Mission Control over Multi UAVs in the Real-time Distributed Hardware-In-the-Loop Environment

Soe Linn Htet[1,*], Tin Naing Lat[2], Ye Chan[3], Soe Mya Mya Aye[1] and Pho Kaung[3]
[1]*Department of Computer Studies, University of Yangon*
[2]*Department of Physics, Loikaw University*
[3]*Universities' Research Centre, University of Yangon*
**E-mail: fulcrum.slh@gmail.com*

Mission control has been designed for Multiple Unmanned Aerial Vehicles (Multi-UAVs) and the overall control performance verified by a real-time distributed Hardware-In-the-Loop (HIL) environment. Four modules - onboard hardware, flight control, ground station and mission control software, have been integrated to form a framework under which the HIL test is realized. This design is successfully utilized for several flight tests including basic flight motions, full-envelope flight and multiple UAV formation flight. Results show that the constructed HIL verification system is highly effective and useful tool for UAV mission operations such as disaster management, civil security applications and military operations.

1. Introduction

Unmanned Aerial Vehicles (UAVs) have been widely used not only in military applications, but also in civilian and commercial applications. These include (but not limited to) environmental monitoring, natural resource assessment, wildlife monitoring, bushfire monitoring, search and rescue,telecommunication, precision agriculture, power line inspection, and pipeline patrol.

UAVs require human guidance to varying degrees and an Unmanned Aerial System (UAS) is essentially defined often through several operators. The challenge in achieving effective management of multiple UAV (multi-UAV) -systems in the future is to determine not only whether automation can be used to improve human and system performance but also how and to what degree across hierarchical control loops, the types of decision support the needs of operators in the high-workload environment. This research addresses when and how increasing levels of automation should be incorporated in multi-UAV systems and discusses the impact on human performance and more importantly system performance. It has developed hierarchical control algorithms and a control architecture for multi-UAV systems and extensively tested in simulations with various fidelities, from low fidelity proof of concept simulations, to high fidelity Hardware-In-the-Loop (HIL) simulations.

2. Architecture of Single UAV Control

The architecture of single UAV control operation is hierarchical as represented in Figure 1. The innermost loop represents the basic guidance and motion control, which is the most critical loop that must obey physical laws of nature such as aerodynamic constraints for UAVs. In this loop, operator actions are focused only on the short term and local control (keeping the aircraft in stable flight), and generally human control in this loop requires skill-based behaviors that rely on automaticity.

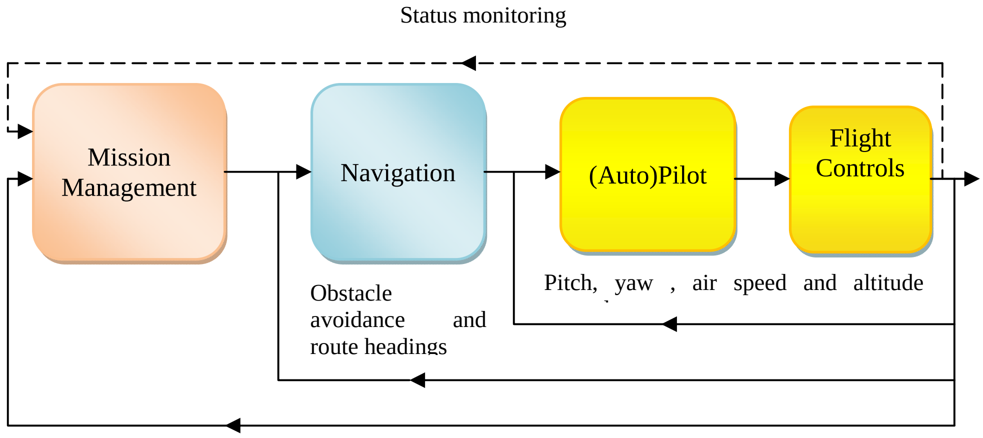
Status monitoring
Mission Management
Navigation
(Auto)Pilot
Flight Controls
Obstacle avoidance and route headings
Pitch, yaw , air speed and altitude

Fig.1 Architecture of UAV Control Operation

The second loop or the navigation loop represents the actions that some agent, whether human or computer-driven, must execute to meet mission constraints such as routes to waypoints, time on targets, and avoidance of threat areas and no-fly zones. The outermost loop represents the highest levels of control, mission and payload management. In this loop, sensors must be monitored and decisions made based on the incoming information to meet overall mission requirements. In this loop, decisions require knowledge-based reasoning that includes judgment, experience, and abstract reasoning that cannot be generally performed by automation.

