

技術論文

J. of The Korean Society for Aeronautical and Space Sciences 41(6), 488-494(2013)

DOI:<http://dx.doi.org/10.5139/JKSAS.2013.41.6.488>

Fly-By-Wire 헬리콥터 비행제어법칙 자동생성코드 개발 및 검증 프로세스

안성준*, 김종섭*, 조인제*, 허진구**

A Development and Verification Process of Auto Generated Code for Fly-By-Wire Helicopter Control Law

Seong-jun Ahn*, Chong-sup Kim*, In-je Cho* and Jin-goo Heo**

Korea Aerospace Industries*, Defense Agency for Technology and Quality Sacheon Center**

ABSTRACT

The control law design and analysis environment of the FBW helicopter system have been developed using model base design method. The model based design is generally used in a aircraft, unmanned aerial system and automobile as well as rotorcraft development. The model based design provides many advantages such as development risk and schedule reduction using simulation and autocode generation. This paper describes a development of process for verification and validation of auto generated code for FBW helicopter flight control law. And this process is applied for Fly-By-Wire Helicopter Development Project. The results of functional test for auto generated code meet several specific requirements.

초 록

전자식 비행제어시스템(Fly-By-Wire Flight Control System)에 적용되는 헬리콥터의 제어법칙 설계 및 해석은 모델기반 설계기법(Model Based Design)을 사용하여 개발되고 있다. 모델기반 설계기법은 회전익뿐만 아니라 고정익, 무인항공기, 자동차 산업에서도 널리 사용되고 있다. 모델기반 설계기법은 핵심기술인 시뮬레이션과 자동생성코드를 이용하여 개발기간 단축 및 개발리스크 감소라는 장점을 제공한다. 본 논문에서는 FBW 헬리콥터 비행제어법칙의 자동생성코드에 대해서 체계적인 기능검증을 위한 프로세스를 개발하였다. 그리고 지식경제부 과제로 진행하고 있는 FBW 헬리콥터 개발에 적용하여 비행제어법칙 자동생성코드의 기능 검증을 수행 결과, 요구 조건을 충족하는 것으로 확인할 수 있었다.

Key Words : Model Based Design(모델기반 설계기법), Auto Generated Code(자동생성 코드), FBW Flight Control System(전자식비행제어시스템)

1. 서 론

모델기반 설계 기법(Model Based Design)은 항공기, 자동차 및 방위산업 등 다양한 산업분야

에서 널리 사용되는 설계 기법이다. 전자식 비행제어시스템이 적용되는 헬리콥터의 제어법칙 설계 및 해석은 모델기반 설계 기법을 사용하여 개발되고 있다. 모델기반 설계기법은 제어법칙의

† Received: February 21, 2013 Accepted: May 31, 2013

* Corresponding author, E-mail : tjdwns6@koreaero.com<http://journal.ksas.or.kr/>

pISSN 1225-1348 / eISSN 2287-6871

설계와 동시에 시뮬레이션 기능을 통하여 설계된 제어법칙의 요구기능 만족여부를 확인하는데 기존의 개발방식에 비하여 시간절약이 가능하다[1]. 뿐만 아니라 설계된 제어법칙을 별도의 구현절차를 거치지 않고 자동생성코드기능을 이용하여 설계된 제어법칙을 탑재프로그램에 적합한 형태의 C코드로 제공한다. 이러한 기능들은 모델기반 설계기법의 주요 장점들로서 새로운 시스템을 개발하는데 필요한 일정 및 개발 리스크를 감소시키는데 큰 공헌을 하였다. 해외의 방위산업체들 또한 모델기반설계기법을 통하여 개발일정과 개발비용을 절약한다[2]. 자동생성코드는 추가적으로 엔지니어가 코딩을 하면서 발생할 수 있는 오류나 기능구현의 불편함을 없애주었지만 제어법칙 구현을 위한 모델(SBM: System Build Model)과 모델기반으로 생성된 자동생성코드의 검증이라는 문제점을 내포하고 있다[3].

