

技術論文

J. of The Korean Society for Aeronautical and Space Sciences 41(10), 805-812(2013)

DOI:<http://dx.doi.org/10.5139/JKSAS.2013.41.10.805>

스마트무인기 비행운용프로그램 개발

박범진*, 강영신, 유창선, 조암

Development of Operational Flight Program for Smart UAV

Bum-Jin Park*, Young-Shin Kang, Chang-Sun Yoo and Am Cho

Korea Aerospace Research Institute

ABSTRACT

The operational flight program(OFP) which has the functions of I/O processing with avionics , flight control logic calculation, fault diagnosis and redundancy mode is embedded in the flight control computer of Smart UAV. The OFP was developed in the environment of PowerPC 755 processor and VxWorks 5.5 real-time operating system. The OFP consists of memory access module, device I/O signal processing module and flight control logic module, and each module was designed to hierarchical structure. Memory access and signal processing modules were verified from bench test, and flight control logic module was verified from hardware-in-the-loop simulation(HILS) test, ground integration test, tethered test and flight test. This paper describes development environment, software structure, verification and management method of the OFP.

초 록

스마트무인기의 비행제어컴퓨터는 탑재장비와의 입/출력 신호처리, 비행제어법칙 연산 그리고 고장진단 및 이중화 기능 등이 구현된 비행운용프로그램이 탑재되어 있다. 비행운용프로그램은 PowerPC 755 프로세서와 VxWorks 5.5 실시간운영체제 환경에서 개발되었다. 비행운용프로그램은 메모리 참조 모듈, 탑재장비 입/출력 신호처리 모듈 그리고 비행제어법칙 모듈로 구성되었고 각각의 모듈은 계층 구조로 설계되었다. 메모리 참조 모듈과 신호처리 모듈은 벤치 테스트를 통해 검증되었고, 비행제어법칙 모듈은 시뮬레이션 시험, 지상통합시험, 안전줄 시험 그리고 비행시험을 통해 검증되었다. 본 논문에서는 비행운용프로그램의 개발환경, 소프트웨어 구조 그리고 검증 및 관리방법에 대해서 기술하였다.

Key Words : Flight Control Computer(비행제어컴퓨터), Operational Flight Program(비행운용프로그램), Smart UAV(스마트무인기), Signal Processing(신호처리), Hardware-in-the-Loop Simulation(시뮬레이션시험)

1. 서 론

한국항공우주연구원(KARI)에서 개발된 스마트 무인기는 550마력의 터보샤프트 엔진이 장착되어

최대속도 500km/h와 체공시간 5시간의 성능을 목표로 개발된 1톤급 틸트로터 비행체이다. 틸트로터 비행체는 회전익 모드뿐 아니라 고정익 모드에서도 비행이 가능하기 때문에, 수직이착륙

† Received: July 29, 2013 Accepted: September 23, 2013

* Corresponding author, E-mail : bjpark@kari.re.kr

<http://journal.ksas.or.kr/>

pISSN 1225-1348 / eISSN 2287-6871



Fig. 1. KARI Smart UAV flight configuration

기능과 고속비행 성능을 필요로 하는 육상 및 해상임무에 최적화된 비행체이다. 스마트무인기의 비행형상은 Fig. 1에 보였다. 틸트로터 비행체의 비행모드 전환 기능은 로터가 장착된 나셀의 장착 각도를 비행속도에 맞게 안정적으로 회전시키는 틸트로터 제어기술로 구현이 가능하다. 틸트로터 제어기술은 지상명령체계와 탑재센서정보를 기반으로 비행제어컴퓨터에서 계산되고, 계산결과는 작동기제어기로 전송되어 비행체의 조종면을 동작시킨다. 비행제어컴퓨터는 항법센서, 대기센서, 엔진제어기, 작동기제어기 그리고 통신장비와 같은 비행필수 탑재장비 뿐 아니라 비행기록장치와 임무장비와 같은 비행비필수 탑재장비와도 연결이 되어있다. 또한 시스템의 신뢰도를 높이기 위하여 비행제어컴퓨터, 항법센서, 작동기제어기 그리고 통신장비는 이중화 시스템으로 구성되었다. 비행제어컴퓨터에는 틸트로터 제어기술 뿐 아니라 각종 탑재장비와의 입/출력 신호처리 기능, 고장진단 기능 그리고 이중화 기능이 구현된 비행운용프로그램이 탑재되어 있고, 소프트웨어 구조는 40% 축소형 비행체 개발 경험[1-3]을 바탕으로 설계되었다. 본 논문에서 스마트무인기를 위해 개발된 비행운용프로그램의 개발환경, 소프트웨어 구조 그리고 검증 및 관리방법에 대해서 전반적으로 기술하였다.

