# Hardware-in-the-loop Test Platform Design for UAV Applications

Emre ATLAS[1,a], Melike İrem ERDOĞAN[1,b], Onur Baki ERTİN[1,2,c], Anıl GÜÇLÜ[2,d], Yusuf Eren SAYGI[1,3,e], Ünver KAYNAK[1,f], Coşku KASNAKOĞLU[1,g]

[1]TOBB University of Economics and Technology, Ankara, Turkey.

[2]ROKETSAN, Ankara, Turkey, [3]YUER Havacılık, Ankara Turkey

[a]emreatlas90@gmail.com, [b]miremerdogan@gmail.com, [c]onurertin@gmail.com, [d]anilguclu@gmail.com, [e]y.erensaygi@gmail.com, [f]unkaynak@gmail.com, [g]kasnakoglu@gmail.com

**Keywords:** HIL, UAV test platform, UAV simulation systems, Hardware-in-the-loop, Autopilot, PID, Matlab model, PID controller, UAV, UAV test, Gyroscope test systems.

**Abstract.**

A hardware-in-the-loop (HIL) platform for unmanned air vehicle (UAV) systems is designed that demonstrates flight attitudes on yaw, pitch and roll axes. The design combines a sophisticated flight simulation software with a platform capable of moving 360 degrees on all axes. This enables the testing of the flight sensors and autopilot algorithms for all sorts of scenarios including emergency and acrobatic cases where an indefinite number of full rotations in the yaw, roll and pitch might take place.

## 1. Introduction

Unmanned aerial vehicles (UAVs) make independent autonomous flights for military and civilian projects. Autonomous flight also minimizes the human factor. Hardware-in-the-loop (HIL) test systems combine software and hardware to test UAV systems under various environment conditions and also act as a pre-flight test to prevent damage from human errors. [1-5] Additionally HIL platforms can also serve as an educational tool for students to test their autonomous flight, navigation and trajectory control algorithms and tune various system parameters. [6]

This paper describes the development of a HIL platform for the purpose of testing our flight equipment and autopilot algorithms.

## 2. System Overview

Inertial measurement units (IMU) are valuable feedback sources for UAV autopilot systems. These units play the most important role in an autonomous vehicle's attitude estimation process. To include IMU into the HIL test loop as hardware, three axes (roll, pitch, yaw) custom flight motion simulator (FMS) is built.

FMS systems consist of numerous different hardware and software elements. In the present work, commercial off-the shelf (COTS) flight simulator software, communication interface software, mechanical platform with three aligned tables, DC motors and driver circuitry, encoders and microcontrollers are integrated.

Mechanical platform is constructed with durable wooden material. Three DC motors are individually connected to the separate control surfaces. Microcontroller unit (MCU) drives the motors via DC motor driver circuitry. Three encoders which are integrated with motor shafts send motor position measurements back to the MCU.

Since it provides realistic flight dynamics and over network accessible data structure, X-plane 10 is used as flight simulation software in our system. Communication interface application we developed maintains connection between X-plane and the MCU via user datagram protocol (UDP) and over universal serial bus (USB) virtual serial ports. Our application collects pitch, rolls and yaw an-

gles of the simulated vehicle from X-plane and sends these data through virtual serial port. MCU receives these inputs as system reference and our own PID algorithm placed on the MCU calculates new motor commands. Motor driver board interprets commands into the voltage and drives the motors. Three individual PID controllers are implemented for each axis and encoders provide position feedback to PID controllers. Block diagram of the system is shown in Fig. 1.

