DOI: http://dx.doi.org/10.14248/JKOSSE.2015.11.2.075
ISSN 1738-480X (Print)
ISSN 2288-3592 (Online)
Journal of the Korea Society of Systems Engineering

Modelica를 이용한 헬기 로터 블레이드 동적해석 M&S 모델 개발

박중용*
한국항공우주연구원

Development of M&S Model for Helicopter Rotor Blades Dynamic Analysis using Modelica

Joongyong Park*
Korea Aerospace Research Institute

Abstract : This paper describes modeling & simulation(M&S) model for dynamic analysis of helicopter rotor blades. Simulation model is developed using Dymola tool which implements the open source language - Modelica. Modelica is appropriate for developing multibody dynamic analysis model. To develop an M&S model efficiently, model based systems engineering(MBSE) is applied. Some diagrams such as requirement diagram, block definition diagram and sequence diagram etc. are drawn to capture the concept of M&S model. This activity is done utilizing the open source tool – Papyrus.

Key Words : Modelica, Modeling & Simulation, MBSE, SysML

* 교신저자 : Joongyong Park, parkjy@kari.re.kr

1. 서 론

시스템엔지니어링의 주요 기능은 개발하고자 하는 시스템을 이해하는 것이다. 고객이 원하는 시스템에 대해서 비기술적 용어를 이용해서 서술하게 되면 이것을 기술적 용어로 바꾸어 시스템 요구사항 및 규격서를 작성한다. 이와 같은 과정을 거치면서 시스템 개발자는 고객이 원하는 시스템을 이해하게 된다. 하지만 글 위주로 작성된 문서 형태의 규격서는 여러 사람이 같은 관점으로 이해하는데 뒤떨어지고 현재와 같이 컴퓨터의 성능이 뛰어난 시대에는 맞지 않는다. 그래서 현대 엔지니어링에서는 모델을 개발하여 상호이해의 가능성을 높인다. MBSE(Model Based Systems Engineering , 모델기반 시스템엔지니어링)는 개발 대상인 시스템이 복잡해지면서 더욱더 필요성이 증폭되었다. 거대해진 시스템의 규모와 복잡성뿐만 아니라 시스템 내/외부와의 인터페이스를 모델을 통하지 않고서는 표현하기 힘들게 되었기 때문이다.

MBSE의 또 다른 장점 중의 하나는 M&S(Modeling & Simulation, 모델링&시뮬레이션)와의 연계가 효율적으로 수행된다는 점이다. MBSE를 위해 구축되는 모델은 다양한 방법으로 만들어지지만 시스템엔지니어링 분야에서는 2007년 버전 1이 발표된 SysML 언어를 구현하는 도구를 통해 작성하는 방법이 정착되어가고 있다. SysML의 다양한 다이어그램들은 도구를 통해 그 자체가 시뮬레이션되기도 하지만 시스템 설계 시 필요한 각종 엔지니어링 해석 모델의 입력 자료가 되거나 직접 연계되어 엔지니어링 해석 도구에서 시뮬레이션된다. 시스템엔지니어링이 문제를 이해하는 것에 초점이 맞춰있다면 시뮬레이션은 이해한 것이 적절한지를 표현하는데 초점이 맞춰있다.

이러한 엔지니어링 동향을 관찰한 Sanford Friedenthal과 Roger Burkhart는 SysML이 앞으로 발전해나가야 할 방향을 제시하면서 그 중 여러 전문 엔지니어링 분야의 엔지니어링 도구와 SysML 언어와 지원도구가 통합되어야하며, 이를 위해서 표준 적용 프로그래밍 인터페이스(API: Application Programming Interface)를 지원해야 한다고 주장한 바 있다. 한 예로서 모델 기반의 내용을 관리 및 통합하는 플랫폼인 Open Services for Lifecycle Collaboration(OSLC)과 Functional Mockup Interface(FMI)와의 통합 기능이 SysML에 필요하다고 제시하였다[1].