본 논문에서는 MATLAB/SIMULINK 기반으로 설계한 모델 형태로 제어법칙을 설계하고, 설계된 제어법칙 모델과 모델에서 생성된 소스코드에 대해서 검증이 필요한 테스트 항목에 대해서 식별하였다. 그리고 식별된 모델을 기반으로 모델 커버리지(model coverage), 일치성 검증, 소스 코드 상에서 코드 커버리지(code coverage), 무결성, 코딩규칙 그리고 상호 유사성(similarity) 검증 등의 항목을 활용하여 자동생성코드의 신뢰성을 검증하였다. 그리고 개발된 절차에 따라서 실제 전자식 헬리콥터 개발 과제에 적용하여 모델과 소스 코드에 잠재적 결함을 내포하고 있음을 확인하였고, 이를 정형화된 검증 절차에 따라 모델을 수정하여 요구도를 만족함을 확인하였다.

II. 자동생성코드 목표

자동생성코드의 궁극적인 목표는 모델 형태로 구현된 제어법칙을 추가적인 검증 없이 비선형 시뮬레이션 환경에서 유사성 검증과 같은 기능 확인 절차만 거친 후, 탑재체에 반영하는 것이다. 그러나 자동생성코드에 신뢰성을 부여하기 위해서는 개발초기에 모델과 자동생성코드의 다양한 검증이 필수적이다. DO-178C/DO-331 모델기반 설계 및 검증에서도 모델 커버리지, 시뮬레이션, 스탠더드, 테스트 환경 등에 대해서 가이드하고 있다. 그러나 모델과 자동생성코드와의 일치성에 대한 검증에 대해서는 개발업체의 자체 검증 도구나 개발업체의 상용 검증 도구에 맡겨놓은 상태이다[4]. 자동생성코드의 궁극적인 목표를 달성하기 위해서는 개발초기에 모델검증, 자동생성코

드와의 일치성 검증, 생성된 코드의 무결성 검증 그리고 자동생성코드 생성에 대한 정형화된 기준이 필요하다.

III. 자동생성코드 개발

자동생성코드 개발을 위해 요구도에 따라 제어법칙 모델을 생성하고, 시뮬레이션을 통해 개발된 알고리즘의 기능동작 확인 후 자동생성코드를 생성하게 된다. 이러한 이상적인 자동생성코드 개발과정은 개발초기에 적용하기에는 위험성이 크다. 모델 초기 개발과정에서는 기본 구성뿐만 아니라 옵션 설정과 가이드라인을 통한 안정화를 필요로 한다. 설정된 옵션과 사용되는 블록에 따라 자동생성코드의 기능적 오류를 발생시키거나 가독성 및 효율성이 저하되기 때문에 개발자들 간에 공통적으로 사용할 수 있는 자동생성코드 도구를 개발하였다[5].

자동생성코드 가이드라인은 모델 개발 시 지켜야 할 사항들과 모델에 포함해야 하는 정보를 정의하였다. 모델에서 사용되는 블록과 신호들의 속성, 이름규약, 신호타입 그리고 블록의 색깔까지 모델옵션에서 설정된 부분을 제외한 모델 구현에 필요한 부분이 포함되어 있다. 또한 자동생성코드를 검증하면서 발생하는 오류나 기능은 같지만 다른 형태의 코드를 생성할 수 있는 블록들에 대해서 정의를 하고 기본 블록들을 사용할 수 있도록 하고 있다. 가이드라인 기반에서 개발된 자동생성코드는 기능적인 검증만을 수행 후 탑재 소프트웨어에 반영될 수 있는 신뢰성을 확보하는 기반이 된다. 선진항공업체에서도 이와 같은 가이드라인을 자동생성코드 개발에 적용하고 있다.

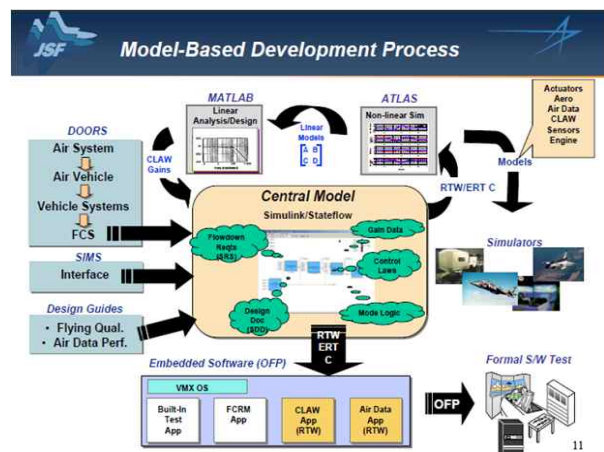


Fig. 1. Model-Based Development Process of F-35 Joint Strike Fighter

모델 옵션설정은 탑재시스템의 하드웨어 특성 및 탑재소프트웨어의 아키텍처에 따라서 달라진다. 전자식 헬리콥터는 32비트 연산이 가능한 DSP (Digital Signal Processor)칩을 사용하며, 비행제어 비행제어법칙 개발 및 시험은 Fig. 1과 같이 JSF(Joint Strike Fighter)의 모델기반 설계기법 프로세스와 유사한 형태로 개발되기 때문에 탑재소프트웨어에 반영되는 자동생성코드 또한 임베디드 C 형태로 생성한다.

