrapper 역할을 하게 하고 C 언어 이외의 다른 언어로 작성된 기존의 Legacy 코드는 모두 활용가능하다[5].

포트란 함수를 C 언어 코드에서 호출할 때, Name Mangling에 주의해야 한다. 이것은 C 언어 코드에서 호출하는 포트란 함수의 이름이 사용한 포트란 컴파일러에 따라 소스 파일 내에서 정의된 이름과 달라지는 것을 말한다. 본 연구에서는 C 컴파일러로는 Microsoft Visual Studio 10.0을, 포트란 90 컴파일러로는 Intel Fortran Compiler 15.0을 사용하였다. 공개 소프트웨어인 gcc 및 gfortran을 사용하여도 된다. Intel Fortran Compiler의 경우 함수명이 모두 대문자가 되며, gfortran을 사용할 경우 함수명 앞에 “\_”(underscore) 문자가 붙는다. 포트란 서브루틴 호출은 C 언어에서 함수 호출 같이 처리하면 된다. 단, 변수의 포인터를 포트란 서브루틴의 입력으로 넣어야 한다.

자유후류 공력 FMU는 C 언어 코드에서 입력으로 공기속도( $u, v, w$ )와 로터의 회전속도( $\omega$ ), 각 블레이드의 Azimuth 각과, 피치각과 플랩각을 입력으로 받은 후 포트란 함수를 호출하여 전달하고, 마찬가지로 C 언어 코드에서 블레이드 하중을 계산하는 포트란 함수를 호출하여 블레이드 하중을 출력으로 받는다.

본 연구에서는 사용되는 관절형 로터 FMU에 맞추어 하중은 전체 블레이드에서 발생하는 공력하중의 총합을, 블레이드 루트 위치를 기준으로 계산하였다.

fmi2Instantiate C함수에서 변수를 초기화하는 포트란 서브루틴을 호출하고, fmi2DoStep C함수에서는 시간 스텝을 한 단계 전진하는 포트란 서브루틴을 호출하게 하였다.

### 5. 연계 시뮬레이션

Dymola S/W를 사용하면, FMU 파일을 import하여 모델리카 모델처럼 사용할 수 있다. 공개 소프트웨어인 OpenModelica는 아직 FMI for Co-Simulation 버전 2.0은 지원하지 않고 FMI for Model Exchange 버전 2.0만을 지원하므로 현재로는 Dymola S/W만이 유일한 대안이다. Fig. 5는 최종적인 로터/공력 연계 시뮬레이션의 Design Model이다.

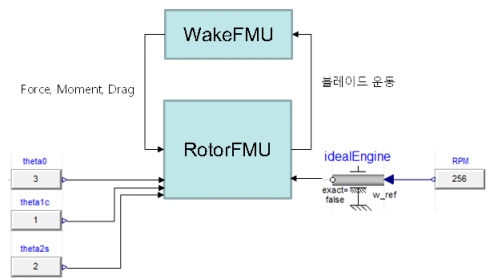


Fig. 5 Interconnected RotorFMU and WakeFMU

페루프 시뮬레이션에 앞서 개루프 시뮬레이션을 Fig. 6과 같이 수행하였다.

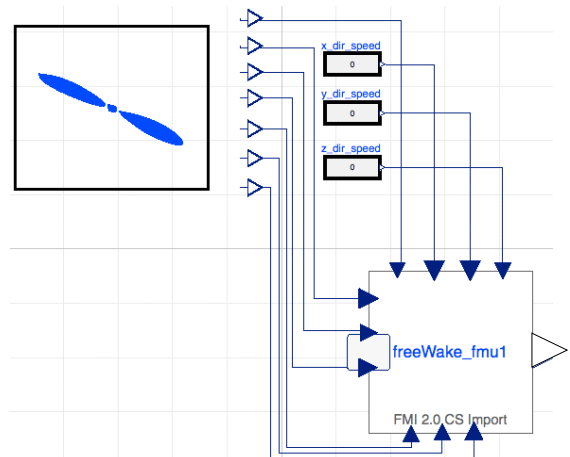


Fig. 6 Simulation Model Combining FreeWake and Rotor FMU

로터 FMU가 회전각속도  $\omega = 24.88 \text{ rad/s}$ , 익단 마하수  $M_{tip} = 0.439$ , 콜렉티브각  $\theta_0 = 8^\circ$ , 사이클릭 각  $\theta_c = 0^\circ$  조건을 공력 FMU에 부여하면서 연계 시뮬레이션을 수행하였고, 추력계수를 Fig. 7과 같이 시험치[9]와 비교하였다. 시험치와 잘 일치하는 결과를 얻었다.