II. 개발환경

2.1 하드웨어 구성

스마트무인기의 비행제어컴퓨터는 이중화 시스템으로 설계되었고 1개의 채널은 CPU보드, 3개의 I/O보드 그리고 전원공급보드가 장착된 형태로 구성되어 있다. CPU보드에는 개발 및 디버깅을 위한 이더넷 포트와 RS-232 포트가 장착되어 있다. 3개의 I/O보드는 ARINC-429, RS-422, RS-232, analog 입력 그리고 discrete 입/출력



Fig. 2. Flight control computer of Smart UAV

Table 1. Specifications of flight control computer

Items		Specifications
Redundancy		1-LRU, 2-Channel
Processor		PowerPC 755
I/O Controller(IOC)		TI-2407
Processor/IOC Interface		Dual-ported RAM
Internal Bus		PCI 33MHz
Memory		Flash(8MB), SDRAM(128MB)
I/O	CPU	RS-232 x 1, Ethernet x 1
	ADIO#1	ARINC-429 9Tx/18Rx RS-422 x 1(CCDL), RS-232 x 3
	ADIO#2	RS-422/485 9Tx/11Rx, RS-231 x 1 Analog input x 11/output x 2
	ADIO#3	Discrete input x 15/output x 23
Dimensions (H x W x D, mm)		155.2 x 361.4 x 306.0
Weight (kg)		10.6
Power dissipation (W)		Up to 100 max.

채널로 구성되어 있다. 이중화 시스템으로 구성된 각각의 비행제어컴퓨터는 이중화 기능에 필요한 2개의 직렬통신 채널이 연결되어 있고, 신호 분기 기능을 갖는 매트릭스 박스를 통해 항법센서, 대기센서, 작동기 제어기, 엔진 제어기 그리고 통신장비와 같은 다 수의 탑재장비와 연결되어 있다. 비행제어컴퓨터의 형상과 제원은 각각 Fig. 2와 Table 1에 보였다.

2.2 소프트웨어 개발환경

스마트무인기 비행운용프로그램의 개발을 위한 소프트웨어 플랫폼은 Wind River사의 VxWorks가 사용되었다. VxWorks는 안전-필수(safety-critical) 제어시스템 개발에 필요한 검증되고 신뢰성 있는 고성능을 제공하는 실시간운영체제로 알려져 있다. VxWorks는 버전 5.5가 사용되었고 통합개발환경인 토네이도는 버전 2.2가 사용되었다. CPU 보드의 RS-232 포트에 하이퍼터미널을 연결하여 디버깅 작업 시 이용되었고,

VxWorks API에서 제공하는 spyLib는 비행운용프로그램의 성능평가에 이용되었다. 비행운용프로그램의 개발 및 검증은 토네이도의 타켓 셀을 이용하여 RAM모드에서 수행되었고, 최종적으로 검증된 비행운용프로그램의 이미지는 플래시메모리에 올린 후 지상 및 비행시험에 이용되었다.

III. 비행운용프로그램 구조

비행운용프로그램의 구조는 신호처리 수준에 의한 계층 구조, 기능에 의한 파일 및 함수 구조, 실행우선순위에 의한 태스크 구조 그리고 신뢰성 있는 데이터 참조를 위한 공유데이터 구조로 구분되어 설명이 가능하다.

3.1 계층 구조

비행운용프로그램은 신호처리 수준에 따라 CPU보드 및 I/O보드 관련 메모리 참조 모듈, 비행제어컴퓨터와 연결된 탑재장비 입/출력 신호를 처리하기 위한 모듈 그리고 비행제어법칙 모듈로 구성되어 있다.

메모리 참조 모듈은 비행제어컴퓨터의 입/출력 신호처리를 위한 일종의 디바이스 드라이버 기능을 수행한다. 메모리 참조 모듈은 PCI 버스를 통하여 메모리를 직접 참조하는 기능과 I/O 보드에 탑재된 DPRAM(dual port RAM)을 참조하는 기능으로 구분된다. 직접 참조되는 메모리는 이산신호 입/출력, 워치독(watchdog) 타이머, 비행제어컴퓨터 출력제어 그리고 CCDL(cross channel data link) 관련 기능으로 구성되어 있다. DPRAM은 순환버퍼의 구조를 가지고 있고, DPRAM 데이터 읽기/쓰기를 통해 비행제어컴퓨터와 연결되어 있는 탑재장비의 아날로그 입력신호와 직렬통신 입/출력 신호를 제어할 수 있다. DPRAM의 순환버퍼의 구조는 Fig. 3에 보였다.