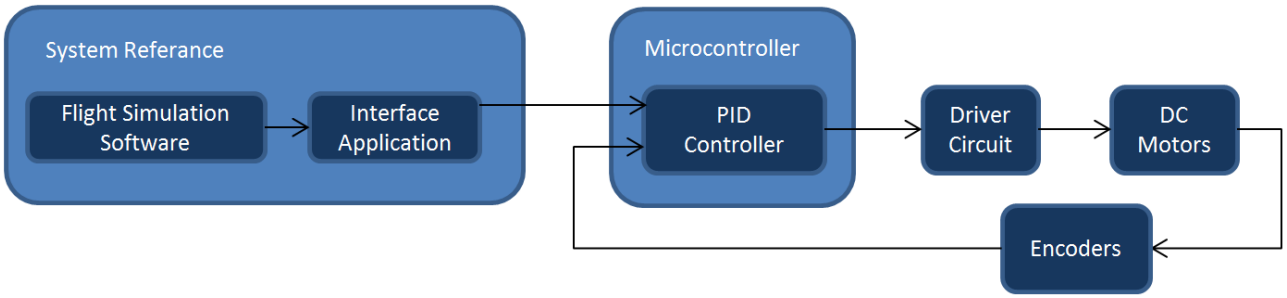



Figure 1. System Block Diagram

## 3. Flight Motion Simulator

### 3.1. Electromechanical Design

FMS (Flight Motion Simulator) is able to perform three axis movements. The placement is as follows: longitudinal axis performs in the inner plane, lateral axis performs in the middle ring and normal axis performs in the outer ring. Our design started from inner plane whose size is chosen according to the size of the autopilot boards to be tested. Then middle ring which is also inner plane holder was chosen considering inner plane and link between inner plane and middle ring. After that outer ring which is also middle ring holder was chosen like middle ring. Finally outer holder was chosen based on the entire platform. Plywood was chosen as main material for constructing prototype platform. Inner plane, middle and outer ring are made of plywood. Also outer holder is made of wood. After that torque was calculated for each angle and motors were chosen accordingly. Also motors were chosen considering real platform could be heavier than the prototype. Then the calculations were iterated according to chosen motors. Encoders are necessary for this application to provide angular position feedback. Based on our calculations, for inner plane and middle ring movement identical motors were chosen and for outer ring another motor was chosen whose specs are shown in Table 1.

| | **Longitudinal and lateral axes** | **Normal axis** |
|---|---|---|
| **Dimensions:** | 37D x 78.3L mm | 37D x 66L mm |
| **Weight of motor:** | 188,4 g | 0,22g |
| **Gear Ratio** | 131:1 | 67:1 |
| **Voltage Range:** | 6 - 15 V | 6 - 15 V |
| **Nominal Voltage(VDC):** | 12 V | 12 V |
| **Speed @ Cont. Torque, 12V:** | 75 rpm | 150 rpm |
| **No-Load Current @ 12V:** | 250 mA | 300 mA |
| **Peak Current (Stall) @ 12V:** | 6000 mA | 5000 mA |
| **Stall Torque @ 12V:** | 20 kg/cm | 14,4 kg/cm |

Table 1. Motor Specifications

While constructing platform each axis was boned from the middle. This ensures that the same point is in the center of all axes. Each axis rotates freely which causes twisting of cables so slip rings are used for solving this problem. Slip rings provide transmission of either data or power without any problem. For each axis different slip rings are used: For outer ring 24 cables x 1A slip

ring, for middle ring 12 cables x 1A and for inner ring 6 cables x 1A slip rings were used. After that all cables were plugged and named. With this cabling the construction is completed as in Fig. 2.