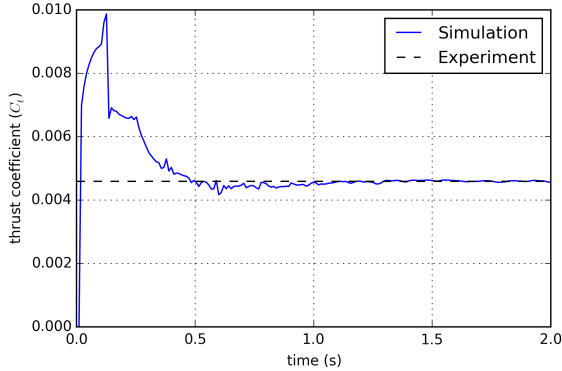


Fig. 7 Simulation Result, Thrust Coefficient

로터 FMU가  $\omega = 24.88 \text{ rad/s}$ ,  $M_{tip} = 0.439$ ,  $\theta_0 = 8^\circ$ ,  $\theta_c = 0.5^\circ$  조건을 공력 FMU에 부여하면서 연계 시뮬레이션을 수행하였다. Fig. 8과 Fig. 9는 각각 블레이드 피치각 및 추력계수이다. 블레이드 피치각은 로터 FMU에서 전달한 값을 공력 FMU에서 저장한 값으로서, 로터 FMU에서 공력 FMU로의 통신이 제대로 이루어지고 있음을 알 수 있다.

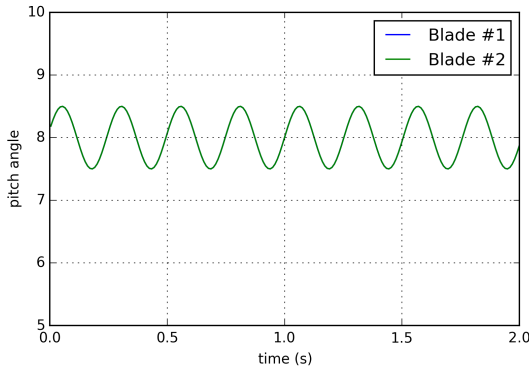


Fig. 8 Blade pitch angle in Aerodynamics FMU

Fig. 9는 0.5도 진폭의 블레이드 피칭운동을 고려할 경우 적절하게 계산된다고 판단된다.

## 6. 결론

서로 다른 모델링 및 시뮬레이션 환경을 이용하여 개발한 코드의 연계 시뮬레이션 기법을 개발하기 위하여 비정상 와류격자법에 기반한 Legacy 포트란 공력

코드를 이용하고 FMI for Co-Simulation 버전 2.0 규격을 만족하는 공력 FMU를 개발하여, 관련 기술을 습득하였다.

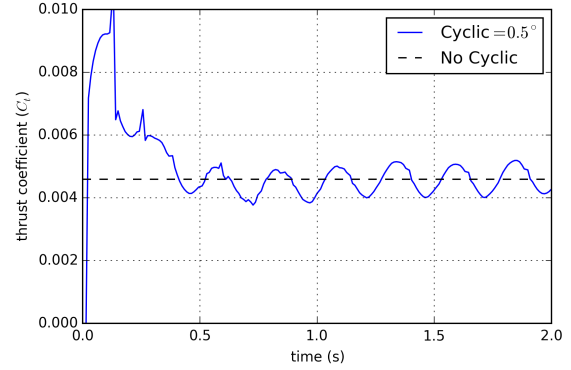


Fig. 9 Simulation Result with 0.5 cyclic angle, Thrust Coefficient

이를 통해 C 언어로 작성된 FMI API에서 호출가능한 오브젝트로 컴파일 및 링크 가능한 모든 언어로 작성한 Legacy 코드를 이용하여 FMU 제작이 가능함을 보였다. 이와 함께 QTronic社의 FMU SDK 활용 방법과 포트란 언어 사용 시에는 특히 Name Mangling에 유의하여야 함을 제시하였다. 모델리카 언어로 개발한 로터 FMU와 결합하여 로터/공력 연계시뮬레이션을 수행하였다. 이를 통해 공력 FMU가 적절히 작동함을 보였다.