탑재장비 입/출력 신호처리 모듈은 메모리 참조 모듈을 기반으로 구현되었고, 탑재장비의 전



Fig. 3. DPRAM ring buffer structure

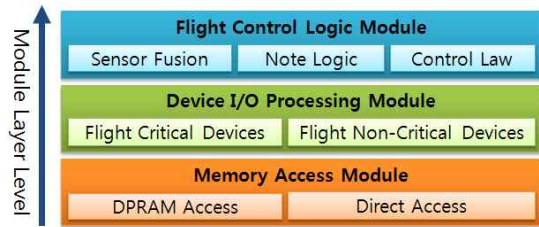


Fig. 4. OFP module structure

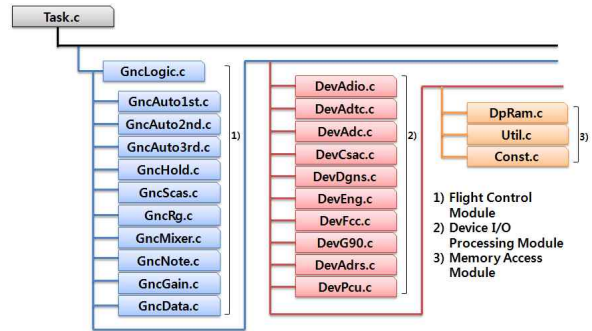


Fig. 5. OFP file structure

Table 2. File list of memory access module

파일명	주요기능
DpRam.c	DPRAM 순환버퍼 참조 아날로그신호 입력, 이산신호 입/출력 처리
Const.c	전역변수 및 전역상수 선언
Util.c	테이블 look-up, 공용 함수, 좌표계 변환, CRC계산, ARINC 레이블 관련 함수

기적 신호와 응용프로그램 수준에서 사용되는 물리량의 인터페이스 기능, 고장진단 그리고 이종화 관련 기능이 수행된다.

비행제어법칙 모듈은 지상명령과 각종 센서정보들을 기반으로 비행모드를 결정하는 운용 소프트웨어와 항공기의 자세, 속도 그리고 위치 등을 제어하는 임무 및 비행제어 소프트웨어로 구성되어 있다. 비행운용프로그램의 계층별 모듈 구조는 Fig. 4에 보였다.

3.2 파일 및 함수 구조

비행운용프로그램의 소스코드는 C언어로 개발되었고, 계층구조 및 주요기능에 따라 파일명과 함수명이 구분되어 있다. 전체적인 파일 구조는 Fig. 5에 보였다.

3.2.1 메모리 참조 모듈

메모리 참조 모듈은 하위수준의 함수로 구성되어 있고, DPRAM 순환버퍼 참조 기능, 아날로그 및 이산신호 입/출력 처리 기능, 공용 함수, 전역 변수 및 상수 관리 기능이 구현된 3개의 소스파일로 구성되어 있다. 파일항목과 주요기능은

Table 2에 보였다.

3.2.2 탑재장비 입/출력 신호처리 모듈

탑재장비 입/출력 신호처리 모듈과 제어법칙 모듈은 기능에 따라 구분된 다수의 파일로 구성되어 있다. 탑재장비 입/출력 신호처리 모듈은 10개의 소스파일로 구성되어 있고 각각의 주요기능은 Table 3에 보였다.

탑재장비 입/출력 신호처리를 구성하는 각각의 소스파일은 초기화 함수, 데이터 패킷 전송 함수, 수신 함수, 공유데이터 참조 함수, 메모리해제 함수와 같은 외부 함수와 송신 데이터 구성 함수, 수신 데이터 해석 함수, 고장진단 함수, 데이터 수신 감시 함수와 같은 정적 함수로 구성되어 있다. 공통 함수 구조는 Fig. 6에 보였다.

탑재통신장비, 항법센서 그리고 작동기제어기와 같은 이중화 시스템은 주/보조 장비에 대한 선택 함수가 추가적으로 구성되어 있다. 이 함수에는 각각의 장비에 대한 연결 상태와 고장진단 결과를 이용하여 주 장비에 이상이 있을 경우 보

Table 3. File list of I/O processing module

파일명	주요기능
DevAdc.c	ADC, RALT, ICDT 입/출력 처리 대기센서 보정, ADC 고장진단
DevAdio.c	아날로그 입력, 이산신호 입/출력 처리 착륙장치와 전원공급장치 고장진단
DevAdrs.c	비행기록장치 입/출력 처리
DevAdtc.c	탑재통신장비 입/출력 및 이중화 관리
DevCsac.c	작동기제어기 입/출력 및 이중화 관리 로터/플래퍼론/승강기 작동기 보정
DevDgns.c	항법센서 입/출력 처리 및 이중화 관리
DevEng.c	엔진관련(EEC, 연료장치, PLA, ADCV) 장비 입/출력 처리, PLA 보정
DevFcc.c	탑재장비 고장진단 비행제어컴퓨터 이중화 관리
DevG90.c	충돌회피장비 입/출력 처리
DevPcu.c	임무장비와 트랜스폰더 입/출력 처리