2.1 Flight Control System

The hardware module consists of the following components: (1) an onboard computer processing system; (2) flight sensors (3) actuators and (3) a wireless modem as shown in Figure 2. Sensors are used to determine the state of the aircraft. Microprocessor analyzes the data attained from the sensors along with commands from ground control radio signals to achieve suitable flight control according to equations of motion. Controls determined by the processor are then used to adjust the flight control surfaces, through the actuators. The control surfaces include ailerons, elevators and a rudder to control the roll, pitch and yaw of the aircraft. The microcontroller does most of the calculations and decision making. In our proposed framework, flight control execution is done by the onboard computer system.

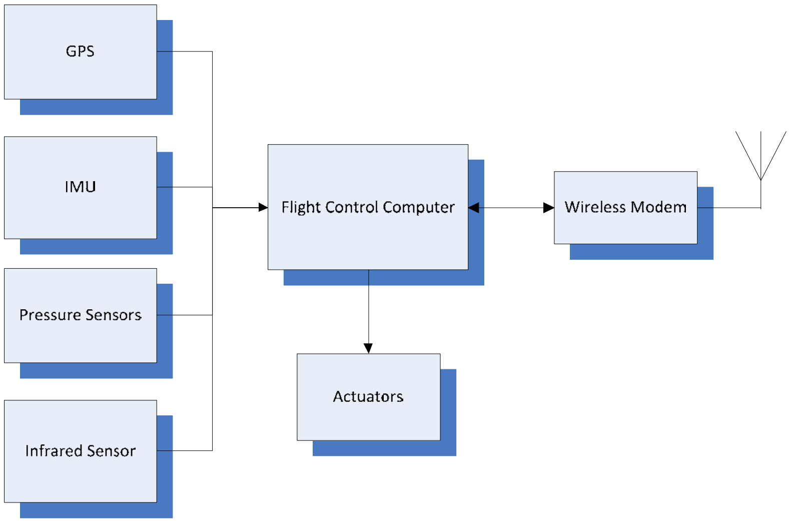

Fig.2 Basic Components of Flight Control System

2.1.1 Controller Design

There are three basic controllers for the UAV flight control which are (a) Attitude controller shown in Figure 3 and Figure 4, (b) Navigation controller in Figure 3 and (c) Altitude and Speed controller in Figure 4 and Figure 5. Attitude controller maintains the aircraft at desired position and altitude and speed controller holds the UAV to fly at a desired altitude and stabilizes the speed in level flight in landing and in take-off. The guidance and navigation controller makes the UAV fly through the desired waypoints.

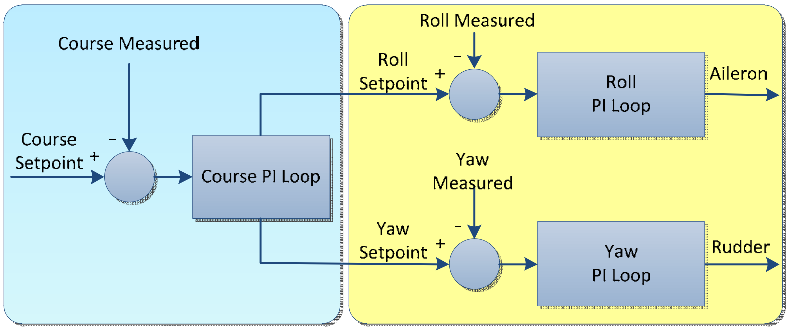

Fig.3 Navigation Control (Outer Loop) and Roll/Yaw Control (Inner Loop)

