

실시간 방재형 무인비행체의 HIL시물레이션을 위한 환경 개발

정덕원*, 민덕기*

*건국대학교 컴퓨터공학부

e-mail:{dwchung,dkmin}@konkuk.ac.kr

Development of Environment for HIL Simulation of Real-time Disaster-Prevention UAV(Unmanned Aerial Vehicle)

Duckwon Chung*, Dugki, Min*

*School of Computer Science & Engineering, Konkuk University

요 약

재해 및 재난 발생에 따른 피해를 최소화하고자 무인헬기 기술과 IT기술의 실시간 방재 시스템을 융합하여 광범위한 환경을 대상으로 실시간 다각적 정보수집 기능을 제공하는 연구가 진행 중이다. 소형무인헬기에 개량된 자동항법기능과 무선 네트워크 기반의 실시간 멀티미디어 중계기능을 탑재하고 쉽게 사용 할 수 있는 지상관제시스템을 개발하여 대형재난 현장에 사용하면 다각적 동영상과 재해 현장 정보를 실시간으로 제공함으로써 초기 대응을 할 수 있어 재난의 확산을 최대한 방지 할 수 있다. 본 논문에서는 이러한 시스템을 보다 효과적으로 개발하기 위한 HILS(Hardware in the Loop Simulation) 기반 무인헬기 시물레이션 환경을 개발하고자 한다.

1. 서론

최근 세계 각국은 여러 재난으로부터 자국을 보호하는 것을 중요하게 생각하고 있으며, 우리나라 역시 방재 정책 목표를 만들어 자연 재해 및 여러 재해들에 대응하고 있다. 그러나 대형 재난의 경우 사고 보고가 된후 즉각적인 출동이 이루어진다 해도 현장에서의 광범위하고 효과적인 대응이 어려우며, 초기 현장 정보를 원활하게 파악할 수가 없어서 소규모 재난 및 재해도 대규모로 확대 되는 현실이다. 이에 재해 및 재난에 무인헬기와 같은 기존의 시스템과 새로운 IT기술을 접목하여 광범위한 환경을 대상으로 실시간으로 여러 정보수집 기능을 제공하여 보다 체계적이고 효과적으로 대응하려 한다. 본 논문에서는 이러한 무인 비행체가 잘 동작할 1)수 있게 HILS[1][2][3]기반의 시물레이션 환경을 개발하고자 한다. 모델 기반 설계 방법론을 이용하여 재사용이 가능하게 하여 효과적인 설계하였으며, 실시간으로 제어 소프트웨어를 테스트하였다.

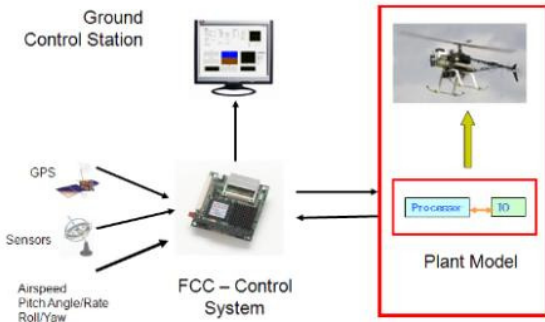
2. 관련연구

RSPE 프로젝트는 무인 소형 헬리콥터에 대한 모델링, 시물레이션, 빠른 프로토타입 개발을 지원함으로써 자율적인 미니 헬리콥터 컨트롤 시스템의 디자인이나 시물레이션과 실시간 구현 환경을 제공한다. 야마하 R-50 프로젝트[4]는 센서와 액추에이터 모델이 시물링크 모듈 모델로 제공되는 야마하 R-50이라는 작은 헬리콥터의 비선형 모델과 통합을 위한 헬리콥터의 시물레이션을 하는 것을 목표로 모델 헬리콥터의 제어 시스템 디자인을 위한 HILS를 적용하였다. FlightGear[5]는 오픈소스 프로젝트로 개발되었으며 다양한 플랫폼에서 개발 및 가용이 가능하고 누구나 확장 및 개질시킬수 이TSms 오픈 비행 시물레이션 프레임워크를 제공한다. MS Flight Simulator X[6]는 2006년 10월 미국에서 출시된 비행시물레이터로 24개의 비행기, 38개의 도시, 45개의 공항을 포함하고 있으며, 시물레이터뿐만 아니라 사용자와 서로 상호작용이 가능한 미션들을 게임을 통해서도 제공하고 있다.

