

무인항공기 비행제어 시스템 검증을 위한 PILS 시스템

유인항공기와 달리 무인항공기는 전기신호 제어장치(FBW, Fly-by-Wire)로 구성된 비행제어 시스템에 의해 항공기의 조종성능과 신뢰성이 결정된다. 이러한 중요성 때문에 무인항공기의 비행제어 시스템도 여타의 유인항공기 비행제어 시스템에 뒤지지 않는 검증 절차를 거쳐야 한다.

■ 김성수, 유창경, 최기영, 박춘배
(인하대학교 항공우주공학과)

1. 서론

현대 무인항공기 시스템은 디지털 비행제어 시스템(DFCS, Digital Flight Control System)의 수준에 따라 항공기의 전체적인 성능이 결정된다. 유인항공기와 달리 무인항공기는 전기신호 제어장치(FBW, Fly-by-Wire)로 구성된 비행제어 시스템에 의해 항공기의 조종성능과 신뢰성이 결정되기 때문이다. 이러한 중요성 때문에 무인항공기의 비행제어 시스템도 여타의 유인항공기 비행제어 시스템에 뒤지지 않는 검증 절차를 거쳐야 한다.

비행제어 시스템의 개발 과정에 따른 검증 절차는 세 단계로 나누어 진행된다. 우선적으로 엔지니어링 시뮬레이션을 통하여 비실시간 환경에서 성능 시험을 수행하고, PILS(Processor-In-the-Loop Simulation) 또는 HILS(Hardware-In-the-Loop Simulation)와 같은 가상의 폐루프 비행환경 시설을 구축하여 실시간 성능 시험을 수행하고, 마지막으로 비행시험을 통해 항공기의 성능을 최종 검증하게 된다. 비행제어 시스템의 주된 검증 내용은 항공기의 제어를 담당하는 비행제어법칙, 항공기 전체를 운용하는 OFP(Operational Flight Program), 시스템 간의 유기적인 통신 체계 및 프로세서 성능 등이 있고, 이러한 부분은 PILS 시스템

을 이용하여 검증하게 된다. 더 나아가 조종면 구동용 서보는 물론 INS(Inertial Navigation System)까지 포함한 HILS를 수행하기도 하나, 이를 위해서는 무인기의 회전운동을 실제로 발생시키기 위한 고가의 운동테이블을 요구한다. 따라서 무인기의 경우 HILS를 생략하거나 차량시험 등으로 대체하기도 한다.

최종 검증단계인 비행시험 방법은 비용/시간/장비/인원 등의 추가 요건이 만만치 않고, 특히 사고 발생에 대한 위험 부담이 항상 존재하고 있다. 무인항공기는 유인항공기에 비해 상대적으로 운용이 편하기 때문에 비행시험에 의한 검증 절차를 자주 수행하게 되는데 이때 가장 간과해서는 안될 사항으로 지상에서의 성능 검증 미비로 인한 사고가 빈번하게 발생한다는 것이다. 즉, 무인항공기의 비행제어 시스템일 수록 PILS(HILS) 시스템과 같은 지상 검증 장비를 이용하여 철저한 성능 검증 후에 비행시험을 수행해야 한다. 이러한 이유로 인해 PILS(HILS) 검증 시스템은 우주 발사체[1], 유도탄[2-3], 항공기[4-6] 등 다양한 분야에서 비행제어 시스템 검증용 도구로 적극 활용되고 있다.

PILS 시스템 구축은 가상의 시뮬레이션 환경을 구축하는 것으로서 근래의 PC가 고성능화 됨에 따라 소프트웨어 개발 및 시스템 구축이 좀더 용이해 졌다. 또한 윈도우, 리눅스 플랫폼

에서 운용 가능한 모델 기반의 시뮬레이션 구성 도구와 실시간 환경을 제공하는 Matlab, dSpace, RTLinux, RT-LAB, LabVIEW 등의 강력한 개발 도구를 사용할 수 있다.

이 글에서는 무인항공기 비행제어 시스템의 지상 검증용 도구로 사용되는 PILS 시스템 구성과 비실시간 시뮬레이션, PILS, HILS 시스템에 대한 단계별 시뮬레이션 검증의 차이점에 대해 서술하였다. 그리고 시뮬레이션 환경 구축을 위한 시스템 모듈의 특징을 열거하고, PILS 시스템 구성을 위해 필요한 시스템 간의 동기화, 실시간성, 동작 주기에 대한 설계 요구 조건에 대해 정리하였다. 마지막으로 무인항공기 비행제어 시스템의 발전방향에 맞춰 다중화 구조, 편대 비행 시스템 검증에 필요한 PILS의 구축 사례를 들어 검증 시스템의 발전 방향을 제시한다.

2. 지상 검증 시스템의 구분

실험실 단위에서 지상 검증 시스템을 사용하는 가장 큰 장점은 예측 가능한 비행 상황을 구성하고 그에 따른 반응을 살펴봄으로써 비행제어 시스템의 전반적인 성능을 사전에 검증할 수 있고, 극단적인 비행 환경에서의 성능을 확인할 수 있다. 특히 무인 항공기 비행제어 시스템 검증은 PC 기반의 상용 기성품을 사용할 수 있어 경제적으로 구성할 수 있다. 지상 검증 시스템은 실제 비행 상황을 가상의 페루프로 구성하기 때문에, 여기에 사용되는 모델이 현실세계와 얼마나 일치하는가에 따라 위에 열거된 장점이 극대화 된다.