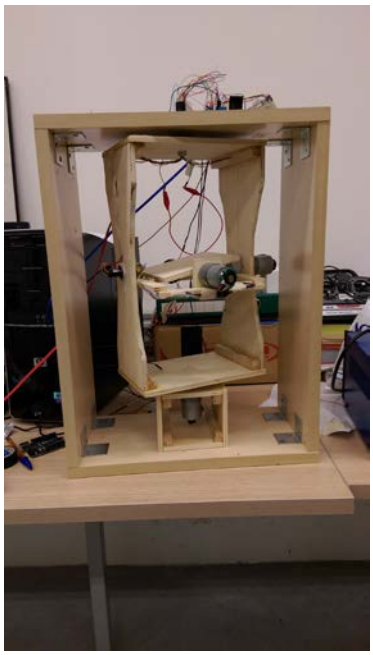


Figure 2. Photo of the HIL platform

Next step is controlling the platform. For this purpose Arduino Mega was used as a microcontroller and Pololu dual VNH5019 motor driver and L298N were used for driving the motors. Motors have six connections: +, -, gnd, vcc, hall sensor A and B for encoders. + and – pins are connected to motor drivers which let us change direction of rotation using high/low values from microcontroller, Hall sensor A and B pins 4 are connected to microcontroller's interrupt pins and gnd and vcc are connected to gnd and 5V pins. Read encoder values give how many steps the motor stepped and multiplying this value with CPR value gives angle value as a result. Calculated angle values for each axis are used by each axis' PID controller. Arduino's communication with the computer on which the fligth simulator operates is provided by RS232 protocol with USB with data transfer speed of 115200 bps. Computer sends data to serial port in 16 byte by 16 byte format. Arduino reads these 16 bytes then processes this data to obtain roll, pitch and yaw values. These values are the setpoints of each axis. After obtaining input and setpoint, Arduino calculates the output of the PID controller; this value is between -255 to 255 where minus sign represents counter clockwise direction. Arduino changes digital pins values High(5V)/Low(gnd) to Low(gnd)/High(5V) according to this output's sign and its absolute value is sent as PWM signal between 0 to 255 from Arduino's pwm pins to motor drivers input pins. Motor drivers have connection between power supply's 12V and ground and gnd, vcc, 3 input which the 2 of them are received from Arduino's digital pins are used for determining direction of rotation and other one received from Arduino's PWM pin is used for determining speed. Speed is determined by applied voltage changing between 0 to 12V based on received signal from Arduino's PWM pin. After this the final step is controller design for the motors.

### 3.2. Software Design

#### 3.2.1. Controller Design

During the tests at each time step the plane's Euler angle values are sent to the FMS. For this reason it is necessary to model the FMS mathematically and design a controller for each axis. Modeling is started after the FMS is fully integrated physically. As stated before, FMS is composed of three axes which are yaw, pitch, and roll. Modeling, parameter estimation, and controller design steps are done for each axis individually. At first a PID controller is set and applied to an axis. At each time step, motor position is logged. It is assumed that the DC motor has a second order transfer function as;

$$\frac{\dot{\theta}}{V} = \frac{K}{JLs + (JR + BL)s + BR + K^2}$$

where $\dot{\theta}$: Angular velocity, rad/sec, $V$: Applied voltage, Volts, $K$: Motor torque constant, Nm/Amp, $J$: Moment of inertia of the rotor, kgm$^2$, $L$: Electric inductance, H, $R$: Electric resistance, Ohm, and $B$: Motor viscous friction constant, Nms.

At the first motor, which actuates yaw axis, a PID type controller is built in MATLAB/Simulink as in Fig. 3.