3. 기본형 헬리콥터 설계 및 시물레이션 환경

다음 (그림 1)은 본 논문에서 제안한 실시간 시스템 기반의 시물레이션을 위한 기본적인 헬리콥터에 대한 시물레이션 환경을 위한 구조이다.

1) 교신저자 (민덕기 dkmin@konkuk.ac.kr)
2) 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터지원사업의연구결과로 수행되었음 (IITA-2009-C1090-0902-0026)
3) 본 연구는 지식경제부의 IT성장동력기술개발사업의 일환으로 수행하였음. [2008-S-007-01, 차량 전장용 통합제어 SW플랫폼 개발]



(그림 1) 기본형 헬리콥터 시뮬레이션 환경

기본 구조는 크게 지상 관제 시스템인 GCS(Ground Control Station) 비행제어를 하는 FCC(Flight Control Computer) 그리고 Plant Model로 구성되어 있다. 오른쪽의 Plant는 중간의 FCC에서 정보를 받아서 헬리콥터의 비행에 따른 비행 정보를 수정하여 이에 대한 상태정보를 다시 FCC로 전송하며 FCC는 GPS, 센서 및 비행 제어 매개변수를 받아서 처리하여 GCS에 비행 통합 정보를 전송하며 Plant모델에 비행에 따라 수정된 데이터를 전송한다. GCS는 FCC에서 처리된 Plant 데이터 및 비행에 필요한 기체 상태 정보를 받아 통합된 GUI환경에서 동작 상태 정보를 보여준다.

4. 실시간 UAV의 HILS 환경 구조도

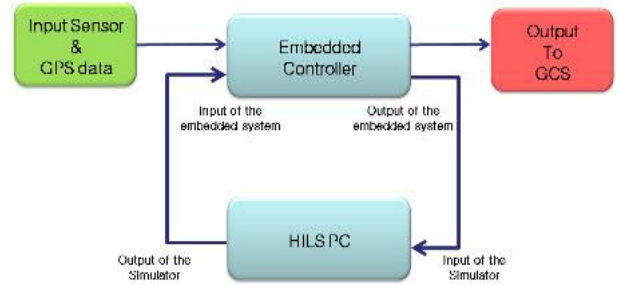
이번 장에서는 HILS를 이용한 실시간 환경 구축을 하기 위하여 필요한 절차를 각 단계별로 설명하고 이를 기반으로 실시간 무인비행체의 HILS 환경 구조를 제시한다.

4.1 SIL(Software-In-the-Loop) Simulation

HILS구축을 위한 호스트 컴퓨터상의 소프트웨어에서의 시뮬레이션 구축단계로서 시뮬레이션 툴의 소스코드는 이미 테스트되어 유효한 호스트 컴퓨터 시뮬레이션을 제공한다. SIL시 시뮬레이션 설계 단계에서 종합적인 데이터 수집과 비행시험 이전의 컴포넌트 테스트등 단일 UAV 플랫폼 비행 테스트와 무인 비행체 운용을 위한 필요한 모듈별로 소프트웨어 단계에서 설계하고 테스트를 한다.