이러한 지상 검증 시스템은 모델 기반의 비실시간 시뮬레이션(Model Based off-line Simulation), PILS, HILS로 구분된다.

2.1 모델 기반의 시뮬레이션

비행제어 시스템의 OFP 제어법칙 검증을 위한 기본 단계에서 사용된다. 그림 1과 같이 주로 Simulink 등과 같은 모델 기반으로 구성하고, PC를 이용하여 비실시간 시뮬레이션을 통해 제어기의 성능 검증을 주목적으로 한다.

장점으로는 최소의 시간과 비용으로 비행제어 시스템의 전반적인 결과를 예측해 볼 수 있다. 단점으로는 항공기 6-DOF, 센서, 작동기와 같은 탑재 장비를 해석적 모델로 구현하였기 때문에 실제 시스템의 적용 결과와 일정부분 차이가 발생한다. 즉, 각각의 구성요소를 얼마나 현실성 있게 모델링 하는가에 따라 사실성이 달라질 수 있다. 시뮬레이션 결과의 현실성이 떨어지는 원인은 주로 항공기 모델, 대기 모델, 센서 모델의 불확실성

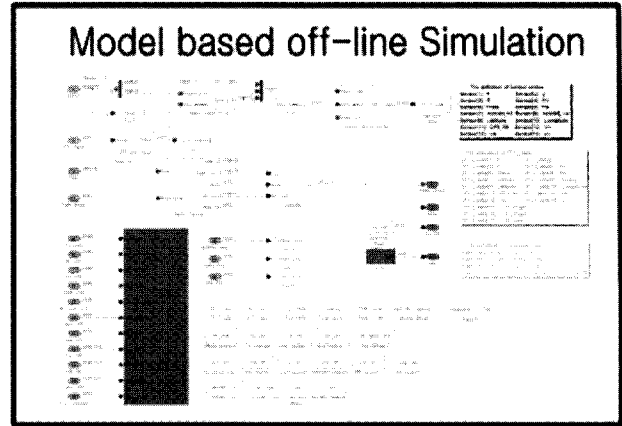


그림 1. 모델 기반의 시뮬레이션

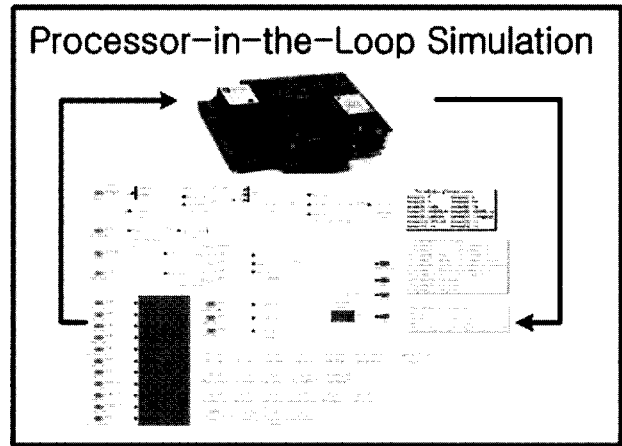


그림 2. PILS 시스템

과 조종면 작동기 모델의 시간 지연 등이 존재하기 때문이다.

2.2 PILS 시스템

무인항공기에 적용될 비행제어 컴퓨터가 갖추어져 있을 경우 PILS를 수행한다. 무인항공기의 비행제어 컴퓨터는 PC-104 형태의 인텔, ARM 프로세서를 사용하거나 PowerPC, 모토로라, Texas Instruments 사의 DSP 계열 프로세서를 적용하여 개발된 임베디드 보드를 사용한다. 그림 2에서와 같이 비행제어 컴퓨터가 추가되고, 모델 기반의 시뮬레이션에서 검증된 OFP를 비행제어 컴퓨터에 탑재한다. 이때 센서, 작동기 등은 모델 기반의 시뮬레이션에 사용하였던 해석적 모델을 그대로 사용한다. 즉, PILS에서는 실제 탑재되는 비행제어 컴퓨터(프로세서)에 의해 제어 법칙을 수행하게 된다.

PILS는 비행제어 컴퓨터의 성능을 확인하는데 유용하다. 즉,

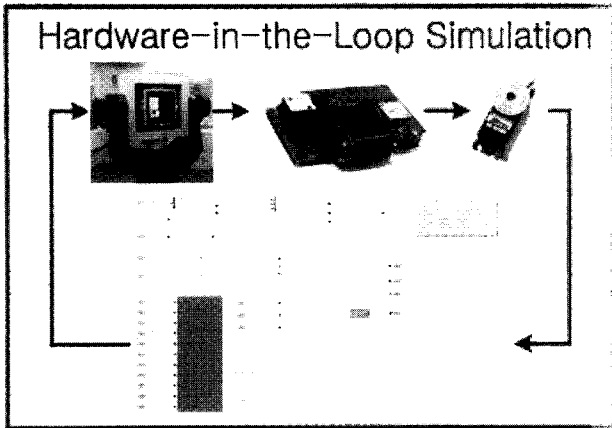


그림 3. HILS 시스템

비행제어 법칙을 수행함에 있어 컴퓨터가 실시간성을 만족하는지, 부동소수점을 필요로 하는 실수 연산은 원활한지, 메모리의 Overflow/Underflow는 발생하지 않는지, 프로세서 특징에 따른 특이점(singular point)은 발생하지 않는지 검증 할 수 있다. 특히, 비행제어 컴퓨터의 연산 능력은 표 1과 비슷한 범위의 계산 성능과 동작 여유가 있음을 확인해야 한다.