최적화된 템플릿 파일은 자동생성코드의 인터페이스를 위해 사용된다. 비선형 6-자유도 시뮬레이션, 조종성평가 시뮬레이션 그리고 탑재소프트웨어에 공통으로 자동생성코드가 적용될 수 있도록 템플릿 파일을 개발하였다.

IV. 자동생성코드 검증

자동생성코드를 탑재시스템에 적용하기 위해서 다양한 방법의 검증방법이 필요하다. 모델 검증, 소스코드의 정적시험, 단위시험, 소프트웨어 비선형시뮬레이션 시험, 하드웨어 통합시험, 통합 비선형 시뮬레이션 시험 그리고 비행시험 등의 시험을 거쳐야 한다[6].

모델 검증을 위해서 모델 스타일 가이드 준수, 모델의 커버리지 그리고 모델의 시뮬레이션 결과와 자동생성코드의 시뮬레이션 결과를 비교함으로써 모델과 코드의 일치성을 검증할 수 있다. 그리고 소스코드의 검증을 위해서 자동 생성

된 코드의 요구하는 수준의 커버리지, 코딩규칙 및잠재적 결함이나 런타임 오류가 포함되어 있는지에 대한 무결성을 검증해야 한다. 그리고 모델 시뮬레이션과 비선형 6-자유도 시뮬레이션과도 유사성이 있음을 검증해야 대상 시뮬레이터에서의 상사성을 유지할 수 있다. Fig. 2와 같이 전자식 헬리콥터 제어법칙 자동생성코드 검증을 위하여 자동생성코드에 대한 검증 절차를 개발하여 적용하였다.

4.1. 모델 검증

전자식 헬리콥터 제어법칙 개발 및 해석은 기본적으로 MATLAB/SIMULINK® 환경에서 개발된다[7]. SIMULINK를 이용하여 모델이 설계 및 구현이 되고 SIMULINK Coder®에 의해서 소스코드가 생성된다. 개발 초기에 자동생성코드는 모델과 다른 시뮬레이션 결과를 보이곤 한다. 복합블록의 사용, 알고리즘 오류, 초기화 설정 등의 여러 가지 요인으로 인하여 모델과 자동생성코드의 일치성 비교를 필요로 하는 경우가 있다. 전자식 헬리콥터 제어법칙을 구현함에 있어서도 문제가 발견되었고 개발 및 검증절차에 의해서 문제를 해결하였다.

모델 커버리지 검증(MC/DC)

제어법칙 모델을 기반으로 모델 커버리지 테스트를 수행하여 모델의 구조적 결함여부를 판단하였다[3][8]. 판단 지표로는 DO-178 Level A에 준하는 변경 조건/결정(MC/DC: Modified Condition