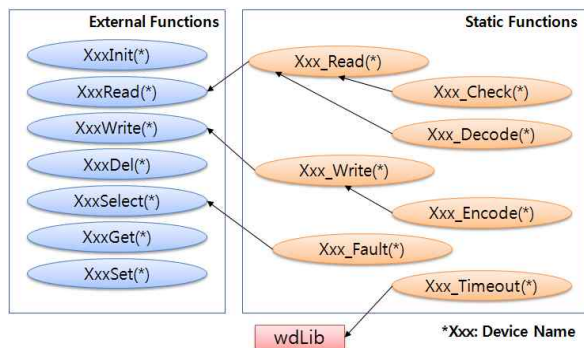


Fig. 6. Function structure of I/O module

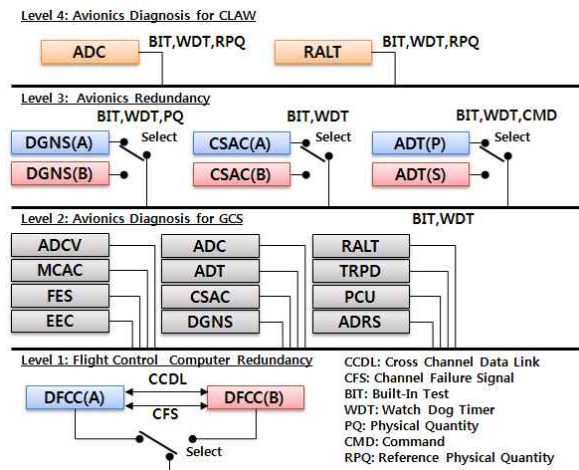


Fig. 7. Fault diagnosis and redundancy

조 장비로 전환하는 기능이 구현되어 있다.

비행제어컴퓨터 관련 입/출력 처리 모듈은 비행제어컴퓨터 이중화 기능을 구현하기 위한 별도의 함수가 추가적으로 구성되어 있다. 비행제어컴퓨터의 이중화 기능은 CCDL을 이용한 채널 전환, 위치독 타이머에 의한 채널 전환 그리고 재부팅 상황에 대한 대처 기능으로 구성되어 있고 각각의 기능은 독립적인 정적 함수로 구현되어 있다. 탑재장비 고장진단 및 이중화 기능에 대한 계층 구조는 Fig. 7에 보였다.

3.2.3 비행제어법칙 모듈

비행제어법칙 모듈은 11개의 소스파일로 구성되어 있고, 공유데이터 참조 기능을 제외하면 운영체제에 종속되지 않도록 개발되었다. 소스파일과 주요기능은 Table 4에 보였다.

제어법칙 관리 모듈은 센서융합, 노트로직 그리고 제어법칙 통합 기능으로 구성되어 있고 각각의 기능은 별도의 정적 함수로 구현되어 있다.

Table 4. File list of flight control logic module

파일명	주요기능
GncAuto1st.c	자동비행 상위 수준, auto recovery
GncAuto2nd.c	자동비행 중위 수준, return home
GncAuto3rd.c	자동비행 하위 수준, 점항법, 사전계획, 자동이착륙/호버, GPS스틱, 카메라유도
GncData.c	제어법칙 관련 공유데이터 참조
GncGain.c	제어법칙 이득 값 참조
GncHold.c	고도/속도/방위유지
GncLogic.c	센서융합, 노트로직 및 제어법칙 관리, 엔진 자동정지, 충돌회피
GncMixer.c	조종믹서
GncNote.c	제어법칙 운용 관련 노트로직
GncRg.c	로터 가버너
GncScas.c	Roll/Pitch/Yaw 축 SCAS

센서융합 함수에서는 항법센서, 대기센서 그리고 노즈봄에서 획득되는 각종 센서정보로부터 모든 비행조건에서 사용될 수 있는 융합된 고도, 속도, 방위 그리고 옆미끄러짐각이 계산된다. 노트로직 함수는 조종명령과 각종 센서정보를 바탕으로 현재의 비행모드와 제어법칙 선택에 관련된 다양한 조건을 결정하는 기능이 구현되어 있다. 제어법칙통합 함수는 비행제어법칙 모듈을 구성하는 각종 함수를 통합하여 운용하는 기능이 구현되어 있다. 각각의 함수들은 비행제어법칙 계층에 따라 자동비행, 유지모드, SCAS(Stability and Control Augmentation System), 로터가버너 그리고 조종믹서 순서로 순차적으로 호출된다. 제어법칙 관리 모듈에는 오버라이드가 가능한 엔진 자동정지 기능과 충돌회피 기능[4]이 별도의 정적 함수로 구성되어 있다. 오버라이드 기능을 위해 엔진자동정지 기능은 조종믹서모듈 다음 단계에 위치하고 충돌회피 기능은 자동비행모드 다음 단계에 위치한다. 비행제어법칙 모듈의 순차적 구조는 Fig. 8에 보였다.