2.3 HILS 시스템

HILS 시스템은 항공기 6-DOF, 대기 모델 등을 제외한 모든 탑재 시스템을 실제 시스템으로 구성하게 된다. 그림 3과 같이 비행제어 컴퓨터가 포함되어 있고, 자세를 표현하기 위한 3축 운동판과 조종면 작동기를 구성하여 실제 데이터로 사용한다. GPS 시뮬레이터는 RS-232 통신을 이용한 데이터 출력 방식과 반송파를 공중으로 출력하는 방식이 있다. 반송파 출력 방식의 시뮬레이터를 사용하면 비행제어 시스템의 GPS 수신기도 검증할 수 있지만, 수천만원대의 장비이므로 실험실 단위에서는 데이터 출력 방식을 주로 사용한다. 진대기속도(true air speed), 압력고도(pressure altitude)를 측정하는 공력자료장치(air data system)는 동압과 정압을 생성해 주는 압력 시뮬레이터를 사용한다.

HILS는 가장 현실성 있는 비행제어 시스템 검증 도구로 활용된다. 하지만 가장 큰 단점은 모든 하드웨어가 전부 갖추어져 있어야 하고 이를 구축하기 위한 비용이 수억원 이상 소요된다.

위에서 비행제어 시스템 검증용 시뮬레이션 방법에 대해 살펴보았다. 지상 검증 시스템은 무인항공기 개발 과정에서 필수적이지만 연구 주체에 따라 보유기술/시간/비용 등 다양한 변수에 의해 적절한 시스템을 구성해야 한다. 대학 기반의 실험실일

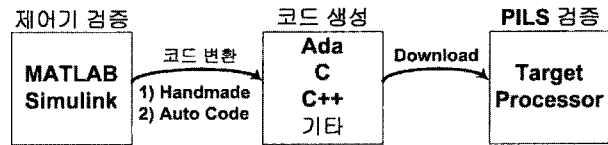


그림 4. 코드 변환 과정

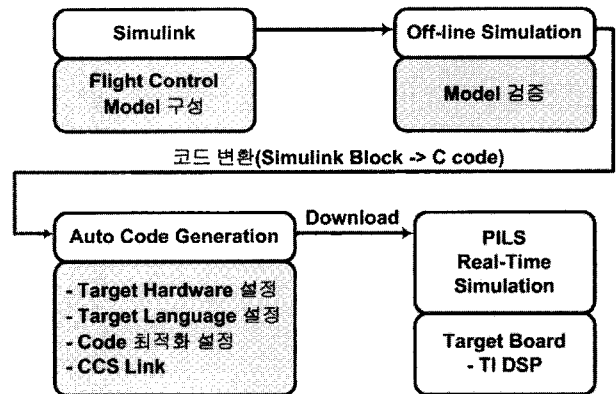


그림 5. 자동 코드 변환 과정 사례

경우 일반적으로 PILS 시스템을 구축하고 연구소의 경우 HILS 시스템을 구축하여 사용하고 있다. 지상 검증 시스템 구축의 추세는 PILS 시스템을 기반으로 구성 모듈의 불확실성을 제거하여 현실성을 극대화 시킴으로써, 저비용 고효율 검증 시스템을 구축하는데 노력하고 있다. 또한 국내 개발환경을 살펴보면 때 3축 운동판과 같이 고가의 시뮬레이션 시설은 연구소를 중심으로 운영되고 있다. 이러한 시설을 연구소 자체에서 운영하는 비율이 그리 높지 않다면 대학 실험실과 교류하여 사용하는 것도 한가지 대안이라 할 수 있다.

2.4 코드 변환의 문제점

앞에서 언급한 3가지 검증 시스템을 구축하여 사용함에 있어 가장 큰 문제점은 코드 변환과정이다. 코드 변환은 모델 기반의 시뮬레이션에서 PILS 또는 HILS 시스템으로 확장할 때 발생한다. 그림 4는 코드 변환 과정을 보여준다. 일반적으로 비행제어 법칙은 Simulink를 이용하여 설계하고 off-line 시뮬레이션을 수행하여 성능을 검증한다. 이렇게 검증된 제어법칙은 비행제어 컴퓨터에 이식할 수 있도록 Ada, C, C++ 등의 언어로 재생성되어야 한다. 코드 변환 과정은 검증된 Simulink 블록을 살펴보면 사람이 직접(handmade) 변환코드를 생성하게 된다. 이러한 과정 중에 제어법칙 오류가 발생할 수 있어 생성된 코드는 재검증 과정을 수행해야 한다. 또한 코드 변환 경험이 적은 엔지니어

어일 경우 오류 발생 확률은 더 높다.

