

# HARDWARE-IN-THE-LOOP TEST PLATFORM FOR A SMALL FIXED WING UNMANNED AERIAL VEHICLE EMBEDDED CONTROLLER

Onur B. Ertin<sup>1,a</sup>, Halim Korkmaz<sup>1,b</sup> and Coşku Kasnakoglu<sup>1,c</sup>

<sup>1</sup>Department of Electrical and Electronics Engineering, TOBB ETU  
06560, Ankara, Turkey

<sup>a</sup>bertin@etu.edu.tr, <sup>b</sup>hkorkmaz@etu.edu.tr, <sup>c</sup>kasnakoglu@etu.edu.tr

## ABSTRACT

Hardware-in-the-loop (HIL) test systems for unmanned aerial vehicles (UAV) embedded flight controllers provides more realistic test environments for pre-flight tests.

The HIL test system which is described in this paper combines a detailed and sophisticated flight simulation software with a moving hardware platform in the pitch and roll axis with actual flight sensors to form a complete testing environment.

## KEY WORDS

UAV, Hardware-in-the-loop simulation, HIL, Autopilot, PID controller.

## 1. Introduction

From target acquisition and surveillance missions to agricultural spraying, unmanned aerial vehicles (UAV) have taken part in lots of different military and civilian projects [1-2]. Because of reducing costs and increasing efficiencies of components like batteries, electric motors and microcontrollers, interest for UAV research has started to grow rapidly [3-4].

To accomplish missions outside the teleoperation range and minimize human errors, the need for autonomous control units has become inevitable for UAVs. Nowadays UAVs are able to do autonomous missions like taking off, landing and cruising between pre-defined waypoints [5-6].

Flight tests are the most comprehensive way of autopilot testing. However due to equipment costs and environmental effects, pre-flight tests are indispensable. Hardware in the loop (HIL) test systems put embedded autopilot hardware and simulation software together and provide a great environment for pre-flight tests [7-9].

This paper introduces a new and reliable HIL test environment developed for the purpose of testing our flight equipment and autopilot algorithms. An example embedded software design developed by using this test environment is also presented.

## 2. Hardware-in-the-loop (HIL) System Overview

Infrastructure of HIL systems are based on numerous different hardware and software elements. In the present work, a flight simulator software, two axes moving platform, microcontrollers and sensors are integrated to build a reliable HIL system.

Because of realistic flight dynamics, advanced weather events modeling capabilities and over network accessible data structure, Xplane 10 is preferred as the flight simulator software. A custom platform moving around the pitch and roll axes is constructed with lightweight wooden material. Two 37g standard servo motors are mounted to the platform and are connected to the moving surfaces. To drive the servo motors within the platform, we use an Arduino Mega microcontroller. Our own PID algorithm is placed on an Arduino-based Ardupilot Mega card which contains an internal inertial measurement unit (IMU).

Since the plane and weather models are at the heart of the UAV controller HIL simulation, Xplane plays a vital role in our system and whole system runs around it. Two C# applications we develop, maintain connection between Xplane and the microcontrollers via user datagram protocol (UDP) and over universal serial bus (USB) virtual serial ports. Ardupilot Mega card transmits user inputs and PID outputs to the first C# application. On the other side, the second C# application collects pitch and rolls angles of the plane from Xplane and sends these data to Arduino Mega card to actuate servos. Block diagram of the system is shown in Fig. 1.

## 3. Simulation Software and the Test Platform

Using an out of a box complete flight simulator software comes with many advantages such as the user being able to manipulate air conditions, change the direction and speed of the wind, specify the take off location of the plane on the earth and most importantly select a proper plane model among a wide range of models. In Xplane it is possible to adjust all of these settings and select the plane model through a graphical user interface (GUI). Working with ready-to-use models lets us skip modeling

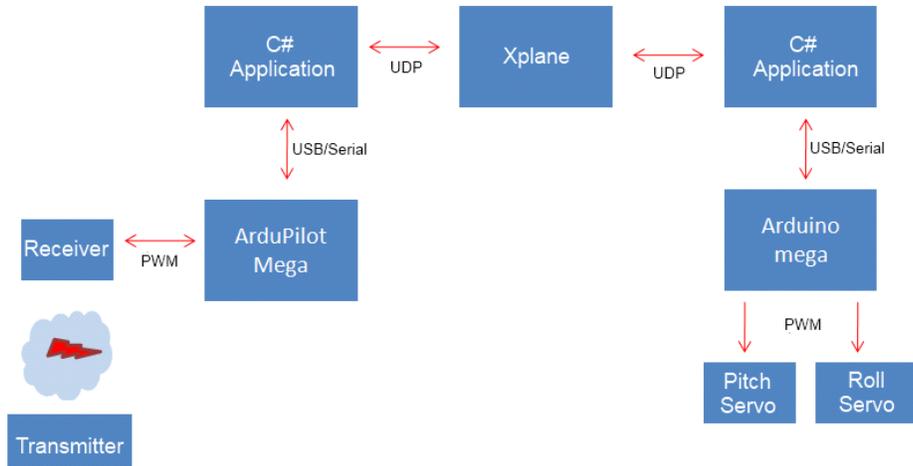


Figure 1. HIL System Block Diagram

process and gives us the flexibility to change the plane model at any time.