비행제어법칙은 PID 제어기 구조로 설계되어 있고, 이득 값은 속도 및 틸트 각에 의해 스케줄링 되어 있다.

조종믹서 모듈에서는 조종기 스틱 혹은 SCAS 로직의 추력/롤/피치/요 명령, 각종 스위치 명령 그리고 현재 속도와 틸팅 각 정보를 입력 매개변수로 하여 좌/우측 로터 콜렉티브 피치, 종/횡 방향 사이클릭 피치, 좌/우 플랩, 승강기, 엔진 PLA(Power Lever Actuator) 그리고 틸팅 각도와 같은 명령 값이 계산되는 함수로 구성되어 있다.

로터 가버너 모듈은 로터의 회전수를 일정하게 유지시키는 제어기가 구현되어 있고 로터의 회전수는 운용모드에 따라 100%와 80% RPM 중에서 선택이 가능하다.

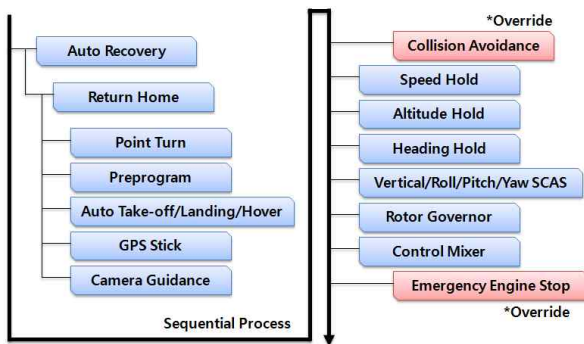


Fig. 8. Sequential structure of flight control logic module

비행제어법칙의 내부루프를 구성하는 SCAS 모듈은 수직/롤/피치/요 축에 대한 PID 제어기 구조로 구성되어 있다. 운용 모드에 따라 조종기 스틱 명령 혹은 고도/속도/방위 유지모드의 출력 값이 SCAS 모드의 명령 값으로 설정된다.

비행제어법칙의 외부루프를 구성하는 유지모드 모듈은 고도/속도/방위 축에 대한 PI 제어기 구조로 구성되어 있다. 운용 모드에 따라 관제장비의 노브 명령 값 혹은 자동비행모드의 출력 값이 유지모드의 명령 값으로 설정된다.

자동비행모드는 별도의 계층 구조로 설계되었다. 서로 독립된 하위 계층은 접항법, 사전계획모드, 자동이착륙 및 정지비행, GPS 스틱모드 그리고 카메라유도모드[5]로 구성되어 있다. 하위 계층은 관제장비의 운용모드 선택버튼에 의해 독립적으로 동작되거나 중위 계층으로 분류된 귀환모드에서 호출되어 사용된다. 상위 계층으로 분류된 자동회귀모드는 중/하위 계층의 자동비행모드 조합으로 구성되어 있다.

3.3 태스크 구조

비행운용프로그램은 50Hz 동작주기로 설계되어 있고 tRtc, tGnc, tNav 그리고 tPay로 구성된 4개의 태스크 구조를 갖는다. tRtc 태스크는 실시간 클럭을 생성하고 나머지 3개의 태스크 동작을 위한 각각의 세마포어 함수를 호출한다. tGnc 태스크에서는 비행제어법칙 모듈 관련 함수가 호출되고, tNav와 tPay 태스크에서는 비행 필수장비와 비행 비필수장비로 분류된 입/출력 신호처리 모듈 관련 함수가 각각 호출된다. tGnc 태스크에서는 태스크 동작 감시를 위하여 VxWorks API에서 제공하는 위치독 타이머가 연결되어 있고, tNav 태스크에서는 하드웨어 레벨에서 제공하는 위치독 타이머가 연결되어 있다. 두 개의 위치독 타이머는 비행제어컴퓨터 이중화 모듈에서 각각 사용된다. 각각의 태스크는 실시간 처리 중요도

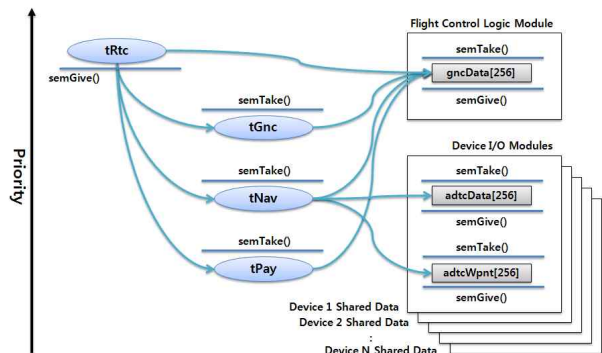


Fig. 9. Task and shared data structure

에 따라 실행우선 순위가 부여되었다. 비행운용 프로그램의 태스크 구조는 Fig. 9에 보였다.