위에 언급된 문제점을 해결하기 위해 모델 기반의 자동코드 생성(auto code generation) 도구들이 등장하였다. 대표적인 코드 생성 도구로는 SCADE, MATRIXx, Simulink 등이 있다. 실험실 단위의 PILS 시스템을 구성할 경우 이 중에서 비교적 쉽게 접근할 수 있는 Simulink 자동 코드 생성 기법을 적용할 수 있다. Simulink에서는 TI사의 C2000, C6000 계열 프로세서에 대한 자동 코드 생성 기능을 제공하고 이외의 프로세서에 적용할 경우 Real-Time Workshop Embedded Coder 기능을 선택하면 원하는 프로세서에 맞는 코드를 생성할 수 있다.

그림 5는 Simulink 모델로 구성된 비행제어 법칙을 자동 코드 생성에 의해 변환하고 타겟(C2000, C6000)이 되는 비행제어 컴퓨터에 다운로드 하는 일련의 과정을 보여준다. 그림에서 코드 변환은 Simulink 모델을 C언어로 자동 생성해 주고, Simulink와 연결되어 있는 Code Composer Studio에 의해 컴파일을 수행한 후 타겟 프로세서에 다운로드 할 때까지의 과정을 Simulink 구성 창에서 일괄 처리할 수 있도록 검증 환경을 구성할 수 있다.

3. PILS 시스템 구성 모델

PILS 시스템은 비행제어 시스템의 지상 검증용 도구이기 때문에 실제 비행 환경을 표현해주기 위한 해석적 모델링[7]이 필수적이다. PILS 시스템은 항공기 6자유도 운동 모델, 항공기 공력 데이터, 엔진 모델, 대기 모델과 자세를 표현하는 INS, 위치를 표현하는 GPS, 속도, 고도를 출력하는 센서 모델 그리고 조종면을 제어하는 작동기 모델로 구성된다. 그림 6은 지상 검증 시스템의 구성품 모델을 나타낸다. PILS 시스템을 구성할 경우 제어 법칙 모듈만 실제 구성품이고, HILS 시스템을 구성할 경우 격자무늬 안쪽에 있는 모듈이 실제 구성품에 해당한다.

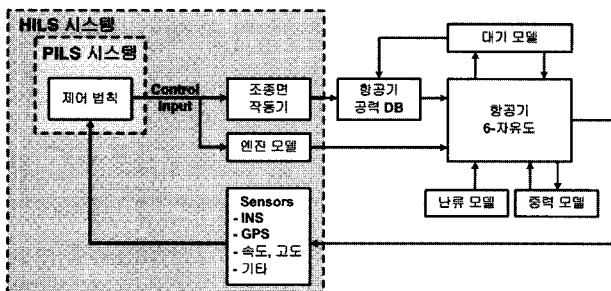


그림 6. 지상 검증 시스템 구성 모델

시뮬레이션에 필요한 항공기 모델은 DATCOM, 풍동실험, 비행시험에 의한 시스템 ID(Identification)[8] 방법 등을 이용하여 구성하게 된다. 엔진 모델은 사용하는 엔진의 데이터베이스가 제공될 경우 고도, 비행속도, 받음각 등에 따라 변화하는 엔진 모델을 구성할 수 있다.[9] 하지만 무인항공기의 경우 엔진에 대한 충분한 자료가 제공되지 않는 것이 일반적이다. 따라서 최대 추력에 대한 비율로서 간략화된 모델을 구성할 수 있다.

지구 대기 모델은 국제 표준대기 모델(International Standard Atmosphere Model)을 사용한다. 표준 대기 모델은 대기 성질의 수직 분포를 중위도에서 일년 동안의 개략적인 대표치를 가정한 것으로, 일반적으로 고도계 보정, 항공기 설계, 성능 해석 등에 사용되며 실제 조건과 매우 유사하다. 가장 최근 모델은 “US Standard Atmosphere, 1976” [10]으로 지구 대기의 정상상태 대표치를 지구 표면에서부터 1,000km까지 정형화한 모델을 사용할 수 있다. 고도에 대한 대기 모델과 더불어 바람에 의한 난류 모델은 Von Karman 또는 Dryden 모델을 적용한다. PILS 시스템에 의한 검증 절차를 수행할 경우 실제 비행환경의 대기와 유사성을 높이기 위해 대기 모델에 난류 모델을 필수적으로 적용해야 한다.

센서 모델[11]은 가속도/각속도를 출력하는 IMU, 자세와 방위를 출력하는 AHRS(Attitude Heading Reference System), 위치 정보를 출력하는 GPS, 이를 결합한 INS 그리고 고도계와 속도계 등을 구성한다. 센서 모델은 비행제어 시스템에 사용하는 센서의 특성을 정확히 반영할 수 있어야 하고, 받음각 센서, 옆미끄럼 각 센서, 전파 고도계 등 필요에 따라 추가 구성할 수 있다. 센서 모델 구성시 중요사항으로는 편향오차, 신호의 잡음, 시간 지연과 같은 오차 모델의 구성에 있다. 특히, 무인항공기의 경우 다양한 MEMS 센서를 적용하고 있기 때문에 이러한 센서의 특성을 모델링에 반영한다.

조종면 작동기는 제어 입력에 대한 반응이 상대적으로 늦게 발생한다. 이러한 시간지연 특성을 반영하기 위해 1차 또는 2차 모델로 표현되는 작동기 모델을 구성하여 시뮬레이션에 반영해야 한다.