## 후기

본 연구는 한국항공우주연구원 시드사업 “비선형 회전익기 시스템에 대한 연계 시뮬레이션 핵심기술 개발” 연구 결과 중 일부임.

## 참고 문헌

- [1] T. Blochwitz, M. Otter, J. Akesson, M. Arnold, C. Clauß, H. Elmqvist, M. Friedrich, A. Junghanns, J. Mauss, D. Neumerkel, H. Olsson, and A. Viel, “Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models,” Proceedings of the 9th International Modelica

Conference, pp.173-184, 2012.

- [2] J.-H., Jin and S.-K. Paek, "Development of Acausal Flight Simulation Program and Application to Preliminary Design," Journal of the Korean Society of Mechanical Technology, vol. 17, no. 3 pp. 497-504, June 2015.
- [3] J.-H., Jin and B.-K. Park, "Development of Integrated Simulation Program for Artificial Satellite Operations by Modelica," Journal of The Society for Aerospace System Engineering, vol. 9, no. 3 pp. 39-46, Sep 2015.
- [4] J. Park, "Development of M&S Model for Helicopter Rotor Blades Dynamic Analysis using Modelica", Proc. of KOSSE Fall Conference 2015, Seoul, Korea, Oct. 2015.
- [5] S.-K. Paek and S.-Y. Wie, "Development of Freewake Aerodynamic FMU complying FMI Standard", Proc. of KSAS Fall Conference 2015, Jeju, Korea, Nov. 2015
- [6] Anonymous, Functional Mock-up Interface for Model Exchange and Co-Simulation (Version 2.0), fmi-standard.org, Sweden, 2014
- [7] <https://www.qtronic.de/en/fmusdk.html>
- [8] Anonymous, FLIGHTLAB X-Analysis User Manual, Advanced Rotorcraft Technology, Inc., USA, 2011
- [9] Anonymous, 2015, *Dymola - Dynamic Modeling Laboratory User Manual*, Dassault Systems AB, Sweden.
- [10] Caradonna, F. X. and Tung, C. 1982, "Experimental and Analytical Studies of a Model Helicopter Rotor in Hover," *NASA TM-81232*
- [11] Chung, K. H., Kim, J. W., Ryu K. W., Lee K. T. and Lee D. J., 2006, "Sound Generation and Radiation from Rotor Tip-Vortex Paring Phenomenon", *AIAA Journal*, Vol. 44, No. 6, pp. 1181-1187.
- [12] S.-Y. Wie, and D.-J. Lee, "An Analysis of BVI Unsteady Rotor Aerodynamics using Unsteady Panel and Time-Marching Free Wake," Journal of the Korean Society for Aeronautical and Space Science,

vol. 37, no. 4, pp. 329-336, April 2009.

## 저자 소개

### 백 승 길

1991년 서울대 항공공학과 졸업. 1993년 한국과학기술원 항공우주공학과 석사. 1998년 동 대학원 박사. 2006년까지 T-50 초음속 훈련기 개발을 위해 공탄성 엔지니어. 그 이후 한국항공우주연구원에서 한국형 헬리콥터 프로그램(KHP) 업무 수행. 현재 회전익기연구팀에서 책임연구원으로 재직 중. 관심분야는 항공기 공탄성, 능동진동제어 및 모델기반 시스템 모델링 및 분석.



### 박 중 용

1991년 서울대 항공공학과 졸업. 1993년 서울대 항공우주공학과 석사. 2003년 아주대학교 시스템공학과 박사. 2003년까지 현대우주항공, 고등기술연구원, SETechnology에서 헬기 및 시스템공학 프로젝트 다수 수행. 그 이후 한국항공우주연구원에서 한국형 헬리콥터 프로그램(KHP) 업무 수행. 현재 회전익기연구팀에서 책임연구원으로 재직 중. 관심분야는 모델기반 시스템엔지니어링 및 시스템 안전성.