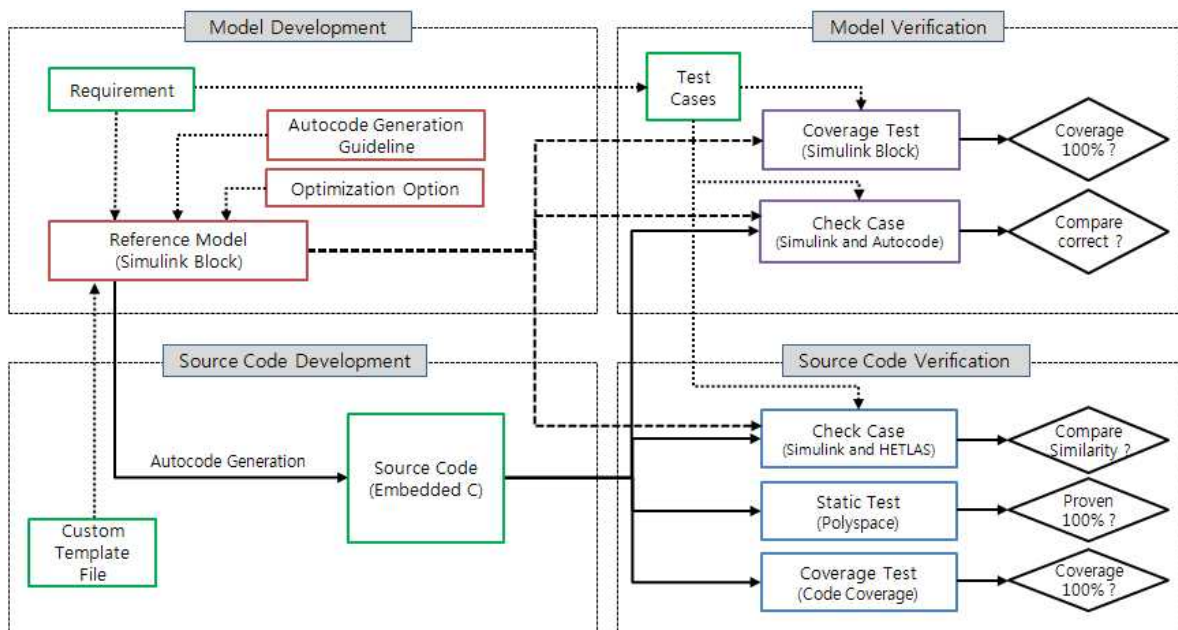


Fig. 2. Auto Code Development and Verification Process for FBW Helicopter Control Law

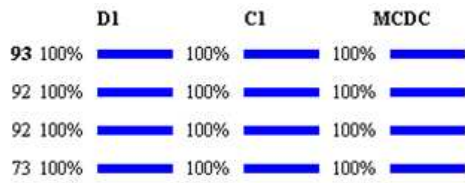


Fig. 3. Model Coverage Verification Result

Decision Coverage)을 만족 여부를 검증하였다. 만족하지 못하는 구조의 모델에 대해서 모델 수정을 통하여 100% 모델 커버리지를 만족시켰다. 수정이 필요한 모델에 대해서는 가이드라인에 명시된 함으로 해서 추가적인 에러를 방지하도록 하였다. 총 48개의 조건 커버리지(condition coverage), 124개의 결정커버리지(decision coverage), 22개의 변경 조건/결정 커버리지(MC/DC)을 만족시켰으며, 모델분석을 통한 자동테스트케이스 8개와 추가적인 수동테스트케이스 4개를 생성하였다. Fig. 3과 같이 최상위 모델의 92개에 대하여 커버리지를 만족함을 보이고 있다. 기본적인 네드코드를 추출해낼 수 있었으며 코드로 변환후의 커버리지를 재확인하기 위해 소스코드 커버리지를 추가적으로 수행하였다.

일치성 검증 테스트(SIMULINK vs S-Function)

자동생성코드의 시뮬레이션 결과가 SIMULINK 결과와 다른 경우가 발생한다. 이러한 경우는 개발자의 의도와 다르게 SIMULINK에서 시뮬레이션을 수행하기 위한 내부설정 부분에 대해서 자동생성코드에 반영이 되어 있지 않은 경우가 많다. 그래서 같은 시뮬레이션 환경에서 자동생성코드를 MATLAB 내부함수인 S-Function을 이용하여 모델과 자동생성 코드가 기능적으로 일치함을 검증하였다. 시뮬레이션 상사성을 최종적으로 만족시키기 위한 테스트 케이스를 활용하여 검증하였으며 자세한 설명은 상사성 검증테스트에서 하도록 하겠다. Fig. 4는 횡측계단(Step) 명령에 따른 10초간 선형모델 시뮬레이션 결과이다. 각각 4축에 대하여 동일한 시뮬레이션을 통해 결과가 일치함을 확인하였다.

본 개발과제에서도 변수 초기화 과정 및 적분기에서 문제가 발생하여 초기화 함수추가, 적분기를 MATLAB에서 제공하는 모델이 아닌 이산적분기를 만들어서 사용하였다. 틀에서 제공하는 복합기능의 블록을 사용하기 보다는 기본 블록들을 사용하여 개발자가 구현한 블록을 사용해야 자동생성코드의 이상 현상에 대한 정비가 쉽고 가독성을 높일 수 있다.

