技術論文

DOI:10.5139/JKSAS.2010.38.12.1217

모델기반 개발기술을 적용한 무인항공기 비행제어 소프트웨어 개발

문정호*, 신성식*, 최승기*, 조신제*, 노은정*

Development of UAV Flight Control Software
using Model-Based Development(MBD) Technology

Jungho Moon*, Sungsik Shin*, Seungkie Choi*, Shinje Cho* and Eunjung Rho*

ABSTRACT

This paper describes the Model-Based Development(MBD) process behind the flight control software of a close-range unmanned aerial vehicle(KUS-9). An integrated development environment was created using a commercial tool(MATLAB Simulink®), which was utilized to design models for linear/nonlinear simulation, flight control law, operational logic and HILS(Hardware In the Loop Simulation) system. Software requirements were validated through flight simulations and peer reviews during the design process, whereas the models were verified through the application of a DO-178B verification tool. The integrity of automatically generated C code was verified by using a separate S/W testing tool. The finished software product was embedded on two different types of hardware and real-time operating system(uC/OS-II, VxWorks) to perform HILS and flight tests. The key findings of this study are that MBD Technology enables the development of a reusable and an extensible software product and auto-code generation technology allows the production of a highly reliable flight control software under a compressed time schedule.

초 록

본 논문은 근접감시 무인항공기(KUS-9)의 비행제어 소프트웨어(S/W) 개발 과정과 모델기반 개발 기술 적용 결과를 다룬다. 대표적인 상용 모델기반 설계도구인 MATLAB Simulink®를 활용하여 통합개발 환경을 구축하고 비행제어법칙, 운용로직, 비행 시뮬레이션 모델, HILS(Hardware-in-the-Loop Simulation) 시스템 모델을 설계하였다. 설계 과정에서 요구사항 충족을 위한 시뮬레이션 및 동료검토를 수행하고 DO-178B 검증 도구를 이용하여 모델을 검증한 후 S/W시험 도구를 통해 C코드의 무결성을 검증하였다. 탑재 소프트웨어는 두 기종의 하드웨어 및 실시간운영체제(μC/OS-II, VxWorks)에 탑재하여 HILS시험과 비행시험을 수행하였다. 모델기반 개발 기법을 적용함으로써 S/W 재사용성과 확장성을 확보하고 자동코드생성 기술을 이용하여 고신뢰 비행제어 S/W를 단기간에 성공적으로 개발하였다.

Key Words : Model-Based Development(모델기반개발), Flight Control Software(비행제어 소프트웨어), UAV(Unmanned Aerial Vehicle, 무인항공기), Auto Code Generation(자동코드생성), HILS(Hardware-in-the-Loop(HIL) Simulation), Flight Control System(비행제어시스템)

† 2010년 8월 25일 접수 ~ 2010년 11월 9일 심사완료

* 정회원, 대한항공 항공기술연구원

교신저자, E-mail : junghomoon@koreanair.com

대전시 유성구 전민동 461-1 대한항공 R&D센터

I. 서 론

걸프전 이후, 전 세계적으로 무인항공기(UAV: Unmanned Aerial Vehicle)에 대한 관심이 높아지고 있으며 최근에는 소형 무인항공기뿐 아니라 무인전투기와 같이 자율성(Autonomy)이 높은 UAV의 개발이 활발히 진행되고 있다. 특히 미공군 지원으로 보잉사가 개발한 X-45A UCAV(Unmanned Combat Aerial Vehicle)의 경우, 두대가 동시에 이륙하여 리소스를 스스로 관리, 협동하여 SEAD(Suppression of Enemy Air Defenses) 임무를 수행하고 또 다른 임의의 위협 상황에 대해 스스로 판단하고 목표물을 파괴하는 등 UCAV의 자율적 임무 수행 가능성을 비행시험을 통해 시험하였다. 또한 최근에는 Global Hawk의 KQ-X 프로그램 및 J-UCAS 프로그램에서 UAV 간 자동 공중급유 기술이 활발히 연구되고 있다. 이러한 UAV 기술이 발전할수록 비행제어 소프트웨어의 기능은 보다 다양해지고 복잡성(Complexity)은 기하급수적으로 증대된다. 반면 개발 기간 및 예산은 감소하는 추세로 이러한 개발 환경의 해결책으로 COTS(Commercial off-the-shelf) 기반의 하드웨어 적용과 재사용성 및 확장성이 보장되는 공통 표준 S/W를 적용하는 경향이 확대되고 있다.