4.2 PIL(Processor-In-the-Loop) Simulation

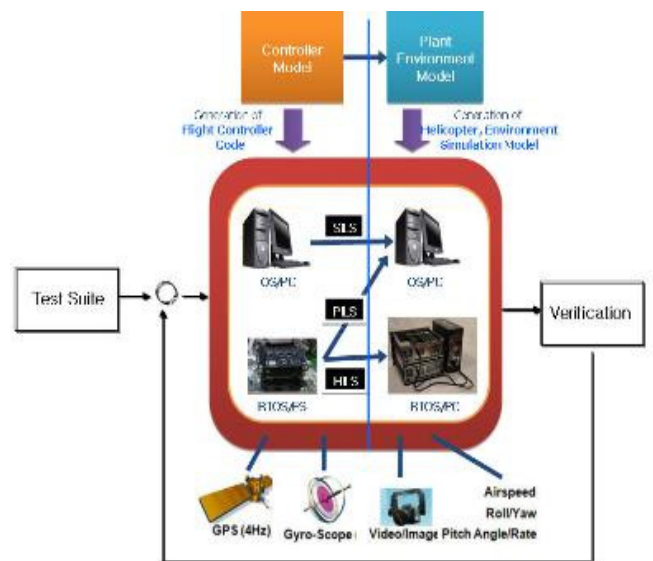
HILS구축을 위한 PC와 임베디드 시스템 (FCC)간의 시뮬레이션 구축 단계로서 AHRS(Attitude Reference Heading System) 패키지를 검증하기 위한 테스트 케이스를 만들기 위해 전송 패킷 순서와 데이터 크기가 필요하므로 두 시스템간의 통신 패킷을 설정한다. 무인 비행체를 위한 GPS 등의 실시간 시뮬레이션 데이터를 연결한다.



(그림 2) HILS 기본 구조

4.3 HIL(Hardware-In-the-Loop) Simulation

위의 (그림2)와 같이 HILS는 무인비행체를 위한 임베디드 시스템들간의 시뮬레이션 구축 단계로서 시뮬레이터의 입력신호와 출력 신호를 갖는 실시간 시뮬레이션은 실시간 프로세스로 같은 시간 의존 값을 나타낸다. 또한 실시간 임베디드 컨트롤 시스템은 서로 다른 실제 사용하중과 조건하에서 테스트한다. 다음 (그림 3)은 위에서 설명한 단계를 기반으로 구성한 실시간 HILS 통합 환경 구조이다.



(그림 3) HILS 통합 환경 구조

5. RTOS QNX기반의 UAV 시뮬레이션 환경구축

이번 장에서는 지금까지 설명한 내용을 기반으로 실제로 임베디드 보드와 실시간 운영체제인 QNX를 사용하여 시뮬레이션 환경 개발과 필요한 시스템 요구사항에 대하여 설명한다.

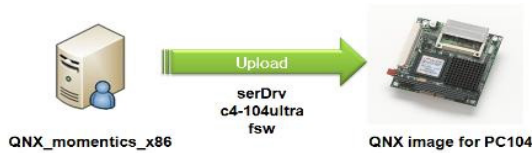
5.1 QNX와 HILS를 위한 시스템 요구사항

Host PC에 사용할 QNX OS (Momentics 6.2.1): HILS를 설정하기 위해 Octagon PC-104 보드에 QNX OS를 포팅하며 호스트 PC 역시 QNX 개발 툴킷 설치하며 이는 타겟 시스템에 배치되는 Plant와 Control System Code를

개발하고 컴파일한다. 비행제어를 위한 코드는 보드에 로드되어 사용한다. **Plant Model:** Plant Model을 위해 QNX OS (Neutrino 6.2.1)와 한 대의 PC를 사용하며 여기에 HILS코드가 로드되어 실행된다. **GCS:** indow XP SP2 이상, Microsoft Visual C++ 6.0, Measurement Studio 6.0, GMS Aircraft instrument Active X Control Ver 3.2, GCS code module 이 필요하며 비행체의 동적변수를 분석하는데 사용하고 비행체를 컨트롤 한다.