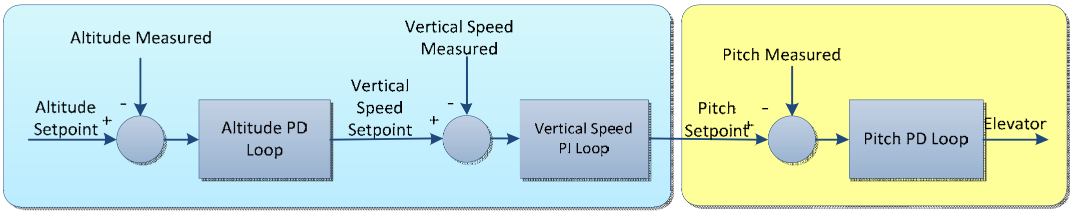

Fig.4 Altitude Control (Outer Loop) and Pitch Control (Inner Loop)

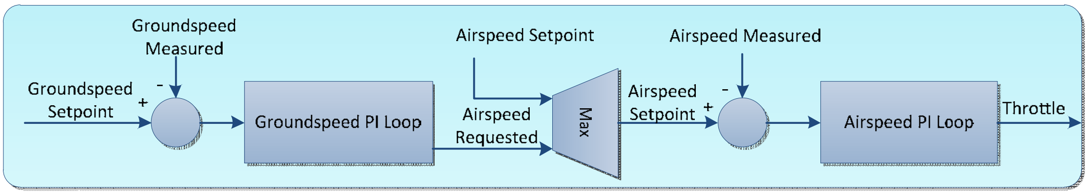

Fig.5 Speed Controller (Outer Loop)

2.1.2 Embedded Firmware

The common methods for embedded application developing environment are super-loop design and thread-oriented design. In this work, the thread-oriented design is used because it has advantages over the super loop design and suitable for more complex applications. With a real time operating system (RTOS), the developer gets not only the tool to create threads, but also tools to communicate between the threads within their real-time constraints.

2.1.3 Thread-Oriented Design of Flight Control System

ChibiOS/RT RTOS is used to develop a flight control system (FCS) on STM32F405RGT6 microcontroller supporting multiple architectures and released under the GPL3 license. It is designed for embedded applications on 8, 16 and 32 bit microcontrollers where size and execution efficiency are their main requirements. Thread-oriented design of FCS is depicted in Figure 6. It is represented for both typical FCS and simulated FCS for HIL testing. There are five groups of thread in its firmware.

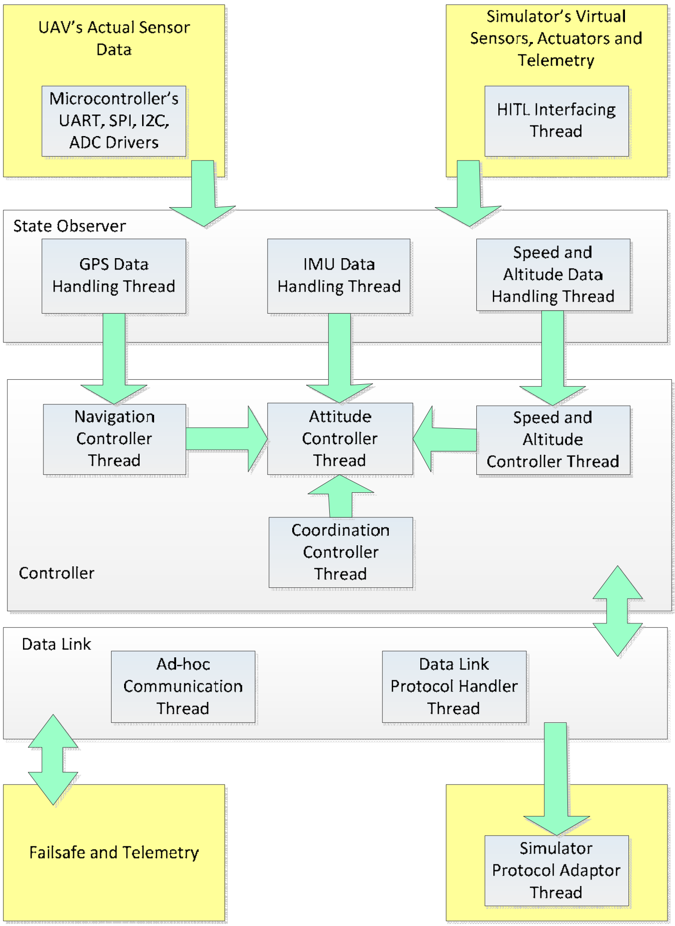
UAV's Actual Sensor Data
Microcontroller's UART, SPI, I2C, ADC Drivers
Simulator's Virtual Sensors, Actuators and Telemetry
HITL Interfacing Thread
State Observer
GPS Data Handling Thread
IMU Data Handling Thread
Speed and Altitude Data Handling Thread
Navigation Controller Thread
Attitude Controller Thread
Speed and Altitude Controller Thread
Coordination Controller Thread
Controller
Data Link
Ad-hoc Communication Thread
Data Link Protocol Handler Thread
Failsafe and Telemetry
Simulator Protocol Adaptor Thread