본 논문에서는 헬기 로터 시스템의 동적 해석을 위한 시뮬레이션 모델을 수립하는데 있어 SysML을 구현하는 오픈 소스(Open Source) 도구인 Papyrus를 이용해 개념을 수립하고 이를 바탕으로 시뮬레이션 작성 언어인 Modelica를 구현하는 도구인 Dymola를 이용해 시뮬레이션 모델을 구축하여 해석을 수행하는 과정과 결과를 보여준다. 헬기 로터 시스템의 동적 해석 이론이나 결과 보다는 시스템엔지니어링 관점에서 SysML을 이용한 개념 수립과 시뮬레이션 기법이 복잡한 시스템의 설계와 해석에 유용하게 활용되었음을 강조하였다.

2. SysML과 Modelica

2.1 SysML

SysML은 시스템엔지니어링을 적용하기 위한 광범위한 목적의 모델링 언어로서 다양한 영역의 복잡한 시스템의 규격, 분석, 설계, 검증 및 확인을 지원한다. 여기에서 시스템은 하드웨어, 소프트웨어, 정보, 프로세스, 인력 그리고 설비 등을 포함한다[2]. SysML은 OMG(Object Management Group)에서 표준을 관리하고 있는 소스가 공개된 언어로서 2007년에 처음 도입되어 2012년 6월에 발간된 버전 1.3이 최신 버전이다. SysML은 시스템의 다음과 같은 측면을 보여줄 수 있다[3].

- 구조적 구성, 연관관계, 분류
- 기능 기반/메시지 기반/상태 기반 거동
- 물리적 성질/성능에 가해진 제약사항
- 거동, 구조, 제약사항 간의 할당
- 요구사항과 요구사항, 설계 요소, 시험 케이스 간의 관계

다이어그램	특성	UML과의 관계
Requirement Diagram		SysML 고유
Activity Diagram	거동 표현	UML 수정
Sequence Diagram	거동 표현	UML과 동일
State Machine Diagram	거동 표현	UML과 동일
Use Case Diagram	거동 표현	UML과 동일
Block Definition Diagram	구조 표현	UML 수정
Internal Block Diagram	구조 표현	UML 수정
Package Diagram	구조 표현	UML과 동일
Parametric Diagram	구조 표현	SysML 고유

[Figure 1] SysML Diagram Taxonomy

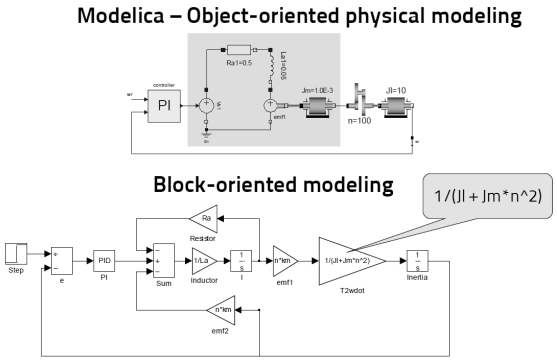
SysML은 총 9개의 다이어그램을 가지고 있다. 각 다이어그램의 특징에 대해서는 참고문헌 2를 참고한다.

SysML을 구현할 수 있는 상용도구는 많이 판매되고 있다. 본 연구에서는 상용도구가 아닌 무료로 배포되는 오픈 소스 도구인 Papyrus를 사용했다. Papyrus외에 Modelio도 사용 가능하지만 Modelio는 SysML의 일부 다이어그램 작성이 지원되지 않는다. 두 개의 도구 모두 UML을 기본적으로 지원하며 추가로 SysML을 지원하는 방식이다. Papyrus는 https://eclipse.org/papyrus/에서 프로그램을 다운로드받을 수 있고 관련 정보를 얻을 수 있다.

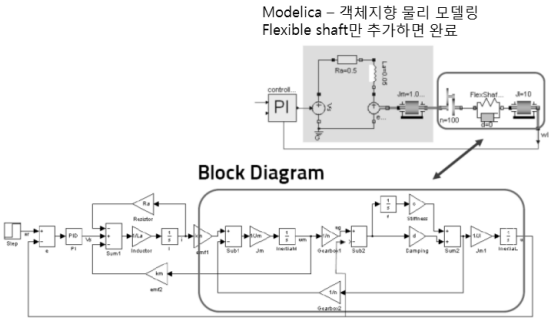
2.2 Modelica

Modelica는 다중물리 시스템(Multi - Physics System)의 모델링을 위한 프로그래밍 언어로서 기계/화학/전자/물리/인간공학 등 여러 분야를 모델링 할 수 있다[4]. Modelica는 Modelica 재단(http://www.modelica.org)에서 표준을 관리하고 있는, 소스가 공개된 언어로서 2000년에 처음 도입되어 2014년 7월에 발간된 버전 3.3 revision 1이 최신 버전이다. Modelica의 주요 특징은 다음과 같다.