이와 같은 시뮬레이션 모델은 MATLAB에서 제공하는 Aerospace Blockset이 대표적으로 사용되고 있으며, 그 이외에도 다양한 형태의 모델 기반 시뮬레이션 도구들이 활용되고 있다.[12-13] 그림 7은 Unmanned Dynamics, LLC에서 무료로 제공하는 AeroSim Blockset[14]의 구성 모델이다. 라이선스를 받아야 하는 MATLAB Aerospace Block을 대신하여 Simulink에서 쉽게 사용할 수 있는 무료 시뮬레이션 모델이므로 교육용으로 많이

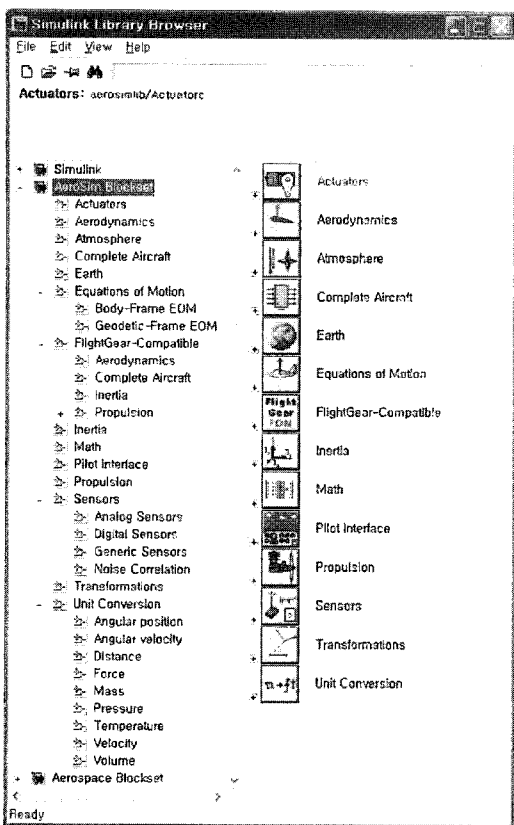


그림 7. AeroSim Blockset

사용되고 있다.

4. PILS 시스템 필수 고려사항

PILS 시스템 구축시 요구사항으로는 비행제어 법칙을 수행하는 비행제어 시스템의 동작 주기, 비행제어 시스템과 모델링 컴퓨터와의 데이터 동기화 및 실시간성이 중요하게 고려되어야 한다.

4.1 시스템 동작 주기

PILS 시스템은 비행제어 시스템의 동작을 모사하기 위한 것으로 비행제어 시스템과 동일한 주기의 실시간성이 보장되어야 한다. 이러한 이유로 인해 폐루프를 구성하고 있는 PILS 시스템의 데이터 갱신주기는 비행제어 시스템의 동작 주기에 비해 충분히 짧아야 한다. 즉, Nyquist 주파수 요구조건을 고려하여 비행제어 시스템 동작 주기의 1/2배 이상으로 데이터를 갱신할 수 있어야 한다. 표 1은 군용항공기의 비행제어 시스템 동작 주기와 프로세서의 동작 여유를 나타낸다. 예를 들어, F-16의 경

표 1. 군용항공기의 비행제어 시스템 계산 성능

%@Hz	F-16D	LAVI	JAS	B-2	F-15E	P-7A	T-50
DUTY	62	85	74	66	-	55	-
CYCLE	@80	@66	@60	@50	@80	@80	@64

우 동작 주기는 80Hz(12.5ms)이고, 프로세서는 약 62%(7.75ms)의 동작 시간을 할애하게 된다. 일반적으로 유인항공기 비행제어 시스템에 사용되는 프로세서의 최대 점유율을 약 60~80% 정도로 규제하고 있다. 이는 미처 예상치 못한 연산과정에 의해 프로세서 다운 현상을 방지 하기 위한 설계 제약조건에 해당한다. 따라서 무인항공기 비행제어 시스템의 프로세서 점유율도 이에 준하는 기준을 정하고 OFP를 작성해야 한다.

표 1에서 처럼 일반적인 고정익 항공기의 FBW(Fly-by-Wire) 비행제어 시스템은 50~100Hz의 주기로 비행제어 법칙을 수행하게 된다. 참고로 유도탄은 약 1000Hz의 주기로 제어법칙을 수행한다. 따라서 항공기를 모델링 하는 6자유도 운동방정식, 센서 모델, 대기 모델, 엔진 모델, 작동기 모델 등은 비행제어 시스템의 동작 주기에 비해 최소한 1/2이상의 갱신율을 보장해야 한다.