5.2 FCC 시스템 구축

다음 (그림 4)와 같이 PC104 옥타곤보드를 이용한 QNX 기반 시스템을 구축(FCC)하여 호스트 QNX IDE에서 컴파일된 FSW 파일을 PC104보드로 업로드하고 시리얼 포트 드라이버를 구동하여 FCC 구동환경을 세팅한다.



(그림 4) FCC 시스템 구축

5.3 Plant 시스템 구축

Plant 시스템은 다음 (그림 5)와 같이 호스 QNX IDE에서 컴파일된 HILS 파일을 업로드 하고 역시 시리얼 포트 드라이버를 구동하여 세팅한다.



(그림 5) Plant 시스템 구축

5.4 FCC 시스템과 Plant 시스템 연동

다음 (그림 6)와 같이 FCC와 HILS간 시리얼 통신을 위해 RS-232 케이블 연결하고 FCC에서 HILS로 컨트롤 데이터를 보내고 HILS를 통해 얻어진 Signal 데이터를 다시 FCC로 재전송한다.



(그림 6) FCC 시스템과 Plant 시스템 연동

5.5 FCC 시스템과 GCS 시스템 연동

다음 (그림 7)와 같이 GCS와 HILS간 통신을 위해 RS-232 시리얼 통신을 하며 FCC 모듈을 통해 얻은 시그

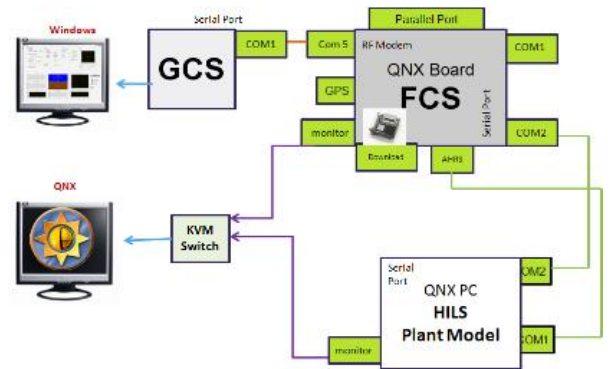
널 데이터를 GCS에 전송하여 현재 FCC 상태와 지상 관제 시스템에 필요한 값을 확인한다. 이들 사이의 통신은 GCS가 PacketAdt 패키지를 보내고, GCS가 FCS의 옥타곤 보드로부터 롤, 피치, 높이, 속도 등의 비행 파라미터를 받음으로써 이루어지게 된다.



(그림 7) FCC 시스템과 GCS 시스템 연동

5.6 QNX HILS 실행

QNX HILS구현은 옥타곤 보드와 PC를 사용하여 시뮬레이션 하는 방법과 두 개의 옥타곤 보드를 이용하는 방법이 있다. 먼저, Host QNX IDE로 코드 개발 완료 후 FCC, HILS, GCS 구동을 위한 환경설정을 한후 각각의 RS-232 시리얼 케이블을 각 디바이스에 연결한 후 시리얼 포트 드라이버를 실행한다. 그리고 FCC 컴파일 모듈을 실행하여 GCS에서 시그널을 확인한 후 HILS를 구동하여 테스트한다.



(그림 8) 전체 시뮬레이션 환경

6. 시뮬레이션 결과

동작중인 무인 비행체의 비행 데이터는 매우 복잡하며 또한 강한 동력회전을 하고 비선형 비행을 하며 무인 비행체의 동작에 대한 다음 (그림 9)와 같은 효율적인 특정 값(Roll, Pitch, yaw)들을 도출해낼 수 있다.