- 객체지향 언어로서 콤포넌트를 조립하여 모델링 가능



[Figure 2] Object oriented Modelica[5]



[Figure 3] Acausal Modeling of Modelica[5]

- 상미분대수방정식에 기반한 비인과적(acausal) 모델링 가능

객체지향 언어를 이용한 모델 구축의 장점을 엔지니어링의 시뮬레이션 도구로 많이 사용하는 Matlab/Simulink와 비교한 Figure 2로 알 수 있다. Simulink의 모델 작성을 위해서는 공학 이론에 기반하여 블록을 끼워넣어야 하나 Modelica는 표준 라이브러리에 있는 컴포넌트를 손쉽게 배치하여 해석이 가능하다.

Figure 3에서 보듯이 Modelica에서는 인과성을 따지지 않고(acausal) flexible shaft라는 구성품 1개만 라이브러리에서 끌어다가 추가하면 모델 보완이 되지만 Matlab/Simulink에서는 같은 효과를 얻기 위해 더 많은 개수인 10개의 블록을 엔지니어링 지식을 기반에 두고 구성해서 삽입해야 하는 어려움이 있다.

Modelica를 구현하는 상용도구로는 Dymola, LMS Amesim, MapleSim, MOSILAB, SimulationX 등

이 있으며, 무료로 배포되는 오픈 소스 도구로서 OpenModelica가 있다. 본 연구에서는 상용도구인 Dassault Systems사의 Dymola(Dynamic Modeling Laboratory)를 사용하여 모델링 및 시뮬레이션을 수행했다.

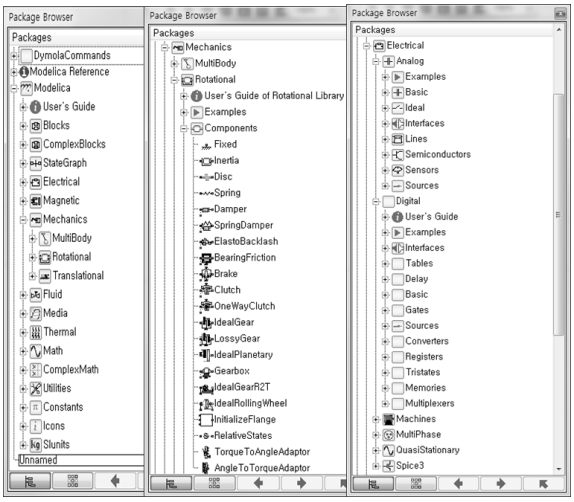
3. 헬기 로터 동적해석 시뮬레이션

3.1 헬기 동적해석 도구

헬기는 수직이착륙이 가능한 비행체로서 회전익기(rotorcraft)로 분류된다. 고정익기가 고정되어 있는 날개를 통해 양력을 얻는데 반해 회전익기는 로터, 틸트 로터, 덕티드 팬과 같은 회전하는 시스템을 통해 양력을 얻는다. 회전익기는 고정익기에 비해 공기역학적 현상이 복잡하고 공기력이 구조물과 간섭을 일으켜 엔지니어링 해석을 통해 성능을 예측하는 것이 쉽지 않다. 헬기의 성능해석을 위해 주로 사용되는 도구로는 CAMRAD II와 Flightlab과 같은 상용도구와 회사나 연구소 또는 대학에서 자체적으로 개발한 인하우스 코드(In-House code)가 있다. 그러나 상용도구는 소스 코드를 수정할 수 없기 때문에 일반적인 형태의 헬기 성능해석을 손쉽게 할 수 있으나 복합형 헬기나 덕티드 로터와 같은 회전익기의 성능해석에는 한계가 있다. 인하우스 코드는 해석 결과의 신뢰성을 보장받기 힘들고 복잡한 동력학 수식이 코드에 반영되어 있어 개발자 외에는 다양한 목적의 해석을 위한 코드 수정이 쉽지 않은 상황이다.