4.2 시스템 동기화 및 실시간성

모델 기반의 시뮬레이션을 제외한 PILS 시스템은 동기화된 비행제어 시스템과 실시간 연산을 수행해야 한다. 동기화는 시뮬레이션 시스템을 구성하고 있는 구성요소 사이의 데이터가 동일한 시점에 사용될 수 있음을 말한다. PILS 시스템은 비행제어 컴퓨터(DFCC), 센서 및 항공기 모델을 수행하는 시뮬레이터 컴퓨터, 시스템 관리자 및 모니터링 컴퓨터로 구성되고 각각의 컴퓨터는 서로 다른 프로세서를 사용할 수 있고 플랫폼도 상이할 수 있기 때문에 반드시 동기화를 위한 노력이 요구된다. 그림 8은 PILS 시스템의 동기화 신호가 발생하는 작업 순서를 보여준다. 관리자 컴퓨터에서 시스템 초기화 명령을 각각의 프로세서에 전달하면 비동기 초기화 과정이 수행된다. 동시에 시뮬레이터 컴퓨터는 UDP 통신을 이용하여 설정된 주기로 동기화 신호를 비행제어 컴퓨터와 관리자 컴퓨터에 보낸다. 그러면 각각의 프로세서는 시뮬레이터 컴퓨터의 동기화 신호를 통해 자체 실시간 타이머와 비교하여 시뮬레이션 작업을 수행한다.

실시간 연산은 데이터의 전송, 제어법칙 계산 그리고 조종면 작동기 제어 명령이 정해진 시간내에 수행되어야 함을 의미한다. 항공기의 OFP는 정해진 시간내에 정확한 연산 결과가 출

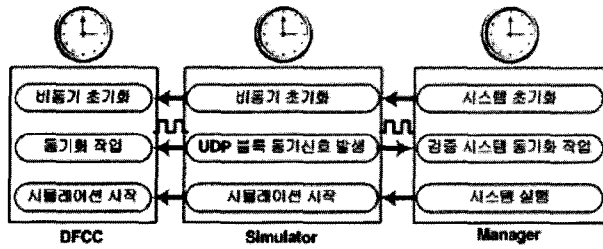


그림 8. Pils 시스템 동기화 과정

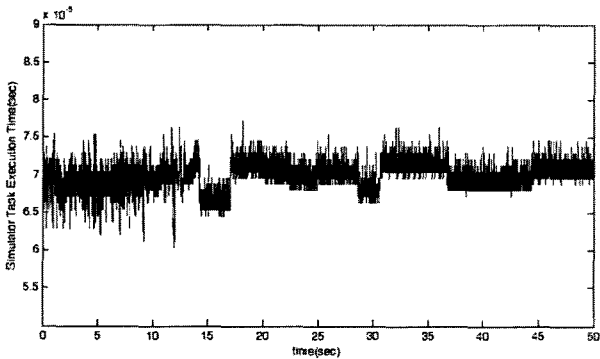


그림 9. 시뮬레이션의 실시간 성능 확인 사례

력 되어야 하므로 대부분 경성 실시간(hard real-time) 시스템으로 구성된다. 이러한 실시간 시스템 구성을 위한 대표적인 운영체제로서 VxWorks등을 사용할 수 있다. 하지만 이러한 고가의 운영체제를 사용하지 못 할 경우 Linux를 사용하거나 또는 MATLAB에서 제공하는 도스 기반의 xPC target을 적용할 수 있다.

4.3 실시간성 검증 사례

모델링 컴퓨터를 xPC target 운영체제로 구성하여 실시간 동작 시험을 수행하였고, 그 결과는 그림 9와 같다. 시뮬레이션 환경은 50Hz로 동작하는 비행제어 컴퓨터를 지원하기 위해 동작 주기의 4배인 200Hz로 설정하였다. 시뮬레이션 컴퓨터에서 동작하는 항공기 모델, 센서 모델의 출력 데이터는 비행제어 컴퓨터에 정의된 데이터 패킷으로 구성하여 UDP 통신을 수행한다. 한 대의 PC에서 이러한 과정을 수행할 경우 한 주기 수행 시간이 10 μ sec가 넘지 않음을 그림 9에서 확인할 수 있다. 이러한 결과에 따라 xPC target 운영체제를 사용하여 Pils를 구성할 때 실시간성이 보장됨을 알 수 있다.