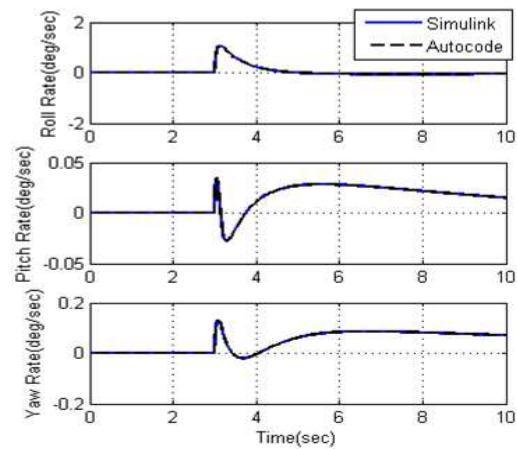


Fig. 4. Comparison of Model and Auto Code Simulation Result(Lateral Step)

4.2 소스코드 검증

소스코드 커버리지 검증(MC/DC)

소스코드의 커버리지 테스트는 소프트웨어 개발에서 널리 수행되는 테스트방법이다. 여러 과제에서 사용되어 검증된 상용툴을 이용하여 전자식 헬리콥터 자동생성코드의 코드 커버리지 검증하였다. 코드 커버리지 검증을 통하여 코드의 구조적 결함 및 네드코드를 제거하였다. 코드 커버리지의 지표는 DO-178 Level A를 만족하도록 Table 1을 참조로 하여 설정하였다.

전자식 헬리콥터 자동생성코드의 구문 커버리지(statement coverage)는 259개, 분기 커버리지(branch coverage)는 111개, 변경 조건/결정 커버리지(MC/DC)는 28개이며, 이를 만족하도록 테스트케이스를 생성하였다. 기본적인 자동테스트 케이스와 매뉴얼 테스트케이스를 사용하여 커버리지를 만족시켰다.

Table 1. DO-178 Software Level and Coverage

Level	Coverage	Description
A	MC/DC	Level B + 100% Modified Condition Decision Coverage
B	DC	Level C + 100% Decision Coverage
C	SC	Level D +100% Statement(or Line) Coverage
D	-	100% Requirement Coverage
E	-	No Coverage Requirement

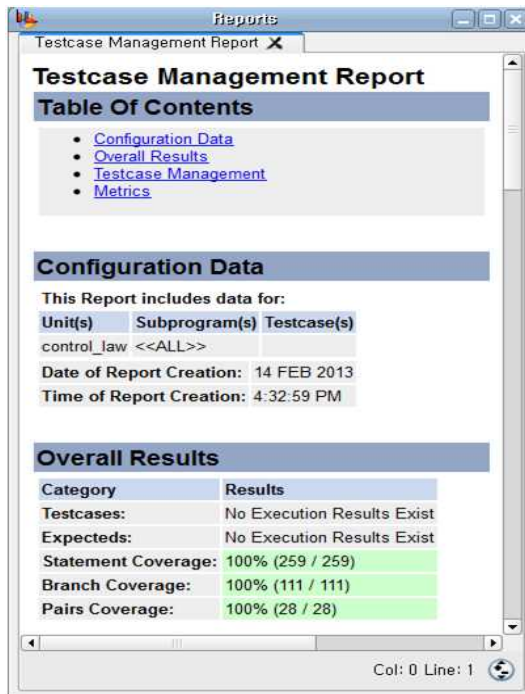


Fig. 5. Code Coverage Verification Result

그리고 모델 커버리지 테스트에서는 발생되지 않았던 소스코드에 사용되는 지역변수로 인한 변경 조건/결정 커버리지(MC/DC) 저하가 발생하였으며, 이를 해결하기 위하여 모델을 수정하였다. 원인은 모델에 사용된 MATLAB에서 제공하는 복합블록과 대수루프(algebraic loop)에러의 복합요인이었으며, 복합블록은 사용자 개발블록으로 변경, 대수루프 에러는 딜레이 블록을 사용하여 커버리지를 만족시킬 수 있었다. 모델 커버리지와 소스 커버리지의 테스트 목적이 다름을 이해하고, 그 목적에 따라 적절한 비중의 테스트 수행이 필요하다. Fig. 5와 같이 자동생성코드의 각 항목별 커버리지를 100% 만족하는 것을 확인하였다.