Modelica는 기존의 헬기 성능해석용 상용 도구와 인하우스 코드의 단점을 극복하고 손쉽게 성능해석과 시뮬레이션을 할 수 있는 대안이다. Modelica의 표준 라이브러리에서 제공하는 기계 파트, 전기전자 파트, 신호 등의 컴포넌트를 이용해 로터 시스템 모델을 구성하고 공기력을 FMI(Functional Mockup Interface)를 통해 적용시키면 Modelica에 내재되어 있는 동적 해석 코드가 작동하여 시뮬레이션이 수행되기 때문이다.

Modelica는 동적 시뮬레이션 프로그램으로 사용



[Figure 4] Modelica standard library[5]

되어 온 기존의 도구들(예: ADAMS)에 비해 확장성이 뛰어나다. Modelica는 script를 조작하는 방식으로 용도에 맞게끔 기계/화학/전자/물리/인간공학 등의 다양한 시스템을 모델링 할 수 있고 누구나 필요로 하는 모듈을 개발하여 제공할 수 있다. 그리고 ADAMS와 같은 도구가 특정 분야에 대해 정밀한 해석 결과를 산출하는 것과는 달리 Modelica를 구현하는 도구는 시스템 수준의 모델링을 추구한다.

본 연구에서는 2개의 블레이드로 구성된 간단한 형태의 로터 시스템을 모델링하되, 공기력은 임의의 형태로 적용할 수 있도록 설정하였다. 실제 헬기 성능해석에서 사용되는 자유후류(free wake)를 FMI를 통해 로터 모델과 연계해서 시뮬레이션 하는 과정과 결과는 Paek 외의 연구를 참조한다[6].

3.2 헬기 동적해석 시뮬레이션 모델의 SysML

시뮬레이션 모델을 만들기 위한 개념수립 과정에서 SysML을 구현하는 오픈 소스 도구인 Papyrus를 활용하였다. SysML의 다양한 다이어그램들 중에서 본 연구에서 작성한 다이어그램은 Requirement 다이어그램, Sequence 다이어그램, Block Definition 다이어그램, Parametric 다이어그램이다. 각종 다이어그램 작성의 기초가 되는 시뮬레이션 모델에 대한 요구사항은 헬기 동적해석 상용도구인 Flightlab을 분석하여 수립하였으며 주요 요구사항은 다음과 같다.

- 블레이드는 유연하지 않은 강건체(Rigid Body)
- 블레이드를 구성하는 구조 세그먼트 6개
- 블레이드 질량, 무게중심, 관성모멘트 수치
- 컬렉티브 피치(Collective pitch) 각, 싸이클릭 피치(Cyclic pitch) 각 입력 가능
- 공기력은 임의의 형태로 적용

컬렉티브 피치는 헬기의 상승과 하강을 조종할 수 있도록 양력의 크기를 가감하는 블레이드 길이 방향 축을 기준으로 하는 비틀림 각도를 뜻하고 싸이클릭 피치는 헬기가 좌우상하로 움직일 수 있도록 로터 평면이 기울어지게끔 블레이드에 가하는 각도를 뜻한다.

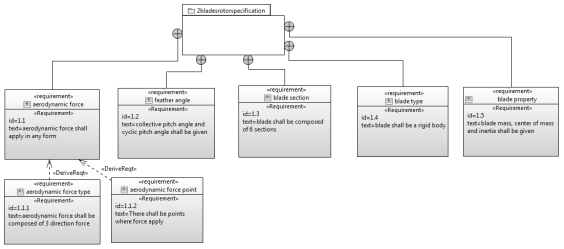
작성한 다이어그램들 중 Requirement 다이어그램과 Block Definition 다이어그램 일부를 소개한다. Requirement 다이어그램은 시뮬레이션 모델의 개념을 수립하는데 활용되는 요구사항을 체계적으로 정리하는데 도움을 주었고 Block Definition 다이어그램은 시뮬레이션 모델의 아키텍쳐 수립을 지원하였다.

