

Article

Not peer-reviewed version

---

# A Chip-Level Testing Platform of Unmanned Vehicle Autopilot Systems with FPGA-Based Hardware-in-the-Loop Simulation

---

[Xunhua Dai](#) and [Ruyi Fang](#) \*

Posted Date: 4 March 2024

doi: 10.20944/preprints202403.0132.v1

Keywords: unmanned vehicles; autopilot systems; chip-level faults; hardware-in-the-loop (HIL) simulation; Field Programmable Gate Array (FPGA)





Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Article

# A Chip-Level Testing Platform of Unmanned Vehicle Autopilot Systems with FPGA-Based Hardware-in-the-Loop Simulation

Xunhua Dai <sup>1</sup>  and Ruyi Fang <sup>2,\*</sup> <sup>1</sup> School of Computer Science and Engineering, Central South University, Changsha 410083, China<sup>2</sup> School of Electronic information, Central South University, Changsha 410083, China

\* Correspondence: 214712287@csu.edu.cn;

**Abstract:** Nowadays, unmanned vehicles are widely used in various fields. However, the safety of unmanned vehicles is directly determined by the autopilot systems, and sensor faults are a very critical factor that affects the normal operation of the autopilot systems. To efficiently and comprehensively test unmanned systems for sensor faults, this paper proposes a chip-level testing platform for unmanned vehicle autopilot systems based on FPGA hardware-in-the-loop simulation. Unlike existing testing platforms, the introduction of FPGA technology allows us to achieve nanosecond real-time simulation frequency, thus realizing chip-level simulation of sensors and black-box automated testing. Firstly, we propose a set of general platform and modeling methods, and the paper presents a detailed application of the four most common necessary sensors in autopilot systems as examples. Secondly, we propose for the first time a chip-level fault testing method for unmanned systems, and build a test platform with a quadrotor vehicle, the PX4 software platform, and the Pixhawk4 hardware platform as examples. Finally, the experimental results verify that the testing platform proposed in this paper can adapt to various autopilot systems and has high simulation test reliability.

**Keywords:** unmanned vehicles; autopilot systems; chip-level faults; hardware-in-the-loop (HIL) simulation; Field Programmable Gate Array (FPGA)

## 1. Introduction

As unmanned vehicles such as quadcopters [1,2], fixed-wing [3], unmanned ground vehicles [4], and unmanned surface ships [5] are widely used in various fields, the electromechanical equipment of these unmanned vehicles is becoming increasingly complex, and the tasks they undertake are becoming more and more diverse. However, intensive flight missions may induce fault that can lead to conflicts, collisions, and crashes. Therefore, for the expected growth of unmanned vehicle equipment, people are paying even greater attention to the reliability of autopilot system. As a life-critical system for unmanned vehicles, the reliability evaluation [6] of the autopilot system is a significant research content for airworthiness testing.

In recent years, the hot research content such as sensor fault diagnosis(FD) [7,8], the fault tolerant control(FTC) [9,10] and prognostics and health management(PHM) [11,12] technology in the field of unmanned vehicles shows that the unmanned vehicle autopilot system is suffering from various types of faults, which revealing that the unmanned vehicle autopilot system still has considerable vulnerability. In order to efficiently verify the reliability of the autopilot system, a testing platform that can accomplish the fault simulation of the autopilot system has a very important role for the above research content. At present, the reliability assessment of most unmanned vehicle autopilot systems relies on the real-aircraft flight tests that inject a variety of faults during flight. Although the real-aircraft experimental method can genuinely reflect the actual situation, faults can only occur under specific conditions, and fatal faults will cause severe damage to unmanned vehicles. In addition, it is difficult to ensure the consistency of the faults under real-aircraft testing (to ensure that the same faults have the same fault action time and the same fault impact degree in multiple tests).