4.4 Pils 시스템 구축 사례

위에 언급된 내용을 바탕으로 그림 10과 같은 Pils 시스템을

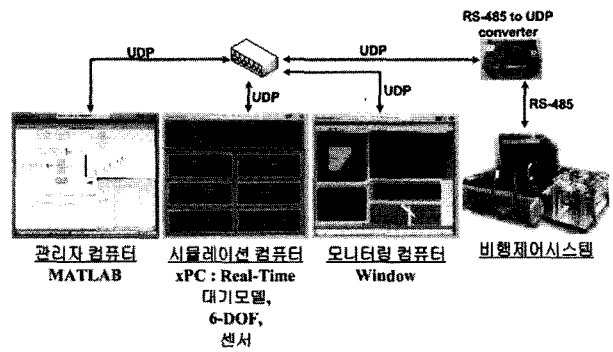


그림 10. Pils 시스템 구축 사례

구성할 수 있다. 구성 요소로는 우측에 OFF가 동작하는 비행제어 컴퓨터가 있고, 중앙 좌측에 xPC target 기반의 시뮬레이션 컴퓨터가 있다. 시뮬레이션 컴퓨터는 항공기 6자유도, 센서 모델, 작동기 모델, 대기 모델 등의 데이터가 실시간 주기에 맞추어 생성된다. 좌측은 Pils 시스템의 동작을 관리하는 관리자 컴퓨터로 MATLAB으로 구성되고 중앙 우측에 윈도우 기반의 모니터링 컴퓨터가 구성된다. Pils 시스템이 동작하는 것은 실제 비행하는 상태와 동일하기 때문에 공개 소프트웨어인 Flight Gear와 연동하여 좀더 현실적인 가시화를 구성할 수 있다. 각각의 컴퓨터는 UDP 통신을 이용하여 데이터를 전송한다. 그림 10의 Pils 시스템에 사용된 비행제어 컴퓨터의 데이터 버스는 RS-485체제를 사용하기 때문에 시뮬레이션 컴퓨터와 비행제어 시스템 사이에 RS-485 to UDP 컨버터를 사용하여 데이터 전송 규격을 통일 시킨다. 일반적인 무인항공기용 비행제어 시스템의 통신 방식은 RS-232, 중고고도 무인항공기와 같이 대형 무인항공기의 비행제어 시스템은 MIS-STD-1553B 등과 같은 유인항공기의 통신체제를 사용하는데 Pils 시스템과 비행제어 시스템 사이에 적절한 데이터 변환 장치를 사용하여 데이터 전송 규격을 일치 시킨다.

5. 발전 방향

유인항공기와 다르게 일반적인 무인항공기의 비행제어 시스템은 하나의 프로세서, 한 무리의 센서 시스템 및 작동기로 구성되기 때문에 단일 구조 시스템에 대한 지상 검증 도구를 구축하여 사용하였다. 하지만 점점 무인항공기의 신뢰성 및 생존성이 중요시 되고 다수 비행체를 이용한 편대비행이 시도됨에 따라 지상 검증 도구도 이에 맞도록 발전하고 있다. 시스템의 신뢰성 확보를 위한 중복구조 비행제어 시스템[15], 고장모드 테

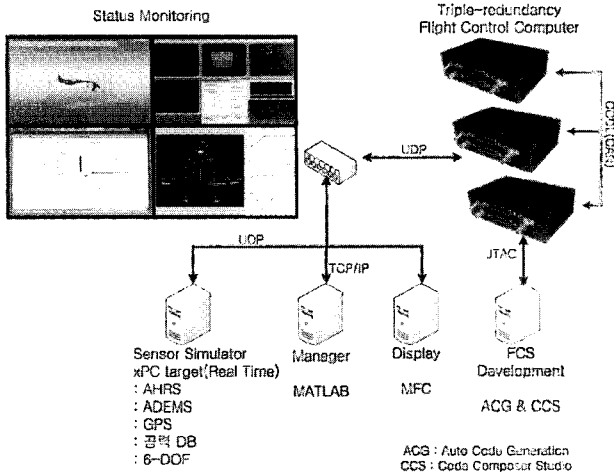


그림 11. 삼중구조 비행제어 시스템 PILS

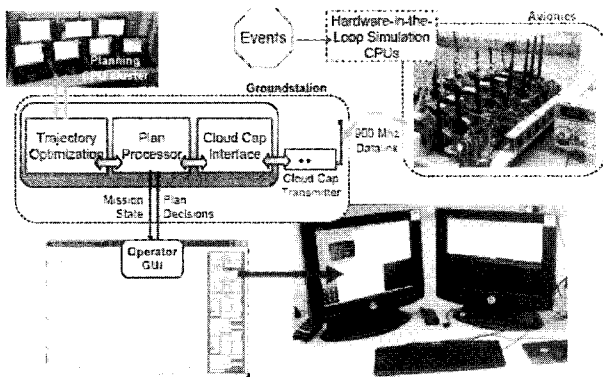


그림 12. 다수(8대) 무인항공기 HILS 시스템

스트(FMET, Failure Mode Effect Test)[16] 수행이 가능한 PILS 시스템을 구축하여 사용한다.

그림 11은 3중으로 구성된 비행제어 시스템의 PILS 시스템의 구축 사례이고, 그림 12는 8대로 구성된 편대비행 시스템의 검증용 HILS 시스템의 구축 사례[17]를 보여주고 있다. 이러한 사례에서 볼 수 있듯이 지상 검증용 시스템으로 사용되는 PILS 또는 HILS 시스템은 어떠한 무인항공기 시스템을 검증할 것인가에 따라 그 형태가 다양하게 구성된다.

6. 결론

현대 무인항공기 개발 단계에 있어서 비행제어 시스템 검증을 위한 PILS 또는 HILS 시스템 구축은 필수 사항에 속한다. 이러한 지상 검증 시스템 구축은 시스템 개발 주체의 역량에 따라

또는 개발 과정에 따라 비실시간 시뮬레이션, 프로세서 기반의 시뮬레이션(PILS), 하드웨어 기반의 시뮬레이션(HILS) 환경을 구축하고 검증 도구로서 활용할 수 있다. 특히, 실시간 시뮬레이션을 수행하는 PILS 시스템은 검증 도구로서 활용되기 위해 시스템 사이의 동기화, 실시간성, 데이터 갱신 주기 등의 요구 조건을 만족시켜야 한다. 그리고 시뮬레이션 도구가기 때문에 최대한 현실성을 만족시킬 수 있는 해석적 모델을 구현하여 실제 비행과 시뮬레이션 사이에 발생하는 불확실성을 최소화 해야 한다.