소스코드 정적테스트(코딩규칙, 무결성)

정적테스트 도구를 이용하여 자동생성코드의 기본적으로 코딩규칙 확인과 런타임 오류, 오버플로 등의 잠재적 결함을 내포하고 있지 않음을 확인함으로써 해서 자동생성코드의 무결성을 검증할 수 있다. 전자식 헬리콥터 자동코드에 적용된 코딩규칙은 유사과제에 적용된 MISRA-C(Motor Industry Software Reliability Association)를 수정을 하여 사용하였다. 총 141개의 규칙 중에 55개가 적용되었으며 54개를 만족하였다. 1개의 불만족 룰(Rule 12.6)에 대해서는 모델수정을 통하여 만족시킬 예정이다.

Table 2. Run-Time Error Checks

Run-Time Error	Description
UNR	Unreachable Code
OBAI	Out of Bounds Array Index
ZDV	Division by Zero
NIV,NIVL	Non-Initialized Variable
OVFL	Scalar and Float Overflow
IRV	Initialized Return Value
SHF	Shift Operations
IDP	Illegal Dereference Pointer
COR	Correctness Condition
NIP	Non-Initialized Pointer
ASRT	User Assertion
NTC	Non-Termination of Call
K_NTC	Known Non-Termination of Call
NTL	Non-Termination of Loop
STD_LIB	Standard Library Function
ABS_ADDR	Absolute Address
IPT	Inspection Points

자동생성코드를 실행하지 않고 정적분석을 통해 소스코드의 특정 런타임 오류가 없을 증명하였다. 코드 실행경로를 분석하고 변수범위에 대한 동시 접속 경로를 미리 파악하여 수학적으로 에러가 없음을 증명하는 방법을 활용하였다. 분석을 통해 증명된 런타임 에러는 Table 2와 같다. Fig. 6과 같이 전자식 헬리콥터 자동생성코드는 17가지의 런타임 에러에 대하여 99% 무결함을 검증하였으며, 나머지 1%에 대해서는 MATLAB에서 제공되는 Look-Up Table 함수에 대한 오버플로 에러로서 향후 제어법칙 형상 확정시까지 배포되는 형상에 대해서 검증을 수행할 예정이다.

본 검증에 사용된 상용 검증도구는 제어법칙 모델과 소스코드와 링크기능을 제공하여 발생한 런타임 에러를 바로 수정 가능한 기능을 제공하고 있다. 자동생성코드의 특성상 기계가 만들어내는 소스코드이며 모델이 복잡해질수록 코드의 양이 늘어난다. 그래서 사람의 손으로 따라가며 이해하기가 거의 불가능하기 때문에 에러가 발생할 가능성이 커진다. 개발초기에 이러한 문제를 해결하는데 코드의 정적테스트는 필수적이라 할 수 있다. 위에서 수행된 커버리지 테스트, 코딩규칙 그리고 무결성 테스트에서 발생한 결함에 대해서는 결함이 발생한 내용을 자동생성코드 가이드라인에 반영하고, 가이드라인의 내용에 따라 제어법칙 모델을 수정 후에 다시 자동생성코드를 개발하는 절차로 개발을 수행한다.

Proven	Green	Red	Grey	Orange
100.0%	172	0	0	0
100.0%	17	0	0	0
100.0%	13	0	0	0
100.0%	13	0	0	0
100.0%	10	0	0	0
99.6%	761	0	0	3
96.6%	86	0	0	3
99.4%	1072	0	0	6

Fig. 6. Polyspace Run-Time Checks Result

이와 같이 자동생성코드의 모델과 소스코드의 무결성 시험이 개발초기에 과도할 만큼 필요한 이유는 자동생성코드의 궁극적인 목적인 기능적 검증절차만을 수행한 후에 탑재소프트웨어에 반영하기 위함이다. 제어법칙 모델이 확정된 이후에는 제어법칙의 유지보수를 위해서 기존에 사용된 모델을 활용이 가능하다. 자동생성코드는 블록의 속성 및 종류 그리고 모델의 옵션에 따라 일정한 템플릿을 바탕으로 코드가 생성되기 때문에 동일한 형태의 코드를 생성할 수 있다. 자동생성코드 가이드라인을 바탕으로 생성된 코드에 대해서도 가이드라인에 제시된 형태로 매번 같은 결함이 제거된 코드를 생성 할 수 있다.