Low weights, low costs, high payloads and durability makes remote controlled (RC) model planes perfect platforms to turn into small size UAVs. Therefore, a very common trainer type RC model plane PT-60 is selected as our Xplane plane model. Since PT-60 is a four channel plane we are able to control aileron, elevator, rudder surfaces and throttle. PT-60 has 1.8 meters wingspan and enough payload to carry embedded controller and sensors. Xplane simulation window and PT-60 plane can be seen in Fig. 2.



Figure 2: Xplane Simulation Window and PT-60 plane

Xplane users can view many different flight data from flight angles and coordinates to engine rpm. Also Xplane lets user transfer these data to any other application which can communicate over UDP. Xplane uses datasets of length 41 bytes to receive and send data. There are more than 100 different datasets in Xplane. After first 9 bytes which contain dataset number and data type rest 32 bytes can carry 8 variables. Each variable

consists of 4 bytes in IEEE 754 single precision binary floating point format. A C# application which is connected to Arduino Mega microcontroller, listens to Xplane's UDP send data port, captures datasets and turns them into meaningful pitch and roll angle values. After the conversion process, the application sends angles to Arduino over serial port. The Arduino is programmed to receive angle data over serial port and drive the motors to these angles therefore moving surfaces of the platform.

The moving platform transfers pitch and roll movements in the simulation to the physical world. Placing sensors on top of the moving surfaces gives us a chance to include real sensor data into our system. With real sensors and the embedded controller card, a real plane-like environment is developed. The platform and the microcontrollers can be seen in Fig. 3.

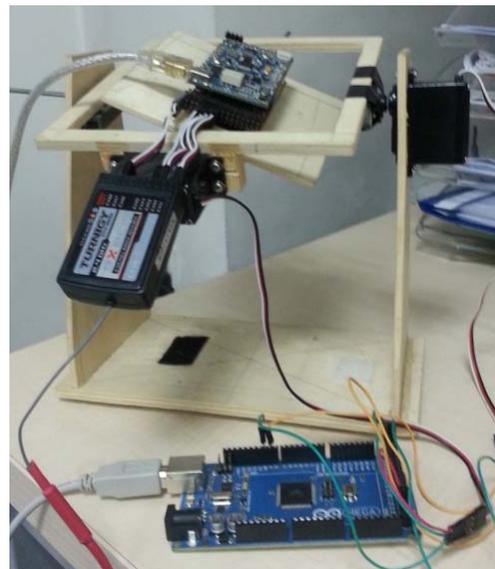


Figure 3. Test Platform and Microcontrollers

To manage the control surfaces of the plane in the simulation a second C# application is used. This application communicates with Ardupilot Mega card and according to control outputs of the card it creates a dataset. After the Xplane-formatted dataset is created, the application sends these data through Xplane's UDP receive data port.

#### 4. Embedded Controller

To show reliability of the system, an embedded controller software is developed. The embedded software runs on the Ardupilot Mega which is a specialized card for autopilot research. Ardupilot Mega's internal IMU contains 3 axis gyroscope and 3 axis accelerometer sensors. Additionally, the open source libraries of Ardupilot Mega give us a chance to use the Ardupilot as an attitude and heading reference system (AHRS). Through AHRS, raw sensor values can be easily converted into pitch, roll and yaw angles.

The main purpose of the embedded controller is stabilizing pitch and roll axes of the plane at zero degrees. Two individual PID controllers are used for this process. While one PID controller works for pitch axis stabilization, the other one stabilizes roll axis. The pitch axis PID controller is directly connected to elevator channel of the plane and the roll axis PID controller is directly connected to aileron channel of the plane. The PID controllers take AHRS data as feedback. Since we

only develop this embedded controller software for test purposes, PID coefficients are picked and optimized manually.

The software on the Ardupilot starts with sensor calibration routine which takes about 15 seconds. A standard remote controller is used to send user inputs to the embedded system. User inputs for the rudder and throttle channels directly go to the rudder and throttle outputs. However, user inputs for the elevator and aileron channels directly go to the elevator and aileron outputs if only user moves the elevator and aileron sticks. When user releases the related sticks the PID controllers become enabled and start to control the aileron and elevator outputs.