3.4 공유데이터 구조

비행제어법칙 모듈과 입/출력 신호처리 모듈은 서로 다른 태스크에서 동작된다. 그리고 입/출력 신호처리 모듈은 비행필수 장비와 비행비필수 장비로 구분되어 실행우선순위에 의해 구분된 태스크에서 각각 동작된다. 한 개의 태스크 내에서 생성된 후 사용되는 데이터는 순차적으로 사용되기 때문에 데이터 참조 시 문제가 발생하지 않지만 비행운용프로그램은 다중 태스크로 구성되어 있기 때문에 태스크 간 공유데이터에 참조 시 발생할 수 있는 데이터 경합(contention) 문제를 고려하여야 한다. 비행운용프로그램에서는 파일 및 모듈 구조를 기반으로 세마포어를 사용한 데이터 참조 함수가 추가 되었다. 비행제어법칙 모듈은 한 개의 세마포어가 사용되었고 입/출력 신호처리 모듈은 파일 기반으로 총 10개의 세마포어가 사용되었다. 공유데이터 참조함수는 초기 값을 갖는 바이너리 세마포어가 이용되었다. 공유데이터 참조 시 세마포어를 얻은 후 반환 하는 구조로 설계되었고 읽기함수와 쓰기함수가 각각 구현되었다. 공유데이터는 256개의 크기를 갖는 double형 배열 변수가 각각 사용되었다. 이러한 구조는 각각의 공유데이터를 참조 하기위하여 함수를 여러 번 호출하여 프로세서의 부하가 발생할 수 있다. 그러나 다 수의 장비가 연결되어 있는 스마트무인기 비행제어컴퓨터의 입/출력 처리를 효율적으로 관리할 수 있는 장점이 있다. 비행운용프로그램에서 사용되는 공유데이터 참조 함수에 대한 호출 횟수를 분석한 결과는 Table 5에 보였다.

프로세서의 부하를 측정하기 위하여 VxWorks API에서 제공하는 spyLib를 이용하였다[6,7]. 그리고 별도의 시험 프로그램을 이용하여 공유데이터 참조를 위해 사용되는 함수호출 및 세마포어 사용에 대한 분석을 수행하였다. 세마포어를 2,000회 정도 사용하는 비행운용프로그램에서는 약 15%의 CPU 부하가 측정되었다. 함수호출 영

Table 5. Analysis of semaphore usage

태스크 생성 및 동작		4
제어법칙	비행모듈	265
	HILS모듈	288
탑재장비/F	비행필수	1,254
	비행비필수	168
총합		1,979

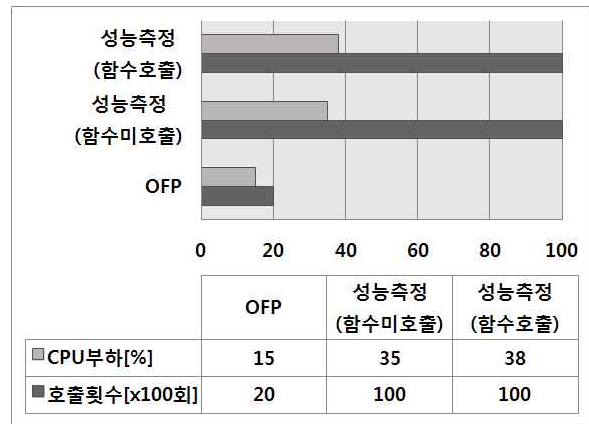


Fig. 10. Analysis of CPU usages

향성 분석을 위한 시험 프로그램에서 세마포어를 10,000회 사용 시 약 35%의 CPU 부하가 측정되었고, 세마포어를 래퍼(wrapper) 함수로 구성하여 10,000회 사용 시 약 38%의 CPU 부하가 측정되었다. 함수 호출에 의해 발생하는 추가적인 CPU 부하는 전체의 3% 미만이기 때문에 비행운용프로그램에서 사용하는 공유데이터 처리 방식에 큰 문제가 없음을 확인 하였다. 비행운용프로그램 및 세마포어 사용에 대한 CPU 부하 측정 결과는 Fig. 10에 보였다.

IV. 검증 및 관리

4.1 비행운용프로그램 검증

비행운용프로그램은 기능에 따라 계층 구조를 갖도록 모듈화 설계가 수행되었다. 각각의 모듈은 단위 시험을 통해 검증된 여러 개의 함수로 구성되어 있다.

디바이스 드라이버 역할을 수행하는 메모리 참조 모듈은 메모리 읽기/쓰기가 정상적으로 수행되어 비행제어컴퓨터의 모든 채널에 대한 전기적 신호가 정상 동작하는 것을 검증하여야 한다. 메모리 참조 모듈은 비행운용프로그램 동작주기에 따라 메모리 관련 읽기/쓰기 함수의 반복 호출 시험을 통해 채널 별 전기적 신호의 정상동작 여부를 확인하여 검증 되었다. 메모리 참조 모듈에 대한 검증시험 단계에서는 비행제어컴퓨터 외부 커넥터에서 전기적 루프백(loopback) 방식이 주로 이용되었다.