기존의 개발 방법(개발 문서, 스크립트에 의한 수동 코딩)으로는 이러한 복잡한 S/W를 단기간에 개발하고 검증하는데 많은 어려움이 있다. 따라서 Global Hawk, Fire Scout, X-45, X-47, J-UCAS 등의 최신 UAV 비행제어 S/W 개발에 모델기반개발(MBD: Model-Based Development) 및 컴포넌트 기반 설계, 객체지향 개발 기법 등이 적용되고 있다.

또한 항공탑재 S/W 국제 인증 규정인 RTCA DO-178B "Software Considerations in Airborne Systems and Equipment Certification"[2]에서도 모델기반 개발의 중요성을 인지하고 규정을 개정하기 위해 모델기반 설계 및 검증, 객체지향 기술(Object-Oriented Technology), Formal Method, 도구 인증(Tool Qualification)에 대한 전문 검토 그룹을 구성하고 2010년말 발표예정인 DO-178C 규정에 반영될 계획이다[1].

본 연구에서는 MBD 적용 분야를 확대하여 비행제어법칙뿐만 아니라 UAV 운용로직, 시스템 모델, HILS 시뮬레이터 모델 등 비행제어 S/W 개발 전반에 MBD기술을 적용하였다. 본 논문에서는 MBD의 특징과 비행제어 시스템 개발단계별 적용 내용에 대해 다룬다.

II. 본 론

2.1 모델기반 개발 기법

항공 탑재 시스템의 기능이 다양화되고 하드웨어가 통합되면서 항공 내장형 S/W의 규모와 복잡성이 증가하고 있다. Fig. 1은 에어버스와 보잉사의 항공기에 탑재되는 내장형 S/W 코드 규모를 나타내는 것으로 S/W 규모가 4년간 2배씩 증가하고 있는 것을 알 수 있다[3].

이러한 내장형 S/W의 복잡성이 증가하면서 새로운 프로그램 언어가 지속적으로 개발/발전하고 있으며 C, C++, C#, Java에 이어 최근에는 모델을 기본으로 한 개발 언어가 활발히 적용 중이다. 내장형 S/W에 적용되는 대표적인 모델기반 개발 도구로는 MATLAB/Simulink®, SCADE®, MATRIXx®, TargetLink® 등이 있으며 본 연구에서는 전 세계적으로 가장 널리 사용되고 지속적인 성능 개선, 기술지원이 가능하며 사용자 정의가 용이한 MATLAB/Simulink®를 기본 설계 도구로 적용하였다.

모델기반 개발 기법은 기존의 C코드로 S/W를 설계, 구현, 통합, 시험하던 것을 모델 차원에서 수행함으로써 다음과 같은 효과가 있다.

- 복잡한 S/W를 구조화하여 정보 공유 용이
- 상위 레벨의 개념설계 및 성능 예측 가능
- 설계단계에서부터 요구사항 검증 가능
- 모델 및 탑재 S/W간 형상 일치
- 자동코드생성으로 휴먼 에러 최소화
- 요구사항 및 핵심 알고리즘 개발에 집중

Fig. 1. Onboard SLOC(Software Lines Of Code) growth

2.2 비행제어 시스템

본 연구에 적용된 KUS-9 UAV는 미국 국방부 분류 기준의 Level2에 해당하며 이러한 분류의 UAV는 일반적으로 모든 탑재장비가 비행조종컴퓨터(FCC : Flight Control Computer)를 중심으로 구성된다. Fig. 2는 KUS-9의 탑재시스템 구성을 나타낸 것으로, FCC는 GPS/INS 항법장비, 대기측정장비, 조종면 구동기 등의 비행 필수(Flight Critical) 장비를 비롯하여 탑재통신제어기, EO(Electric Optic)/IR(Infrared) 감지기, 전방영상 카메라, 영상처리장비, 엔진모니터링장비, 전원제어장비 등 모든 탑재장비를 통합 관리한다[5].

KUS-9 FCC는 고신뢰도를 확보하기 위해 프로세서보드, 입출력보드, 전원보드를 이중화하고 이중화 고장진단 S/W, 다수 조종면을 이용한 조종면 고장허용 S/W, 비행영역보호 S/W, 전방카메라와 영상처리장비를 이용한 영상기반 자동회수 S/W를 탑재하고 있다.