참고문헌

[1] Gomez, Martin, Hardware-in-the-Loop Simulation, Embedded System Design, 2001, URL <http://www.embedded.com/showArticle.jhtml?articleID=15201692>

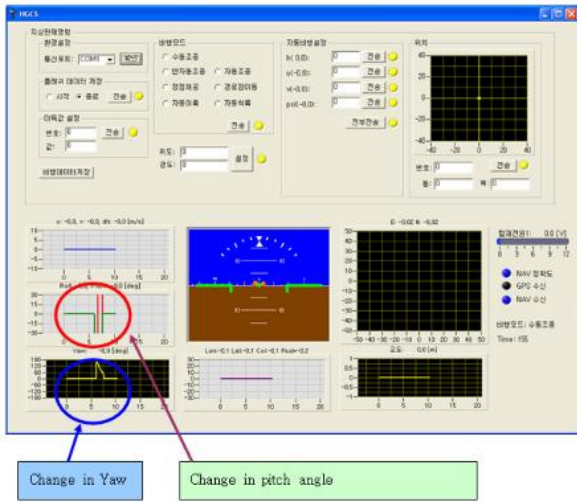
[2] Desbiens, Andre and Manai, Myriam,, Identification of a UAV and Design of a Hardware-in-the-Loop System for Nonlinear Control Purposes, Université Laval, Quebec City, Canada

[3] Sanvido Marco,, Hardware-in-the-loop Simulation Framework, Automatic Control Laboratory, ETH Zurich

[4] Muhammad Lutfi, Agus Budiyo, H.Y Sutarto "Hybrid Simulation For Safety Investigation Of Embedded Control Yamaha R-50 Helicopter Flight Control System"

[5] <http://www.flightgear.org/>

[6]<http://www.microsoft.com/games/pc/flightsimulatorx.aspx>



(그림 9) 비행시작 시 FCC에 전달되는 신호

비행 상태일 때 모든 파라미터가 정상적인지를 검사할 수 있으며 안정적 비행 상태를 유지할 수 있도록 컨트롤 가능한 데이터를 보여주며, 다음 (그림 10)와 같이 고도를 30으로 자동 비행 설정을 각각 $u=10$, $v=15$, $\psi=25$ 로 변경 후 위도 경도 및 FCC, GCS로 전송되는 데이터와 GCS의 네비게이션에서 받는 데이터가 변경되는 걸 확인할 수 있다.

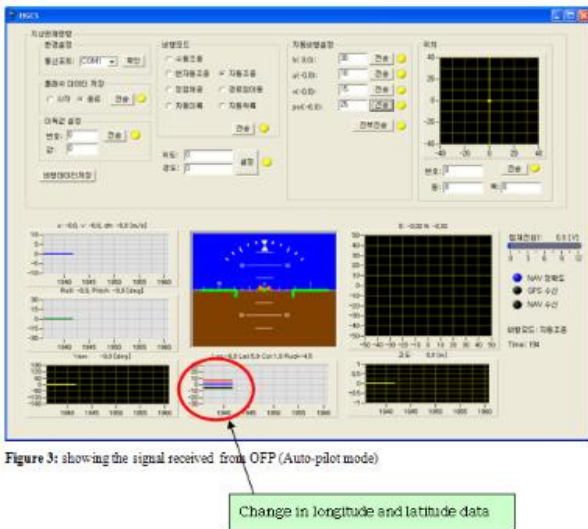


Figure 3: showing the signal received from OFP (Auto-pilot mode)

(그림 10) 자동 비행 모드로 FCC에서 얻은 신호

7. 결론 및 향후과제

재난방지를 위한 무인비행체와 IT기술을 접목하는 시스템의 효과적인 개발을 위한 시뮬레이션 환경을 본 논문에서 제시하였다. 앞으로 시뮬레이션이 아닌 실제 무인 비행체에 탑재될 FCC 시스템을 개발하고 GPS 센서를 탑재하여 실제 GPS값을 받아서 처리할 수 있게 할 것 이고, 전체 시스템을 완성하여 불가피한 자연재해나 사고 시에 능동적으로 대처하여 최소한의 피해를 얻도록 할 것이다.