탑재장비 입/출력 신호처리 모듈은 비행제어컴퓨터와 연결된 모든 장비에 대해서 송/수신 데이터 패킷에 대한 검증이 필요하다. 이러한 검증 시험을 수행하기 위하여 비행제어컴퓨터와 전

기적으로 연결되는 탑재장비모사컴퓨터가 개발되었다. 탑재장비모사컴퓨터는 실제 비행체에 탑재되는 모든 장비의 입/출력 신호를 모사하는 기능과 송신 데이터를 구성하고 수신 데이터를 분석하는 기능이 구현되어 있다. 탑재장비모사컴퓨터를 이용하여 비행제어컴퓨터에 연결되는 모든 탑재장비와의 실시간 데이터 통신 시험이 가능하기 때문에 입/출력 신호처리 모듈에서 발생하는 최대 부하조건에 대한 시험도 수행되었다.

비행제어법칙 모듈은 스마트무인기 비선형운동모델 컴퓨터를 이용한 HILS 시험평가를 통해 검증되었다. HILS 시험환경을 구성하기 위하여 비행운용프로그램에는 별도의 HILS 모듈이 추가되었다. HILS 모듈은 탑재장비모사컴퓨터를 소프트웨어로 대체하기 위해 개발되었고 HILS 시험환경에서만 추가되는 별도의 모듈이다. HILS 모듈은 매크로에 의해 관리되며, 비행용 OFP에서는 빌드 단계에서 제거된다. HILS 모듈을 이용한 시험환경 구성은 Fig. 11에 보였다.

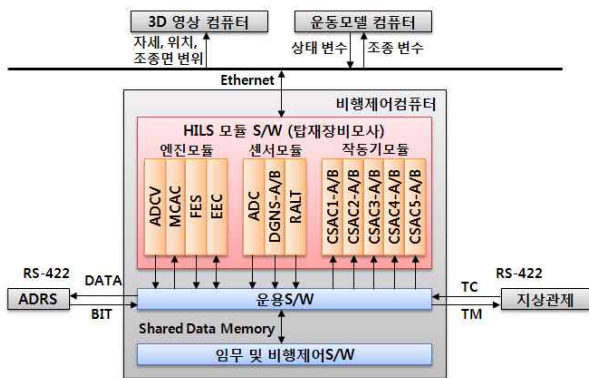


Fig. 11. HILS test configuration using HILS module S/W

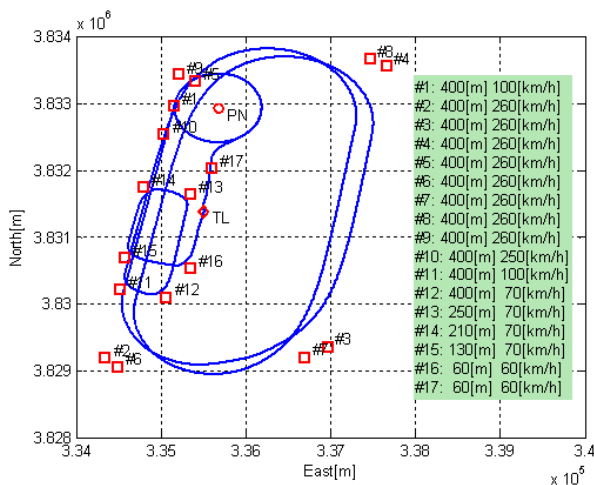


Fig. 12. Flight test plan for flight control logic module evaluation

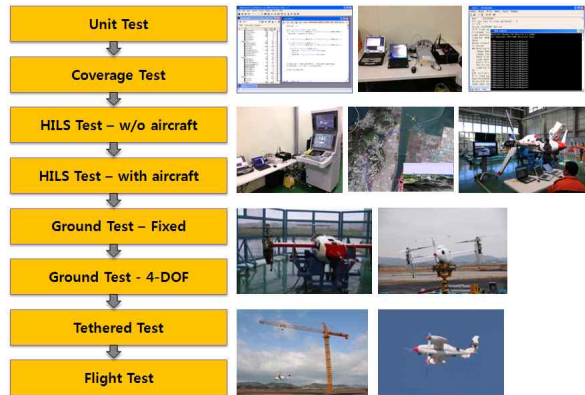


Fig. 13. OFP test and evaluation process

HILS 시험평가는 비행제어법칙 모듈의 기능 및 성능시험을 위해 수행된다. 비행제어법칙 모듈이 수정되면 수정 부분에 대한 기능과 성능시험이 제어법칙 및 비행운용프로그램 개발자에 의해 먼저 수행된다. 1차적으로 검증된 비행운용프로그램은 조종사가 참여하여 비행시험에 사용되는 비행계획에 따라 수행된다. 비행제어법칙 모듈에 대한 전반적인 기능점검에 사용되는 비행시험계획은 Fig. 12에 보였고, 비행운용프로그램의 전체적인 시험평가가 절차는 Fig. 13에 보였다.