Fig.6 Thread-Oriented Design of Flight Control System

2.2 Ground Control System

Ground control can be divided into three sub systems. They are (1) Flight Instruments, (2) Map and (3) Communication Status. All flight data from FCS via communication system are displayed on the fly on the flight instruments while real time aircraft trajectory and mission status are updated on the localized Map. Task planning and mission control can also be performed on the map system. Communication system, the heart of the entire system, takes responsibility for data transferring between FCS and ground control system (GCS).

In both flight time and pre-flight time, mission controller can plan the missions for all UAVs on the Graphical User Interface (GUI) of the GCS and hence, the communication backbone must be robust. Moreover, weather condition of operation area can also be obtained from the weather server via this link.

2.2.1 Flight Instruments

Flight instruments include Artificial Horizon, Speed Indicator, Altimeter, and Direction Indicator. These instruments can connect to communication system and monitor the flight data of entire flight.

2.2.2 Map System

In GCS, implementation of map system is based on the Google tile map engine. Figure 7 shows the structure of Google tile image.

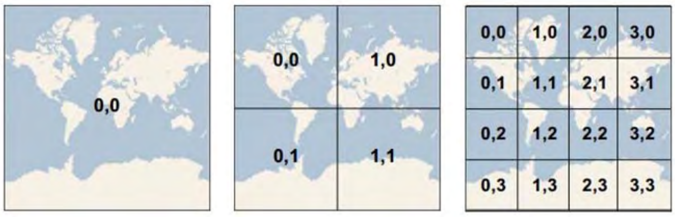

Fig.7 Google Tile Image Structure

Google holds the world in a number of 256*256 pre-rendered images ("tiles") for about 18 zoom stages. The lowest zoom is 0 and the highest 17. At zoom 0, the entire world is kept in one single tile. At zoom 17, the world spreads over giant 17,179,869,184 tiles. The number of tiles needed to cover the entire world for a zoom stage can be calculated with the relation, Number of Tiles = 2 ^ (2*Zoom).

Google uses Mercator projection. There are a couple of algorithms out in order to determine bitmap pixel coordinates for a given geo-coordinate and zoom-stage. The following relations are not only simple but easy to understand the solution of this problem.

$$X = \lambda - \lambda_0$$
$$Y = \frac{1}{2}\ln\left(\frac{1+\sin(\Phi)}{1-\sin(\Phi)}\right)$$

where,

λ = Longitude
λ_0 = Longitude in center of map
Φ = Latitude

There is a simple HTTP request that can be sent to Google Maps servers to obtain these tiles. This is an example of the HTTP request sent to Google:

http://mts0.googleapis.com/vt?lyrs=m&x=48&y=96&z=8

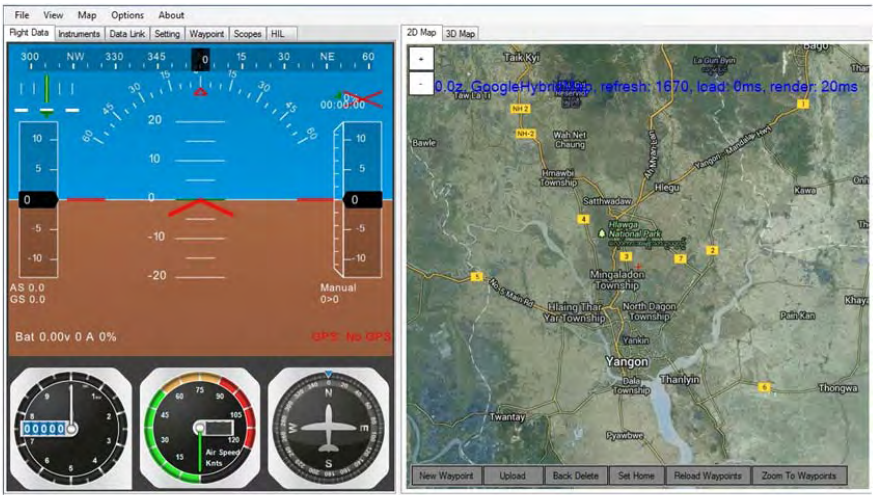

Fig.8 Ground Control System with Customized Map

3. Multi-UAV System Architecture

The hardware architecture of a typical multi-UAV system is illustrated in Figure 9. The main components in the ground station are the Ground Control Stations (GCSs), Mission Controller (MC), weather station and uplink controllers (GBB). During the taxi, takeoff and landing, human pilots control the UAVs with modified hand held Radio Control (R/C) units.It can send the servo control signals to the uplink controllers instead of transmitting them directly to the UAVs. An uplink controller is a unit built around an embedded microcontroller and a spread spectrum radio modem.The GCS for each UAV displays and logs the real-time telemetry data and monitors the data related to mission and cooperative control.

3.1 Mission Controller

Mission controller (MC) is a Ground Control System integrating some authority featuresMission controller has some authority to interact the entire flight and ground system such as task planning, dynamically changing the mission objectives and all other high level decisions. Whereas the low level ground control system can control only its predefined UAV, the mission controller can control all the UAVs and the low level ground control system.