3.2.1 Requirement 다이어그램

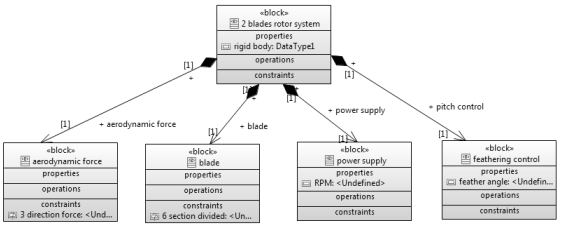
Flightlab을 분석해서 도출한 요구사항을 다이어그램으로 도식화하였다. 최상위 수준의 요구사항은 다섯 항목으로 분류되었는데 블레이드 관련 항목이 3개이고 나머지는 공기력과 블레이드 피치 각과 관련된 항목이다. Figure 5에서 보듯이 공기력은 x, y, z 세 방향의 힘으로 나누어 작용하되 공기력이 작용되는 곳을 지정할 수 있어야 한다는 요구사항이 최상위 요구사항으로부터 유도되었음을 알 수 있다. 이와 같이 요구사항은 상위 요구사항에서 하위 요구사항으로 유도할 수 있다.

3.2.2 Block Definition 다이어그램

시뮬레이션 모델을 구성하는 컴포넌트를 Block Definition 다이어그램을 통해 구축했다. 최상위 수준의 요구사항은 강건체의 블레이드 두 개로 구성된 로터 시스템이고 요구사항을 바탕으로 유추한



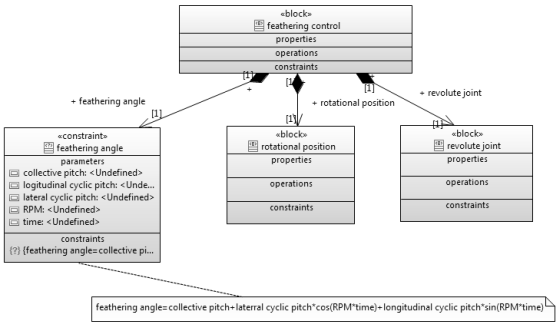
[Figure 5] Requirement Diagram



[Figure 6] Rotor system Block Definition Diagram

필요 컴포넌트는 Figure 6과 같이 크게 네 가지로 분류할 수 있다. 블레이드는 6개의 섹션으로 이루어짐을 알 수 있고 공기력은 세 가지 방향의 힘으로 분해된다. 동력 공급(power supply)은 로터가 회전할 수 있도록 동력을 공급하는 컴포넌트로서 Modelica 모델에서는 로터의 회전수를 직접 입력하는 방식으로 동력을 공급하는 방안을 선택하였다. 마지막으로 페더링 조정(feathering control)은 블레이드가 회전하면서 컬렉티브 피치 각과 싸이클릭 피치 각을 조종하는 컴포넌트이다. 실제 헬기에는 피치 각을 조정하는데 사용되는 스와쉬 플레이트(Swashplate), 피치 로드(Pitch rod), 피치 혼(Pitch horn) 등의 구성품이 있지만 Modelica 모델에서는 실제와 똑같이 복잡하게 만들 필요가 없으므로 파라미터 간의 관계식으로 feathering 각을 구현하도록 한다.

네 가지 컴포넌트에 대해 다시 Block Definition 다이어그램을 작성하여 아키텍쳐를 좀 더 구체화했다. Figure 7은 그 중에서 페더링 조정 컴포넌트의 하부 계층을 보여준다. 블레이드가 회전을 하면서 컬렉티브 피치 각과 싸이클릭 피치 각이 조종되어야 하는데 이를 위해서 Modelica에서 회전 위치를 만들어주는 컴포넌트인 Modelica.Mechanics.Rotational.Sources.Position과 실제 회전각을 표현할 수 있는



[Figure 7] Feathering control Block Definition Diagram

회전 조인트인 Modelica.Mechanics.MultiBody.Joints.Revolute를 선정하여 모델을 작성하면 된다. 두 개의 컴포넌트는 Modelica의 표준 라이브러리에서 제공하는 것으로 Position은 rotational source에 속해 있는 컴포넌트이고 조인트는 Multibody에 속한 컴포넌트임을 알 수 있다. 라이브러리의 다양한 컴포넌트 중에서 모델이 해석해야 할 거동을 잘 표현할 수 있는 컴포넌트를 선정하는 것이 중요하다. 다이어그램 왼쪽에 constraints로 표현된 것은 실제 feathering 각이 여러 파라미터와 어떤 관계식을 갖는지 보여준다.