Fig. 2. Onboard system of KUS-9 UAV

2.3 비행제어 소프트웨어

비행제어 S/W는 실시간 운용체제와 이중화 관리, 운용 로직, 비행제어법칙 모듈로 구성된다. Fig. 3은 비행제어 S/W 전체 구성을 나타내는 것으로 본 연구에서는 응용 S/W에 해당하는 UAV 운용로직과 비행제어법칙 개발에 모델기반 개발 방법을 적용하였다.

Fig. 4는 모델기반 개발환경을 나타낸 것으로 FCC 모델은 C코드를 생성하여 실제 FCC에 탑재할 모델이며 통합 시뮬레이션 모델은 FCC 모델을 시뮬레이션하고 모델 레벨에서 검증하며 HILS 시험시 실시간 비행 시뮬레이션 장비에 탑

Fig. 3. Components of flight control computer software

Fig. 4. Integrated model-based development environment

재되는 모델이다. 두 모델은 FCC 입출력 버스를 이용하여 “Model Reference” 형태로 링크함으로써 FCC 모델 형상을 유지한 상태로 기능 및 성능 시험이 가능하도록 설계하였다. 통합 시뮬레이션 모델은 6자유도 비행운동모델, 구동기, EO/IR 감지기 운동 모델을 비롯하여 각 탑재장비를 모의하는 시스템 모델과 HILS 하드웨어 및 외부 인터페이스 모델로 구성된다. 또한 시뮬레이션을 위한 지상통제 UI(User Interface)를 설계하고 마이크로소프트사의 ESP®를 활용하여 비행 상태를 시험하도록 설계하였다. 통합 시뮬레이션 모델은 총 5,500개의 블록과 51,500개의 논리로 구현되었다.

기존의 개발 방법에서 Simulink®는 제어기 설계 및 시뮬레이션 용도로 활용되고 내장형 S/W는 별도의 C코드 개발 환경에서 개발된다. 하지만 모델기반 개발 방법에서는 자동코드생성 기술을 이용하므로 별도의 S/W 구현 절차 없이 Simulink 모델이 곧 내장형 S/W가 된다. 따라서 모델 설계시 보다 많은 사항들이 추가적으로 고려되어야 한다.

- 불연속 시간 기준 블록 적용
- 입출력 변수 구조, 크기, 형태 정의
- 서브시스템, 입출력, 신호의 명칭 부여
- 정보 도시 방법 정의
- m파일 모델 대신 S-Function 권장
- Stateflow 설계 지침 준수

위 사항들을 고려하기 위해서는 모델 설계 기준이 필요한데 본 연구에서는 MABB(Mathworks Automotive Advisory Board)에서 제시한 가이드 라인을 테일러링하여 적용하였다.

FCC 모델 중 운용로직은 시간에 의존, 순차적으로 작동하는 것이 특징이므로 다이어그램과 이벤트 주도 방식의 Stateflow[®]로 설계하였으며 Fig. 5는 운용로직의 일부인 영상기반 자동회수 모드 설계 결과이다. Stateflow[®]는 여러 State와 State간 흐름을 이어주는 Transition, State 및 Transition에서의 실행문, 단위 기능을 가진 함수로 구성된다. 이러한 다이어그램 방식의 설계는 복잡한 로직을 단순화하여 도식화하므로 개발자간의 의사소통 및 동료검토에 매우 효과적이다. Fig. 6은 비행제어법칙 모델로 기본제어기, 비행영역보호시스템, 영상기반자동회수를 위한 서브시스템 등으로 설계하였다.

Fig. 5. Automatic recovery mode design using state diagram

Fig. 6. Flight control law design using modeling tool

Fig. 7. Correction case using DO-178B Check of "model advisor"

모델 설계 지침은 개발 편의성, 효율성과 관련되기도 하지만 일부 지침은 코드생성에 직접적으로 영향을 미치므로 모델의 설계 지침 준수 여부와 모델링 결과를 검증할 필요가 있다. 최종 통합된 FCC 모델은 "Model Advisor"를 통해 DO-178B 지침 49개 항목, MAAB 지침 43개 항목의 준수 여부를 검사하여 모델을 개선하였다. Fig. 7은 DO-178B 검사를 통한 모델 개선 사례로, Floating point에서 '~=' 비교 연산자는 연산 오류 가능성이 있으므로 DO-178B Section 6.3의 알고리즘 정확도 요구조건을 만족시키기 위해 비교 연산자를 '>'로 변경하였다. 그 외에도 Detection Precision Loss, Underspecified Data Types, Vector/matrix block input conversion, Signal label mismatch, Absolute time 등을 개선하여 S/W 오류 가능성을 최소화하였다.