4.2 비행운용프로그램 관리

비행운용프로그램에는 비행제어컴퓨터 채널 별 이중화 기능과 비행체 호기에 따라 별도로 관리되는 작동기 보정 식이 구현되어 있다. 비행운용프로그램의 소스코드와 생성된 이미지 파일을 통합 관리하기 위하여 비행체 식별 번호와 비행제어컴퓨터 채널 번호가 관리되는 두 개의 환경 변수가 비휘발성 메모리에 기록되었다. 환경 변수는 비행운용프로그램이 시작 될 때 비행제어컴퓨터의 비휘발성 메모리를 참조하여 설정된다. 비행체 식별 번호는 비행제어법칙 모듈의 이득 값 관리 모듈에서 사용된다. 비행체 식별 번호에 해당되는 이득 값 테이블은 이득 값 관리 모듈에서 참조되고 각각의 작동기 보정 식에 적용된다. 비행제어컴퓨터 채널 식별 번호는 입/출력 신호처리 모듈의 비행제어컴퓨터 관리 모듈에서 사용되고 비행제어컴퓨터의 채널 별 CCDL 통신 데이터 구성과 출력 전환기능에 참조된다.

V. 결 론

스마트무인기의 비행제어컴퓨터와 비행필수 탑재장비들은 이중화 시스템으로 구성되었으며, 이중화 시스템은 비행제어컴퓨터에 탑재된 비행

운용프로그램에 의해 관리된다. 비행운용프로그램은 계층 구조로 설계되었고 디바이스 드라이버 역할을 수행하는 메모리 참조 모듈, 탑재장비 입/출력 신호처리 모듈 그리고 비행제어법칙 모듈로 구성되어 있다.

비행운용프로그램의 전반적인 구조와 기능을 설명하기 위하여 모듈을 구성하고 있는 파일 구조, 함수 구성 및 기능, 태스크 구조 그리고 공유 데이터 참조 기능에 대해서 설명하였다. 비행운용프로그램 모듈은 다 수의 함수로 구성된다. 각각의 함수는 단위시험을 통해 먼저 검증되었고, 모듈 단위로 통합하여 검증시험이 수행되었다. 그리고 비행제어컴퓨터의 채널과 비행체 호기에 따라 다르게 운용되는 비행운용프로그램의 형상 관리 방법에 대해서 설명하였다.

스마트무인기의 비행운용프로그램은 실시간운영체제 환경에서 신호처리 수준, 탑재장비 기능, 임무 및 제어법칙 수준, 태스크 실행우선순위, 공유 데이터 참조 등을 고려하여 계층 및 모듈화 구조로 설계 및 구현되었고, 150회 이상의 비행과 누적 40시간 이상의 비행시간을 통해 기능, 성능 그리고 신뢰성이 검증되었다. 검증된 소프트웨어 구조 및 모듈이 향후 KARI의 차세대 무인기 프로젝트에 적용된다면 비행운용프로그램 개발 및 검증을 좀 더 효율적으로 수행할 수 있을 것으로 기대한다.

References

- 1) Park, B. J., Kim, S. P., Kang, Y. S. and Yoo, C. S., 2007.4, "Development of Operational Flight Program for Small-Scaled Smart UAV", The Korean Society for Aeronautical and Space Sciences, pp. 484~487.
- 2) Park, B. J., Kim, S. P., Kang, Y. S. and Yoo, C. S., 2007.4, "Operational Flight Program Optimization for Small-Scaled Tiltrotor Aircraft", The Korean Society for Aeronautical and Space Sciences, pp. 1268~1271.
- 3) Yoo, C. S., Park, B. J., and Kang, Y. S., 2008.8, "Evaluation of Operational Flight Program in Tilt Rotor UAV", AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii.
- 4) Park, B. J., Yoo, C. S., Cho, A., Kang, Y. S. and Koo, S. O., 2012.11, "Development of HILS System for Collision Avoidance Test of Smart UAV", The Korean Society for Aeronautical and Space Sciences, pp. 715~720.
- 5) Park, B. J., Kang, Y. S., Cho, A., Chang, S. H. and Yoo, C. S., 2012.4, "Development of HILS System for Camera Guidance Mode Test of Smart UAV", The Korean Society for Aeronautical and Space Sciences, pp. 500~505.
- 6) "VxWorks Programmer's Guide, 5.5," Wind River Systems Inc., 2002.
- 7) "Tornado User's Guide, 2.2," Wind River Systems Inc., 2002.