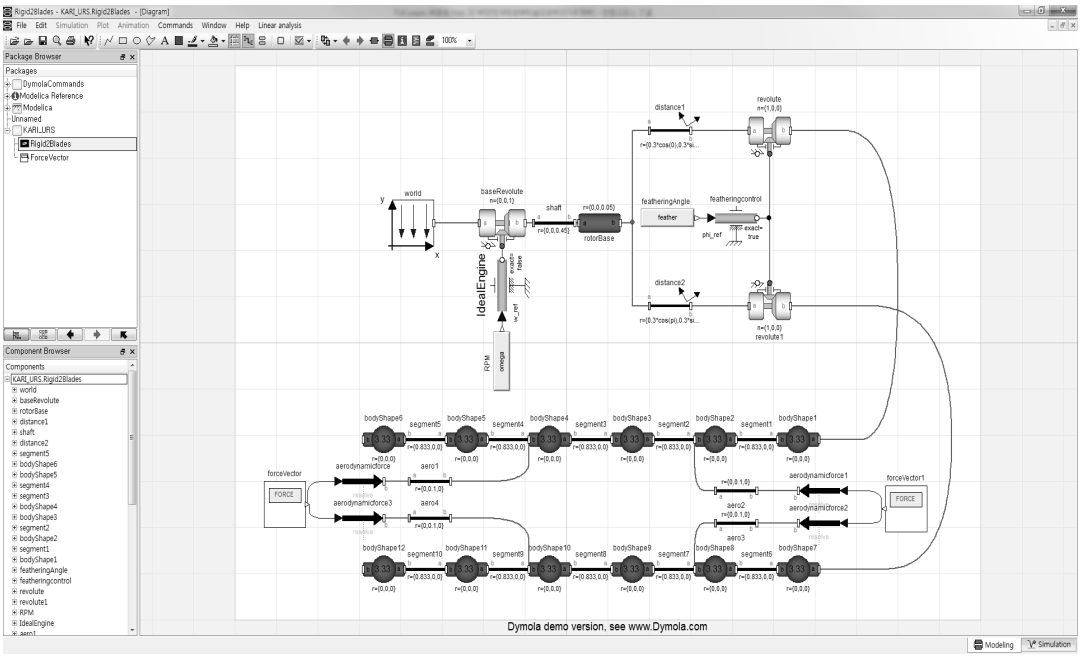
관계식이 길어 constraints에서 잘 볼 수가 없으므로 다이어그램의 하단에 주석을 달아 관계식을 명확하게 볼 수 있게 했다.

3.3 헬기 동적해석 시뮬레이션 Modelica 모델

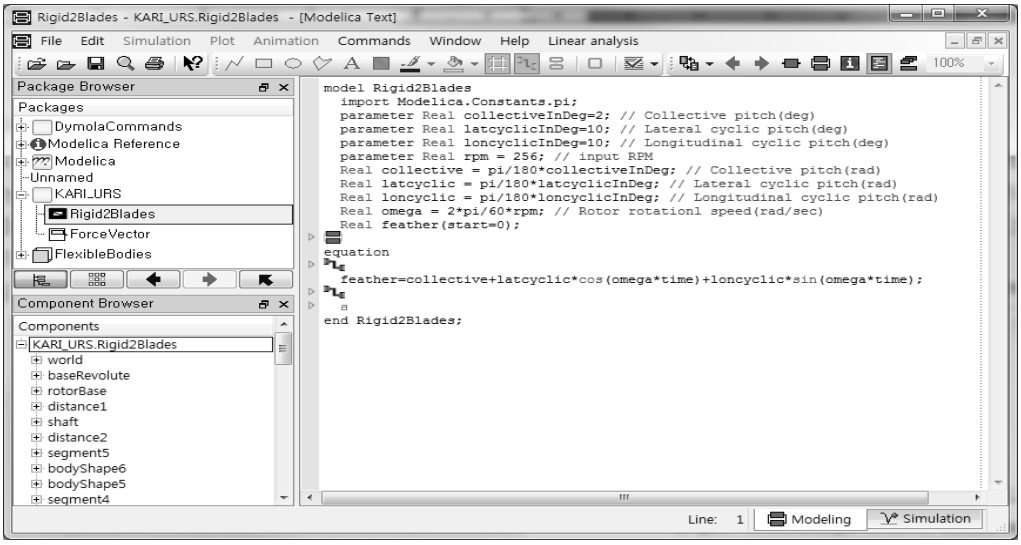
헬기의 동적해석 시뮬레이션 모델을 구성하는데 필요한 컴포넌트는 3.2의 요구사항과 Block Definition 다이어그램으로부터 식별하여 다음과 같이 Modelica의 표준 라이브러리에서 선정한다.