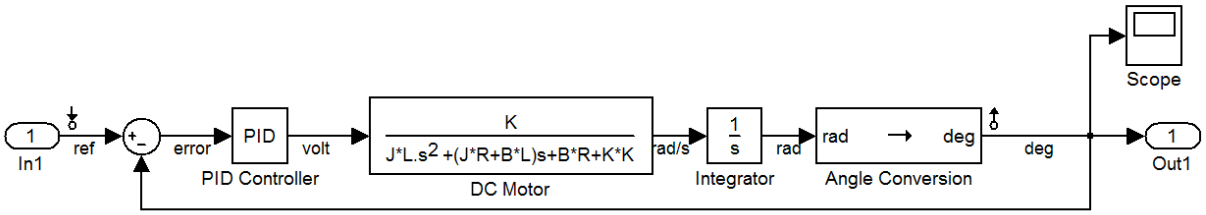



Figure 3. Simulink Model

As stated before PID controller parameters are set with empirical methods. Some reference commands are sent to the motor and its rotational position information, which is taken from encoder, is logged. Then, by means of Parameter Estimation Toolbox of the MATLAB software, K, J, L, R, B parameters are estimated. Initial values are assigned to the parameters, which will be estimated. In the model (Fig. 3), reference commands and encoder data are defined at In1 and Out1 port, respectively. PID controller gains are fixed as they are. To estimate the parameters which are placed in the DC motor block, Pattern Search optimization method and Latin Hypercube search method are used. After the parameters are estimated, PID gains are estimated with using Control and Estimation Toolbox. Above steps are done for yaw, pitch, and roll axes separately. After the estimation, motor and PID controller gains are found as Table 2.

| | Parameters | Values | Controller Gains | |
|---|---|---|---|---|
| Yaw Axis | K | 1.6839 | P | 1.226 |
| | J | 0.1044 | I | 0.0172 |
| | R | 1.0462 | D | 0.12 |
| | L | 2.2295e-8 | | |
| | B | 1.0918 | | |
| Pitch Axis | K | 2.2800 | P | 1 |
| | J | 2.1244e-4 | I | 0.008 |
| | R | 1.2017 | D | 0.035 |
| | L | 4.0099e-4 | | |
| | B | 1.6677 | | |
| Roll Axis | K | 2.7598 | P | 1.226 |
| | J | 0.0291 | I | 0.0172 |
| | R | 0.0641 | D | 0.0702 |
| | L | 0.3589 | | |
| | B | 1.6472 | | |

Table 2. Controller and Plant Parameters

### 3.2.2. Flight Simulation Software Integration

Xplane 10 allows user to work with out-of-the-box vehicle models; this feature lets us skip vehicle modeling process and gives us the flexibility to change the plane model at any time. Low costs, high payloads and durability makes remote controlled (RC) model planes perfect platforms to build up UAV systems. A very common trainer type RC model plane PT-60 is selected as our Xplane plane model. PT-60 is a four channel plane thus it is possible to control aileron, elevator, rudder and throttle channels. According to these channel inputs and environmental effects (if enabled) the simulation software calculates the vehicle's attitude. Xplane simulation window and PT-60 plane can be seen in Fig. 4.

Figure 4. Xplane Simulation Window and PT-60 plane

Xplane lets user review and transfer flight data to any other UDP enabled application. Xplane uses 41 byte-length data arrays to receive and send data. First 9 bytes contain dataset number and data type, rest 32 bytes can carry up to 8 variables, each of 4 bytes in IEEE 754 single precision binary floating point format. Xplane has to be configured to send dataset number 17 which contains roll, pitch and yaw angles of the vehicle [1].

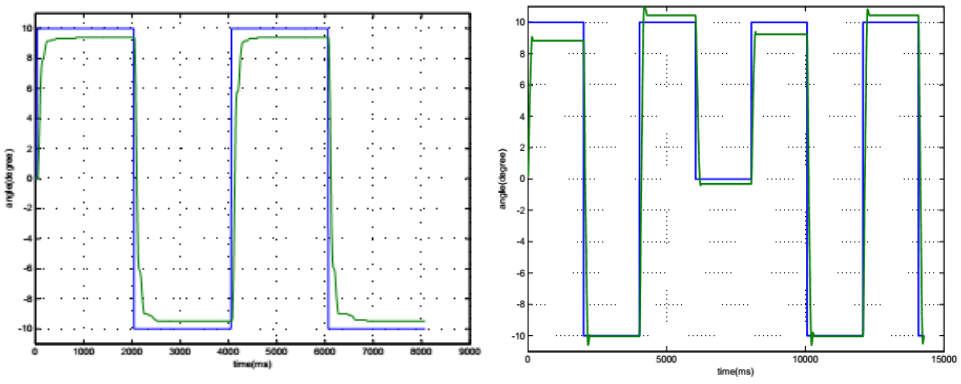
Communication interface application listens to Xplane's UDP send data port, captures datasets, distinguishes headers from data and sends angle bytes to the MCU over serial port. MCU turns byte data into meaningful integers and uses these values as the axes' reference angles.