2.4 자동코드생성 및 통합

모델기반 개발 방법을 적용하기 위해서는 설계한 모델을 탑재 소스코드로 변환하는 코드 생성기가 필수적이다. Simulink[®]에 적용 가능한 코드 생성기로는 MATLAB의 RTW-EC, SCADE의 KCG, dSPACE의 TargetLink 등이 있다.

Table 1. Comparison of automatic code generator[4]

Properties	RTW-EC	Target Link	KCG
Simulink Support	High	Medium	Low-Medium
User-Friendliness	Medium	High	Medium
Customization Options	High	Medium-High	Low
Standard compliance	Low-Medium	Low-Medium	High

Table 1은 각각의 코드생성기 특성을 비교한 것으로 KCG는 에어버스사의 항공 S/W를 기반으로 개발된 만큼 개발 규정을 가장 잘 준수하는 반면, Simulink®의 다양한 블록 사용에 제약이 있고 대부분의 설정을 수동으로 함으로써 사용자 편의성이 떨어지는 단점이 있다. Target Link는 자동차의 엔진제어장치 S/W를 기반으로 개발되어 코드 생성을 편리하게 할 수 있으나 Simulink® 블록 및 사용자 옵션 지원이 제한되며 DO-178B 지원이 되지 않는 단점이 있다[4]. 본 연구에서는 다양한 블록, 하드웨어, 사용자 옵션을 지원하고 DO-178B 인증 키트를 제공하는 RTW-EC를 기본 코드생성 도구로 사용하였다.

Table 2는 2006년 볼보사에서 수행한 자동 생성 코드의 정적 해석 결과이다[4]. 코드 품질의 정량적 지표를 나타내는 Cyclomatic Complexity, Estimated static path count의 경우 RTW-EC는 권장하는 범위내로 들어오는 반면 다른 도구는 권장 범위를 벗어난 것을 알 수 있다. 또한 프로그램의 난이도를 나타내는 Halstead 수치는 1에 가까울수록 프로그래머가 이해하기 쉬운 코드로 RTW-EC가 가장 낮은 수치를 기록하였다. KCG는 대체적으로 범위를 벗어나는 경우가 많은데 그 이유는 DO-178B를 충족시키기 위해 Else문 또는 If-Else문을 사용하지 않고 코드 및 모델을 최적화하기보다는 신뢰성을 높이기 위해 요구사항에 입각하여 코드를 생성하기 때문이다.

반면 RTW-EC는 모델링 방법에 따라 코드의 안전성이 결정되는 단점이 있다. 따라서 본 연구에서는 자동생성 코드를 Formal 방식의 코드 검증 도구인 PolySpace®를 이용하여 검증하였다. 전체 20만 SLOC의 Run-time 에러에 대한 무결성을 약 92% 달성하였으며 8%는 하드웨어에 의존적인 메모리 관리 등의 특수한 코드들이다.

Table 2. Matric of generated code

Metric	Recomm ¹⁾	RTW-EC	Target Link	KCG
Cyclomatic Complexity	2-15	11	19	20
Maximum nested control structures	1-5	3	2	1
Estimated static path count	4-250	57	34992	1E+05
Halstead's program difficulty	-	11,12	18,55	19,99
Executable lines	1-70	45	77	144

1) Recommended metric value from "A comparison between handwritten and automatic generation of C code from SDL using static analysis"

2.5 HILS 및 비행시험

FCC S/W 개발이 완료되면 H/W와 S/W를 검증하기 위해 HILS시험을 수행한다. HILS시험을 위해서는 HILS 시뮬레이터, 영상 생성기, 비행 운동모사장비 등의 시험장비와 관련 S/W 개발이 선행되어야 한다. MBD 방법에서는 HILS 모델을 별도로 개발하지 않고 FCC S/W 개발시 사용했던 통합 시뮬레이션 모델을 100% 활용함으로써 개발 시험의 일관성을 유지하고 개발 기간을 단축할 수 있다. Fig. 8은 통합 시뮬레이션 모델 중 대기자료시스템 모델 개발 사례를 나타낸 것으로 본 연구에 적용된 탑재장비 모델들은 인터페이스 시험 및 고장 장입을 위한 Test 서브모델, 장비의 특성 및 로직을 구현한 Dynamics 서브모델, 결과값을 변환하는 데이터 패키징 서브모델의 3가지 형태로 구성하여 개발하였다.