- 블레이드의 구조 세그먼트 표현 컴포넌트는 Modelica.Mechanics.MultiBody.Parts.BodyShape 사용
- 블레이드의 구조 세그먼트 간의 연결 컴포넌트는 Modelica.Mechanics.MultiBody.Parts.FixedTranslation 사용
- 공기력은 Modelica.Mechanics.MultiBody.Forces.WorldForce 사용
- 공기력의 크기, 로터의 회전수, 컬렉티브 피치와 싸이클릭 피치 양은 Modelica.Blocks.Sources.RealExpression 사용

Figure 8은 컴포넌트를 배치하고 컴포넌트 간에



[Figure 8] Rotor Modelica Model Diagram window



[Figure 9] Rotor Modelica Model Text Window

커넥터를 연결하여 완성한 헬기 로터의 시뮬레이션 모델의 다이어그램이다. 상단부는 로터 회전축을 표현한 것으로서 로터를 회전토록하고 블레이드의 피치 각을 입력하는 부분이다. 하단부는 두 개의 블레이드를 표현한 것으로서 각 블레이드는 요구사항대로 6개의 세그먼트로 구성되어 있음을 알 수 있다. 블레이드의 양단에는 공기력을 적용하는 컴포넌트가 구성되어 하나의 블레이드 당 2개의 지점에서 공기력이 입력된다.

시뮬레이션 모델을 컴포넌트 배치가 아닌 Modelica 언어로 직접 코드를 작성해서 완성할 수도 있다. Figure 9는 Figure 8을 코드로 전환한 것으로서 Dymola를 사용해 다이어그램을 그리면 코드가 자동으로 생성되며 반대로 코드를 생성하면 다이어그램이 그려진다.

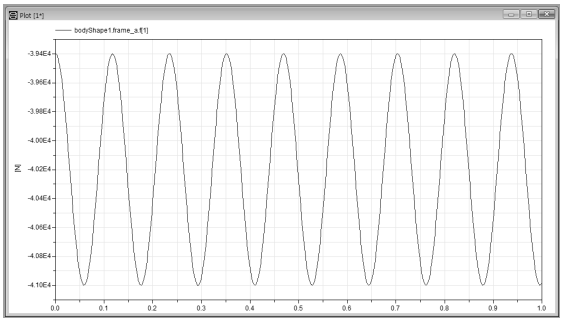
그러나 다이어그램을 통해 표현하는데 한계가 있는 경우가 많기 때문에 Dymola 고급 사용자가 되려면 코드를 작성해서 모델을 만들 수 있어야 한다. 코드 앞부분에 관련 변수들을 정의하고 이어 필요한 수식과 선정된 컴포넌트 성질 및 컴포넌트 간의 연결 관계가 나열된다.

Figure 10은 완성된 모델을 대상으로 시뮬레이션 한 결과를 애니메이션으로 표현한 것의 한 장면이다. 가운데 빨간색 원판을 축으로 로터가 회전을

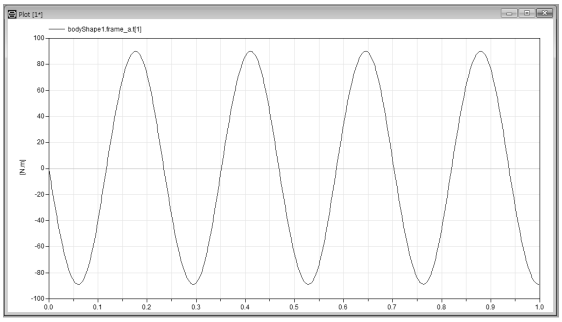
[Figure 10] Rotor Modelica Model Animation

하며 좌/우 블레이드가 블레이드 길이 방향의 축을 기준으로 비틀어져 피치 각이 제대로 입력되었음을 알 수 있다. 블레이드 중간 중간에 보이는 짙은 색 점은 질량이 집중된 질점을 뜻하며 옅은 색 점은 공기력이 적용되는 지점이다.