## 4. Results, Conclusions and Future Works

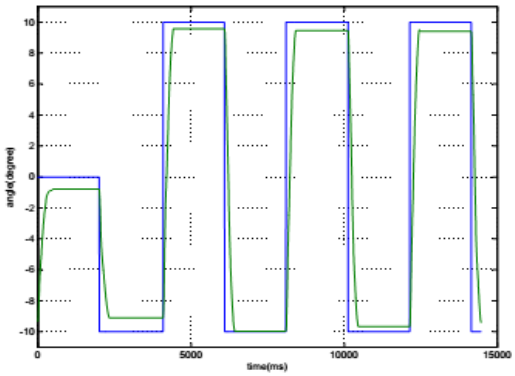
Using the HIL platform we can find the most efficient control algorithms and route planning under various environmental effects. HIL platform co-operates software packages including X-plane for the flight visualization, Arduino microcontroller and C# code on the computer.

Matlab/Simulink is also used for the dynamic flight model, and control algorithms. HIL platform , management computer and radio control are also connected to each other through Arduino, motor driver and USB cable. HIL platform has been successfully used to test the software systems in real time with realistic visual feedback of UAV.

HIL simulation platform has been successful in reducing the number of flight tests and enabling finding software faults or the best flight control algorithms before flights. All measurements are realistic results taken from real sensors of the real-time controller placed on the platfrom. The platform's behaviors under different inputs are shown in Fig. 5.




Roll Angle Pitch Angle



Yaw Angle

Figure 5. Platform Responses for roll, pitch and yaw axes

In the figure every two seconds the set point changes from 10 to -10 degrees and is sent to platform; it can be seen that platform successfully follows the set point with very small error (about 0.5 degrees) under desired circumstances.

Future research directions include building a more reliable platform construceted from another material and the testing of various autopilot algorithms on the platform.

## 5. Acknowledgements

The authors would like to thank the Scientific and Technological Research Council of Turkey (TÜBİTAK) for supporting this work under project number 113E581.

## 6. References

[1] Onur Baki Ertin, Halim Korkmaz, Ünver Kaynak ve Coşku Kasnakoğlu. Hardware-in-the-Loop Test Platform for a Small Fixed Wing Unmanned Aerial Vehicle Embedded Controller. The 32nd IASTED International Conference on Modelling, Identification and Control, Innsbruck, Austria, 2013.

[2] McManus, Iain A., Greer, Duncan G., & Walker, Rodney A. (2003) UAV Avionics "Hardware in the Loop" Simulator. In 10th Australian International Aerospace Congress, 29 July – 1 August 2003, Brisbane, Queensland, Australia.

[3] Dongwon Jung, Panagiotis Tsiotras, Modeling and Hardware-in-the-Loop Simulation for a Small Unmanned Aerial Vehicle, AIAA Conference, 2007

[4] Guowei Cai , Ben M. Chen, , Tong H. Lee , Miaobo Dong, Design and implementation of a hardware-in-the-loop simulation system for small-scale UAV helicopters,

[5] McManus, Iain A., Greer, Duncan G., & Walker, Rodney A. (2003) UAV Avionics "Hardware in the Loop" Simulator. In 10th Australian International Aerospace Congress, 29 July – 1 August 2003, Brisbane, Queensland, Australia

[6] Claire Tomlin, Stanford Univ., CA; Jung Jang, AIAA, Autopilot design for the Stanford DragonFly UAV - Validation through hardware-in-the-loop simulation (2001)