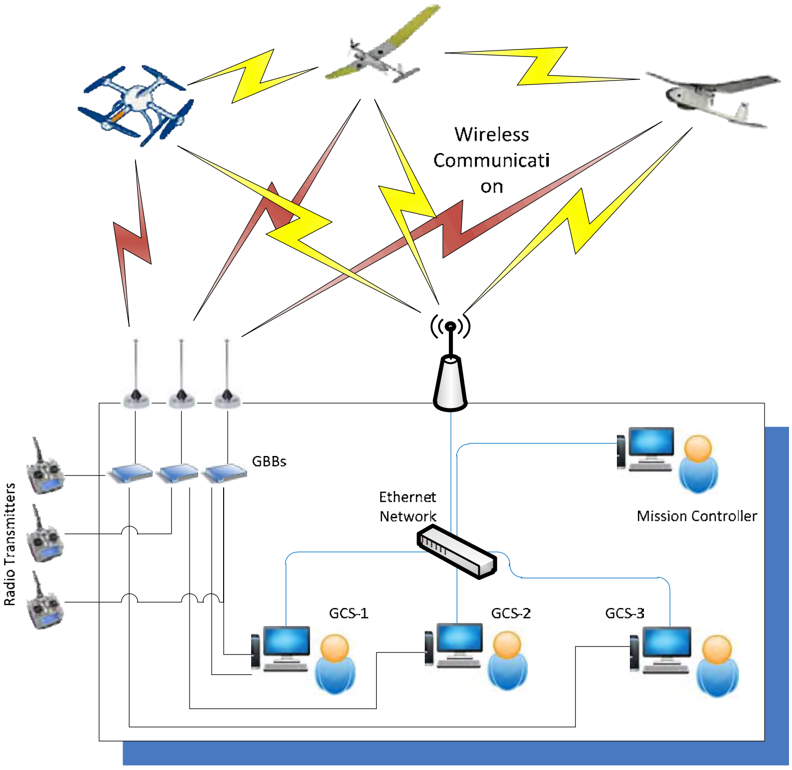

Fig.9 Typical Multiple UAVs Control and Coordination System

3.2 Hardware in the Loop Simulation (HIL)

Figure 10 illustrates the architecture of the distributed real-time HIL test for multi UAVs system. It consists of a number of computers networked with the hardware and software components of the UAVs. In order to reflect the real operational system configuration, the distributed HIL system is designed in such a way that it has almost identical architecture with the system hardware of the real flight missions. Ethernet network (Wi-Fi 801.11b) is used as the communication backbone of HIL system. To perform the HIL test, FlightGear flight simulator is run on each GCS computer of UAVs and it generates the flight data and weather condition of the operation area. These computers also concurrently run flight vehicle dynamics models to simulate the UAV motions.

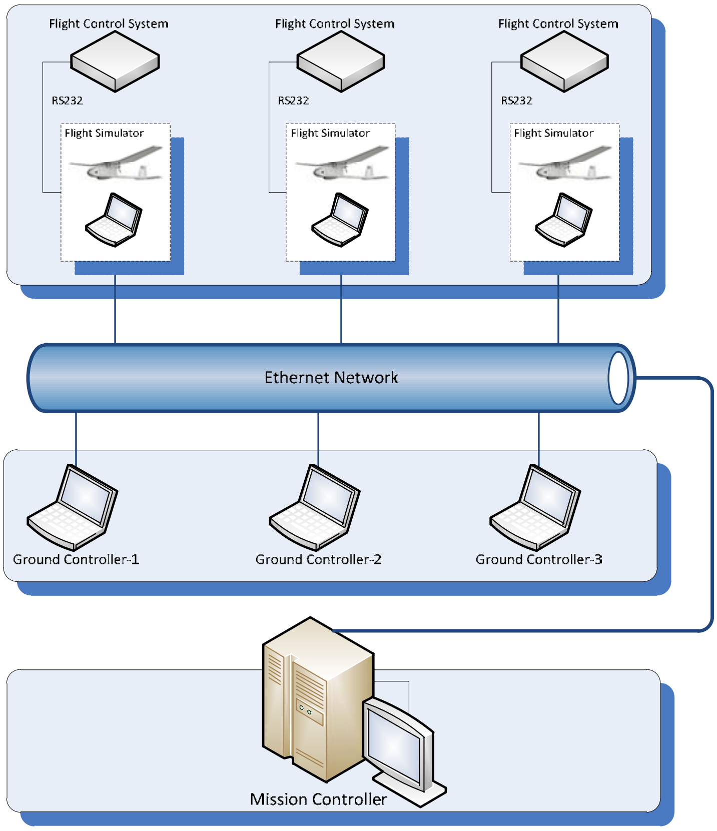

Fig.10 HIL Simulation Network

4. Results and Discussion

In this HIL simulation test, three UAVs are flown in a same flight path at different height levels as shown in Figure 11. They can coordinate each other at specified height levels and be able to avoid the collisions.The results have clearly indicated that the proposed multi-UAV control and coordination system is capable of accurately and efficiently realizing the HIL trial in actual field. Figure 12 shows how accurate each UAV maintained the specific altitude at single flight level.