상사성 검증 테스트(SIMULINK vs HETLAS)

제어법칙 개발환경중 하나인 헬리콥터 비선형 6-자유도 시뮬레이션도구 (HETLAS : Helicopter Trim, Linearization and Simulation)에 제어법칙 자동생성코드를 탑재하여 모델의 시뮬레이션 결과와 상사성을 비교함으로써 MATLAB 환경이 아닌 대상 시뮬레이터 환경에서 자동생성코드의 상사성을 검증할 수 있었다. 상사성 검증을 위한 조건은 Table 3과 같이 제자리 비행영역에서 각 축에 대한 계단(Step) 명령 및 가진(Doublet) 명령을 가하여 나오는 응답특성을 비교하였다.

상사성 검증 조건은 헬리콥터 선형모델을 기반으로 설계된 비행제어법칙이 CONDUIT 평가 환경에서 ADS-33E-PRF 규격을 만족하는 것을 확인하기 위해 사용되었던 테스트 조건 선정하였다.[9] 그리고 검증시 조종명령의 형태는 비행제어법칙에서 설계된 명령형태(response type)를 기준으로 선정하여 이를 검증할 수 있도록 하였다. Fig. 7 과 8의 결과와 같이 상사성을 갖는 것을 확인하였다. 시뮬레이션 결과는 종축과 횡축으로 가진(Doublet) 명령을 주어 On-Axis의 결과이며 종축명령을 주었을 때 피치자세, 자세각속도, 속도, 위치가 유사한 형태로 시뮬레이션이 진행됨을 확인 할 수 있었다.

Table 3. Test Case for Simulation Similarity Verification

Command Type			Res. Type	Response Tracking
Axis	Type	Mag.		
Pitch	Step	-0.1	ACAH	Att. Hold
Roll	Step	0.1	ACAH	Att. Hold
Pitch	Step	-0.1	TRC	Vel. Hold
Roll	Step	0.1	TRC	Vel. Hold
Yaw	Step	0.1	RCDH	Att. Hold
Heave	Step	±0.2	RCDH	Att. Hold
Pitch	Doublet	±0.2	ACAH	Att. Hold
Roll	Doublet	±0.2	ACAH	Att. Hold
Yaw	Doublet	±0.2	RCDH	Att. Hold
Heave	Doublet	±0.2	RCHH	Att. Hold

ACAH: Attitude Command Attitude Hold
 TRC: Translational Rate Command
 RCDH: Rate Command Directional Hold
 RCHH: Rate Command Height Hold

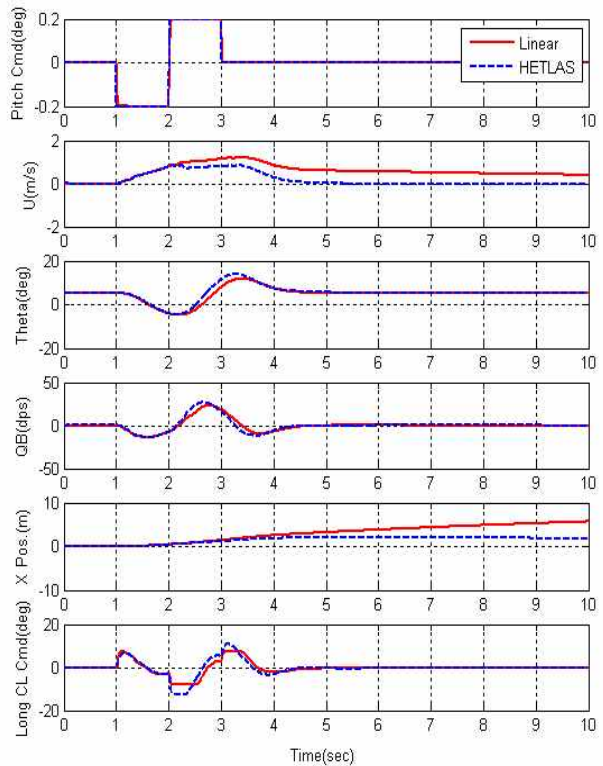


Fig. 7. Comparison of Model and HETALS Simulation Result (Longitudinal Doublet)