다음 Figure 11과 Figure 12는 1초 동안 시뮬레이션한 결과로서 시간에 따라 회전축에 제일 가까운 질점에 작용하는 원심력과 블레이드 길이 방향 모멘트를 그래프로 표현한 것이다. x축이 초를 단위로 하는 시간, y축이 각각 Newton, Newton · Meter를 단위로 하는 원심력과 모멘트이다. 이와 같이 한 번 시뮬레이션을 수행하게 되면 힘, 가속도, 속도, 변위 등 다양한 해석 결과를 원하는 지점에서 그래프로 그려 검토할 수 있다.



[Figure 11] Centrifugal force of Bodyshape1



[Figure 12] Blade Spanwise Direction Moment of Bodyshape1

이와 같이 구축된 시뮬레이션 모델은 설계하고자 하는 로터 블레이드가 어떤 동특성을 가지고 있는지 사전에 파악할 수 있도록 지원하므로 복잡한 로터 시스템의 설계 및 해석에 도움을 줄 수 있다. 본 연구에서는 Modelica를 이용하여 로터 시스템의 M&S 모델을 구축할 수 있는 가능성과 기법을 보여준 것으로 실제 로터 시스템의 동적 해석은 정교한 모델이 필요하다.

4. 결 론

헬기 로터 블레이드의 동적해석을 위한 시뮬레이션 모델을 Modelica를 이용해 구축하였다. Modelica로 작성된 M&S 모델을 개발해야 하는 시스템으로 설정하고 시스템 개념 수립을 SysML의 몇 가지 다이어그램을 이용해 수행하였다. 이 때 Modelica는 상용도구인 Dymola, SysML은 오픈 소스 도구인 Papyrus를 활용했다.

SysML의 Requirement 다이어그램과 Block Definition 다이어그램 등을 작성하여 M&S 모델에 대한 요구사항을 구체화하면서 모델의 개념을 수립하였다. 이 과정에서 개발해야 할 M&S 모델의 이해도가 상승하였다.

Modelica는 수식 기반의 객체지향 모델링 언어로서 기존의 모델링 언어에 비해 손쉽게 모델을 작성할 수 있음을 확인하였다.

추가 연구사항으로는 개발한 로터 블레이드 동적해석 모델을 보완하여 좀 더 복잡한 형태의 블레이드 모델을 개발하고 공기력과의 연계도 원활하게 수행되도록 할 예정이다. 공기력이 연계된 모델을 통해 산출되는 동적 해석 결과치인 로터에 작용하는 힘과 모멘트는 헬기 개념/기본설계와 성능해석의 입력자료로서의 기능을 할 수 있다. 또한 본 연구에서는 블레이드를 강체로 가정했으나 유연한 보로 모델링하는 연구와 더불어 헬기의 비행역학모델을 추가로 개발할 계획이다. 그리고 SysML을 본 연구와 같이 제한적으로 사용하지 않고 완전한 규격 모델이 될 수 있도록 할 계획이다. 그리고 3D 설계 프로그램인 CATIA의 모델을 바탕으로 Modelica 모델을 완성하는 모듈을 이용해서 Modelica 모델을 효과적으로 구축하고 가시적으로 높은 품질의 시뮬레이션 결과를 볼 수 있도록 할 예정이다.

후 기

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

References

1. Sanford Friedenthal and Roger Burkhart, “Evolving SysML and the System Modeling Environment to Support MBSE”, INSIGHT Vol 18. Issue 2, 2015.
2. Object Management Group, OMG Systems Modeling Language (OMG SysML™), Version 1.3, 2012.

3. Sanford Friedenthal, Alan Moore and Rick Steiner, A Practical Guide to SysML, Elsevier, p29, 2012.
4. Modelica Association, Modelica® – A Unified Object-Oriented Language for Systems Modeling Language Specification, Version 3.3 Revision 1, 2014.
5. Hubertus Tummescheit, "Modelica and FMI – Two disruptive open standards for model-based systems engineering", NDIA SE Modeling & Simulation Committee Meeting, 2014. 2. 11.
6. Seungkil Paek and Seong-yong Wie, "Development of Freewake Aerodynamic FMU complying FMI Standard", KSAS Fall Conference, 2015. 11. 18.