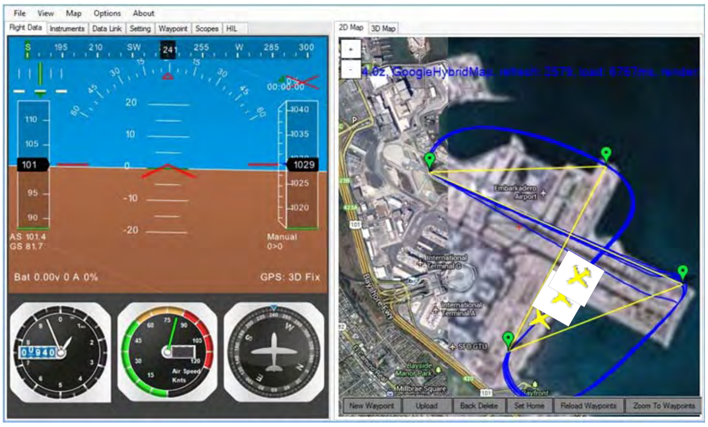

Fig.12 Coordination Flight of Three UAVs on Single Flight Path

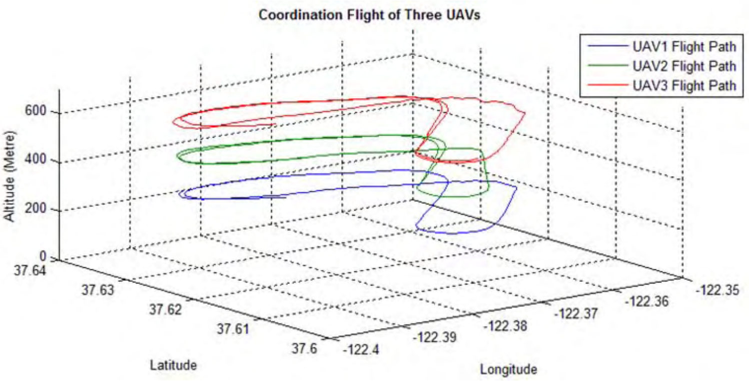

Fig.13 Comparison of Three UAVs' Altitude and Flight Path Coordination

5. Conclusion

The multiple UAV control and coordination system and the complete design of an HIL simulation system have been addressed. The basic motion flight test and the multiple UAV formation test have been carried out at the specific operation area with interactive weather conditions. The results have clearly indicated that the designed HIL system is capable of accurately and efficiently predicting the real flight situations including potential dangers and accidents. The system is certainly an effective and useful tool for UAV mission operations such as disaster management, civil security applications and military operations.

Our system is especially designed for rescue mission in disaster like Cyclone. UAVs are used to assess the situation at first because in rescue situations time is an important factor. Thus, the mission control on the ground station computes initial flight routes for every UAV which can be adopted during mission execution either by sending updated plans from the ground station to the UAVs or by the UAV's on-board drone control. High-resolution images of the affected area will be taken by UAVs within minutes. After the primary danger has been averted, the UAVs will have been used to monitor the progress of securing the operation area by updating the overview image periodically.

References

[1] A HGöktoğan, SSukkarieh, “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 (ISMA07), Sharjah, U.A.E. March 26-29, 2007

[2] Caccia Claudio, “ChibiOS-EmbrIO, Embryo Virtual Machines running on Chibi RTOS”, September, 16 2012

[3] Darkoman, “GoogleMapsNet - GoogleMaps Control for .NET”, http://www.codeproject.com/Articles/463434/GoogleMapsNet-GoogleMaps-Control-for-NET, 21 Sep 2012.

[4] GuoweiCai, Ben M. Chen, Tong H. Lee, Miaobo Dong, “Design and implementation of a hardware-in-the-loop simulation system for small-scale UAV helicopters”, Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576, Singapore, 2009.

[5] Jerry Farmer, “Real-Time Operating System for PICmicro™ Microcontrollers”, Microchip Technology Inc, 2002.

[6] Markus Loecher, “Overlays on Google map tiles in R”, Berlin School of Economics and Law (BSEL), 2013-08-27.

[7] Mats Pettersson, “Basic Concepts for Real Time Operating Systems”, IAR System

[8] Richard E. Kowalski, Frank Huy, “Real-Time Operating Systems”, NASA.

[9] Robert L BorettiJr, Breakdown - HTTP Request Headers, WebSphere Application Server L2 Support, IBM Software Group, 2002.

[10]Wikipedia: Chibios/rt. http://en.wikipedia.org/wiki/ChibiOS/RT

(Received; 23rd December, 2013, Accepted; 19th April, 2014)