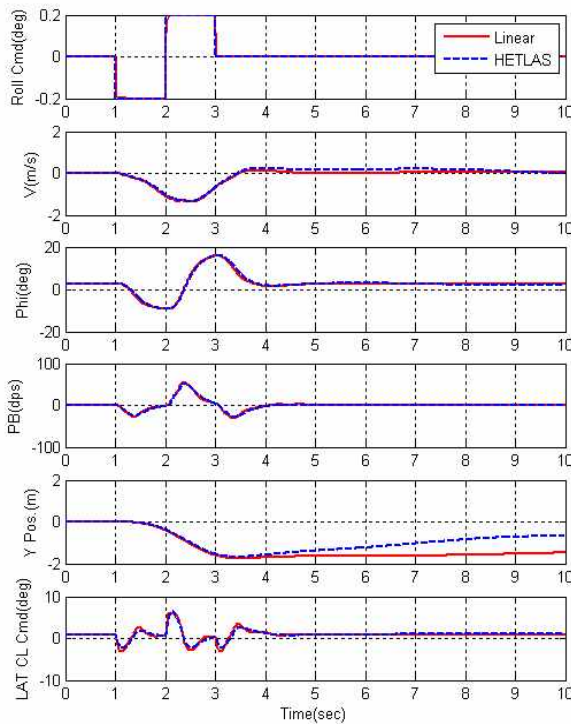


Fig. 8. Comparison of Model and HETALS Simulation Result (Lateral Doublet)

V. 결 론

모델기반설계 기법은 여러 가지 편리한 기능과 장점을 갖고 있는 설계 기법이며 그러한 장점으로 여러 분야에서 사용되고 있다. 그리고 그런 추세에 맞추어 DO-178C/DO-331에서 모델기반설계 기법에 대한 설계/검증절차에 대한 기준을 제시하고 있다. 그러나 모델과 소스코드의 일치성에 대해서는 모델기반설계 도구 개발업체에 의존하는 실정으로 개발자의 자동생성코드의 대한 추가적인 검증절차가 필요하게 되었다. 본 논문에서는 자동생성코드의 궁극적인 목표를 위하여 자동생성코드의 정형화된 개발 및 검증절차를 제시하였다. 그리고 개발된 검증절차를 이용하여 현재 개발중인 전자식 헬리콥터 제어법칙 개발에 적용하면서 단계별로 발생하는 모델과 소스코드의 결함을 감지하고 최종적으로 모델을 업데이트하고 있다. 그리고 소스코드의 일치성과 무결성을 검증함으로써 자동생성코드의 기능검증만으로도 배포 가능함을 목표로 하고 있다. 향후 개발완료 시까지 개발 및 검증절차의 성숙도를 높임으로써 다른 개발프로젝트에서도 중요한 개발요소로서 사용될 것으로 판단된다.

후 기

본 연구는 지식경제부가 주관하는 산업융합원천기술개발사업(신산업·주력산업)의 일환으로 수행 되었으며 연구를 지원해주신 관계자 여러분께 감사드립니다.

References

- 1) Tom Erkkinen, "Simulink Capabilities for Embedded Code Generation", The Mathworks, June 3, 2003
- 2) David W. Nixon, "Flight Control Law Development for the F-35 Joint Strike Flight", Lockheed-Martin Aeronautics, 5 Oct. 2004
- 3) Sang-Ook Cho, Kee-Young Choi, "A Study on Validation of OFP for UAV using Auto Code Generation", Journal of The KSASS Vol. 37 No.4, 2009
- 4) Marty Gasiorowski, LearnDO-178C, LDRA, Mar. 2012
- 5) Seong-Jun An, In-Je Cho, Chang-Duck Yang, Han-Ju Lee, "A Process and Tool Development of Auto Generated Code for Fly-By-Wire Helicopter Flight Control Law", Proceedings of the 2012 KSAS Spring Conference, 2012
- 6) Sang-eon Park, Seong-Jun An, "Autocode Generation for a Fly-by-Wire Air Vehicle Flight Control Operational Flight Program", ISET, 24-25 May, 2012
- 7) Chong-Sup Kim, In-Je Cho, Seung-Duck Lee, Han-Ju Lee, "A Study on Fly-By-Wire Flight Control Law Development Environment", Proceedings of the 2011 KSAS Spring Conference, 2011
- 8) Chong-Sup Kim, Jeong-Gi Lee, Se-Young Choi, Eun-Sik Jung, In-Je Cho, "A Study on Verification of Auto Code for UAV Flight Control Law Software", Proceedings of the 2012 KSAS Fall Conference, 2011
- 9) Chong-Sup Kim, In-Je Cho, Seung-Duck Lee, Han-Ju Lee, "Design and Validation of Model Inversion Flight Control Law for Fly By Wire Helicopter", Journal of The KSASS, Vol.40, No.8, 2012, pp 678-687