#### 5. Results, Conclusions and Future Works

HIL simulation systems for plane controllers reduce the number of flight tests and give researchers a chance to make some further improvements before flights. Expanding HIL system with a moving platform and including real sensors make these tests more reliable furthermore provide far better and realistic results. The HIL test system which was presented in this paper combines a flight simulator software, an embedded controller and real sensors, and subsequently provides a complete test platform for UAV controllers. Also an embedded controller software was designed to illustrate the performance of our system. The embedded controller's pitch and roll axis stabilization behavior is

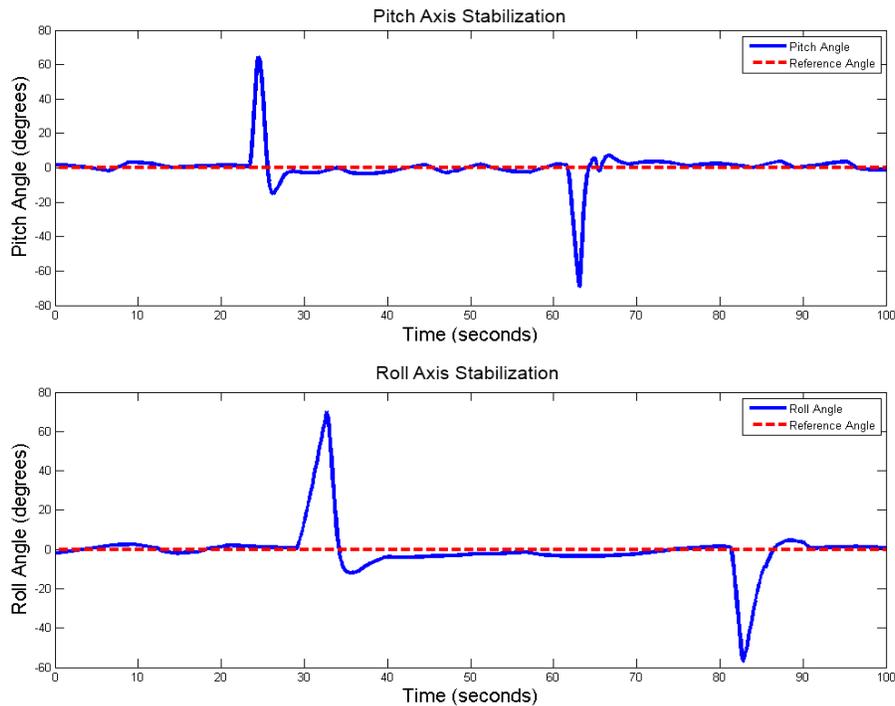


Figure 4. Pitch and Roll Axes Stabilization

shown in Fig. 4.

In the figure, positive and negative peak points represent the moment when the remote controller sticks is released and the PID controllers step in. It can be seen that the controllers successfully drive the system pitch and roll outputs to follow the reference of 0 degrees.

Future research directions include expanding the HIL system to also include the yaw axis, and testing different types of autopilot algorithms.

## Acknowledgements

We thank Turkish Scientific and Technological Research Council (TUBITAK) for funding this study under the 1507 programme (project number: 7110829). We also thank TK3 Teknik and TOBB ETÜ for providing resources which have made this work possible.

## References

- [1] Jung Soon Jang and D. Liccardo, "Automation of small UAVs using a low cost MEMS sensor and embedded computing platform," in 25th Digital Avionics Systems Conference, 2006 IEEE/AIAA, October 2006, pp. 1–9
- [2] Jodeh, N., Blue, P., Waldron, A., "Development of Small Unmanned Aerial Vehicle Research Platform: Modeling and Simulating with Flight Test Validation," AIAA Modeling and Simulation Technologies Conference, Vol. 1, AIAA, Washington, DC, 2006
- [3] Haiyang Chao; Yongcan Cao; YangQuan Chen; , "Autopilots for Small Fixed-Wing Unmanned Air Vehicles: A Survey," Mechatronics and Automation, 2007. ICMA 2007. International Conference on , vol., no., pp.3144-3149, 5-8 Aug. 2007
- [4] Hongxi Yang; Qingbo Geng; , "The design of flight control system for small UAV with static stability," Mechanic Automation and Control Engineering (MACE), 2011 Second International Conference on , vol., no., pp.799-803, 15-17 July 2011
- [5] J.M. Sullivan, "Evolution or revolution? The rise of UAVs," IEEE Technology and Society Magazine, vol. 25, no. 3, pp. 43–49, Fall 2006
- [6] Stojcsics, D.; Molnar, A.; , "Fixed-wing small-size UAV navigation methods with HIL simulation for AERObot autopilot, " Intelligent Systems and Informatics (SISY), 2011 IEEE 9th International Symposium on , vol., no., pp.241-245, 8-10 Sept. 2011
- [7] Santos, S.R.B.; Oliveira, N.M.F.; , "Test platform to pitch angle using hardware in loop," Frontiers in Education Conference, 2009. FIE '09. 39th IEEE , vol., no., pp.1-5, 18-21 Oct. 2009
- [8] Shixianjun; Song Jiakun; Liu Hongxing; , "Hardware-in-the-loop Simulation Framework Design For a UAV Embedded Control System," Control Conference, 2006. CCC 2006. Chinese , vol., no., pp.1890-1894, 7-11 Aug. 2006
- [9] Ta-ming Shih; Ho-chung Chang; , "FPGA based hardware in the loop test platform of small size UAV," Computational Intelligence in Robotics and Automation (CIRA), 2009 IEEE International Symposium on , vol., no., pp.551-556, 15-18 Dec. 2009