개인용 컴퓨터의 발전으로 인해 무인항공기 비행제어시스템 검증을 위한 PILS 시스템은 상용 기성품(COTS, commercial off-the-shelf)을 이용하여 충분히 구축할 수 있게 되었다. 이와 더불어 단순히 항공기의 비행제어법칙에 대한 기능 검증뿐만 아니라, 비행제어 시스템의 신뢰성 확보를 위한 중복구조 시스템 검증용으로 활용되고, 편대 비행과 같은 다수 비행체의 지상 검증 시스템으로 확장하여 사용되고 있다.

참고문헌

- [1] M. J. Mahoney and L. Cook, "Development and Operations of Flight Systems for Responsive Missions," *AIAA 2nd Responsive Space Conference*, April 2004.
- [2] P. C. Son, M. C. Song, B. C. Kim, C. K. Oh, and Y. D. Park, "Implementation of PILS for Top-attack guidance and control algorithm," *International Conference on Control, Automation and Systems 2007*, pp. 2096-2099, Oct. 2007.
- [3] 박정진, 장세아, 윤중섭, 유창경, 최기영, 박승훈, "휴대용 전술유도무기 PILS 시스템 구축," 한국항공우주학회 춘계 학술발표회, pp. 307-310, 04. 2008.
- [4] S. Ates, I. Bayezit and G. Inalhan, "Design and Hardware-in-the-Loop Integration for a UAV Microavionics System in a Manned-Unmanned Joint Airspace Flight Network Simulator," *J Intell Robot System*, pp. 359-386, March. 2008.
- [5] A. H. Goktogan and S. Sukkarieh, "Simulation of Multi-UAV Missions in a Real-Time Distributed Hardware-in-the-Loop Simulator," *Proceeding of the 4th International Symposium on Mechatronics and its Applications*, pp. ISMA07-1-ISMA07-06, March 2007.
- [6] 장세아, 김성환, 박종후, 박춘배, 최기영, 이훈희, "무인표적기 비행제어시스템 검증을 위한 PILS 시스템 개발," 한국항공우주학회 춘계학술발표회, pp. 47-51, 04. 2006.

- [7] 인하대학교 항공우주 제어 시스템 연구실, 비행체 특화연구센터 비행제어 연구실 1단계 종결 보고서, 비행체 특화연구센터, 10. 2006.
- [8] 박옥제, "항공기의 실시간 미계수 추정기 설계에 관한 연구," 한국항공대학교 대학원 박사 논문, 02. 2005.
- [9] 정명진, 박미현, 박춘배, 이인석, "SIMULINK를 이용한 항공기 공력-추력 모델 구성," 한국항공우주학회 추계학술발표회, pp. 1200-1203, 11. 2008.
- [10] M. Dubin and S. Teweles, "U. S. Standard Atmosphere," 1976 Stock No. 003-017-00323-0, U. S. Government Printing Office, Washington D. C. USA.
- [11] D. H. Titterton and J. L. Wetron, Strapdown inertial navigation technology, Peter Peregrinus Ltd, 1997.
- [12] www.mathworks.com, Simulink Aerospace Blockset
- [13] www.dutchroll.com, FDC 1.4-Simulink Toolbox for Flight Dynamics and Control Analysis
- [14] www.u-dynamics.com, AeroSim Blockset
- [15] 인하대학교 항공우주 제어 시스템 연구실, 비행체 특화연구센터 비행제어 연구실 2단계 중간 보고서, 비행체 특화연구센터, 07. 2008.
- [16] 윤형식, 변진구, 이기훈, 양지연, 서준호, "비행조종컴퓨터 FMET 수행 방안 연구," 한국항공우주학회 추계학술발표회, pp. 1176-1179, 11. 2008.
- [17] J. How, E. King and Y. Kuwata, "Flight Demonstrations of Cooperative Control for UAV Teams," AIAA 3rd "Unmanned Unlimited" Technical Conference, September 2004.

○ 저자 약력



김 성 수

- 2002년 조선대학교 항공우주공학 공학사.
- 2005년 인하대학교 항공공학 공학석사.
- 2006년~현재 인하대학교 항공공학과 박사과정 재학중.
- 관심분야 : 비행제어 시스템, 신뢰성 공학, 고장 허용 시스템.



유 창 경

- 1989년 인하대학교 항공공학 공학사.
- 1991년 KAIST 항공우주공학 공학석사.
- 2006년 KAIST 항공우주공학 공학박사.
- 1991년~2006년 국방과학연구소 선임연구원.
- 2006년~현재 인하대학교 조교수.
- 관심분야 : 유도무기/무인기 유도제어, 최적제어,

비선형 제어.



최 기 영

- 1987년 서울대학교 항공우주공학 공학사.
- 1989년 서울대학교 항공우주공학 공학석사.
- 1994년 Stanford University 공학박사.
- 1993년~1999년 Advanced Rotorcraft Technology, Inc, Senior Research Engineer.
- 2000년~현재 인하대학교 교수.

· 관심분야 : 최적제어, 무인기 항법/제어, 회전익 시뮬레이션.



박 춘 배

- 1974년 서울대학교 항공우주공학 공학사.
- 1976년 서울대학교 항공우주공학 공학석사.
- 1976년~1980년 공군사관학교 교관(전임감사).
- 1989년 서울대학교 항공우주공학 공학박사.
- 1980년~현재 인하대학교 교수.
- 2007년~현재 인하공업전문대학 학장.

· 관심분야 : 항공전자, 비행제어, 시뮬레이터.