최종 통합된 HILS 모델은 시뮬레이터의 H/W 관련 서브 모델을 포함하도록 구성한 후 Targetlink를 이용하여 전체 모델에 대한 C코드를 생성하고 Fig. 9와 같이 시뮬레이터에 탑재되도록 개발하였다. 또한 HILS 모델에는 실제 FCC에 탑재되는 모델을 동일하게 내장하여 FCC 하드웨어 없이도 실시간 시뮬레이션이 가능하도록 구현함으로써 개발단계에서 운용 로직 및 영상 알고리즘 검증과 조종사 훈련에 매우 유용하다.

Fig. 8. Architecture of Air Data System model

Fig. 9. Model auto-code generation and targeting on HILS simulator

Fig. 10. Configuration of HILS test

Fig. 11. Flight test of vision based control

Table 3. Application of MBD technology

UAV	KUS-9X	KUS-9
Configuration		
Processor	Intel	MPC
RTOS	uC/OS-II	VxWorks
Flight test	Basic controller Fault redundant controller	Vision-based auto recovery controller

Fig. 10은 본 연구에 적용된 KUS-9의 HILS 구성도와 시험 형상을 나타낸 것으로 시뮬레이터와 FCC외에도 제어 루프에 영상처리장비를 포함시킴으로써 영상기반 자동회수 제어 성능을 실시간으로 검증하여 HILS 충실도를 향상시켰다. 모델기반 개발의 또 다른 장점 중 하나는 재사용성과 이식성으로 H/W와 무관하게 모델을 개발하고 코드 생성시 사용자 정의에 따라 탑재 가능한 형태로 코드를 생성할 수 있다는 점이다.

Table 3은 본 연구 결과를 적용한 FCC의 특징을

나타낸 것으로 3개월 내에 다른 형태의 H/W 및 실시간 운용체제에 탑재하여 HILS시험과 비행시험을 성공적으로 완료하였다. Fig. 11은 KUS-9의 영상기반 자동회수 오프셋 통과 비행시험시 외부 영상 및 탑재 영상처리장비의 영상을 나타낸다.

III. 결 론

본 연구에서는 무인항공기의 비행제어 소프트웨어 개발에 MBD 기술 적용 사례를 기술하였다. 상용 모델기반 설계 도구를 활용하여 모델 개발 환경을 구축하고 비행제어 소프트웨어 및 HILS 모델 설계, 모델 검증, 자동코드생성, C코드 검증을 수행한 후 최종 개발된 소프트웨어를 2종의 비행제어컴퓨터에 탑재 후 HILS시험과 비행시험을 수행하여 MBD 기술의 효용성을 확인하였다.

후 기

본 연구는 지식경제부의 근접감시용 무인항공기 시스템 기술개발 사업의 일환으로 수행되었으며 지원에 감사드립니다.

참고문헌

- 1) Tom Erkkinen, Bill Potter, "Model-Based Design for DO-178B with Qualified Tools", AIAA Modeling and Simulation Technologies Conference and Exhibit, Aug 2009.
- 2) RTCA Inc., "Software Considerations Airborne Systems and Equipment Certification", Document RTCA/DO-178B, Dec 1992.
- 3) J.P.Potocoki De Montalk, "Computer Software in Civil Aircraft", Sixth Annual Conference on Software Assurance, 1991.
- 4) Robert Hammarstrom, Josef Nilsson, "A Comparison of three code generators for Models Created in Simulink", Master's Thesis, Chalmers University of Technology, 2006.
- 5) 신성식, 최승기, 문정호, 조신제, 노은정, "근접감시 무인항공기의 비행제어시스템 개발", 한국군사과학기술학회, 2010.