Hardware-in-the-loop(HIL) simulation [13], as a very popular simulation method in recent years, can make up for the problems of randomness, consistency and repeatability that are difficult to be

solved by real machine testing methods. The publicly released UAV simulation platforms can be categorized into two types: turntable platform and generic platform. As a typical representative of the rotary HIL simulation platform, the asymptotic tracking controller test platform proposed by Zhao [14] et al. has outstanding advantages when attitude control and external factor interference are considered, but it is difficult to inject rich sensor fault types due to the use of actual unmanned vehicles. Generic platforms proposed by large tech companies or research teams, such as XTDrone [15], Gazebo [16], Jmavsim [17], Airsim [18], and FlightGoggles [19] are more concerned with the validation of control law algorithms and intelligent algorithms, and very few are able to simulate various types of faults and complete the reliability assessment of unmanned vehicle autopilot systems. A few other researchers, Gong [20] et al. proposed a semi-physical simulation platform for UAV autopilot system to generate various fault data, but the platform pay more attention to the characteristics of the fault data realizations without considering the impact of the fault behavior on the whole system in depth. Khan [21] et al. proposed a fault injection framework to simulate UAV flight and evaluated the reliability of the flight autopilot system undergo multiple GPS faults but did not consider other sensors in the hardware system. In addition, the above HIL platforms, besides failing to meet the strong demand for fault generation, also generally suffer from difficulties in adapting to different unmanned vehicles, compatibility problems between the software or hardware platforms of the autopilot system, and difficulties in meeting the low real-time performance brought about by the bandwidth of data transmission.

Field Programmable Gate Array(FPGA) [22] are a special type of digital circuitry that is unique in that it can be programmed at the hardware level to perform specific logic functions. Compared to Application Specific Integrated Circuit(ASIC) in sensor chips, FPGA offer significant flexibility, parallel processing, and operating frequency advantages that make them a powerful tool in the field of chip-level design and verification [23]. Thus, in this paper we simulate in FPGA the chip-level behavior of components of various sensor products of an autopilot system, including sensor outputs, and these simulated signals are subsequently used in flight control algorithms to ensure stability and correct response of the vehicle. Further, various flight conditions and sensor failure scenarios can be easily simulated with FPGA, which are very helpful in verifying the performance and robustness of sensor chips and autopilot systems under extreme conditions.

In this article, we propose a chip-level testing platform for reliability evaluation of unmanned vehicle autopilot systems, focusing on solving the generalized adaptability of the simulation platform as well as generating highly reliable sensor faults. The main research content and corresponding contributions are as follows:

- (1) This paper proposes a multi-sensor digital modeling method for adapting to various UAV autopilot software platforms, which is divided into two parts: measurement model and communication modeling, and is verified on the FPGA platform. The simulation sensor model outputs the same digital signal as the real sensor product, realizing the interaction of sensing data between the simulation model and the autonomous driving hardware platform.
- (2) This article proposes a coarse-grained failure model of inertial measurement unit(IMU), barometer, magnetic compass sensor and global position system(GPS) receiver under the action of multiple common failure factors for the necessary sensor chips of the autonomous driving hardware platform. Used to provide rich fault data for UAV health management research work.
- (3) This article proposes a HIL platform for reliability verification of various autopilot systems, adapting to a variety of popular autonomous driving software platforms and hardware platforms as well as various types of UAV systems, including fixed-wing, multi-purpose Rotors, unmanned ships and unmanned vehicle systems. The platform has many excellent features such as high fidelity, ultra-high real-time performance, and black box testing.

## 2. Proposed HIL System Structure

In the study [24], we introduced a standardized framework for UAV autopilot systems based on FPGA HIL simulation. Building upon this foundation, we propose a testing platform that dedicated to fault generation for autopilot sensor system. As shown in Figure 1, the platform is divided into four parts: Real-time HIL Platform(RHILT), Unmanned Vehicle Autopilot System(UVAS), Host Computer(HC) and Radio Control(RC).

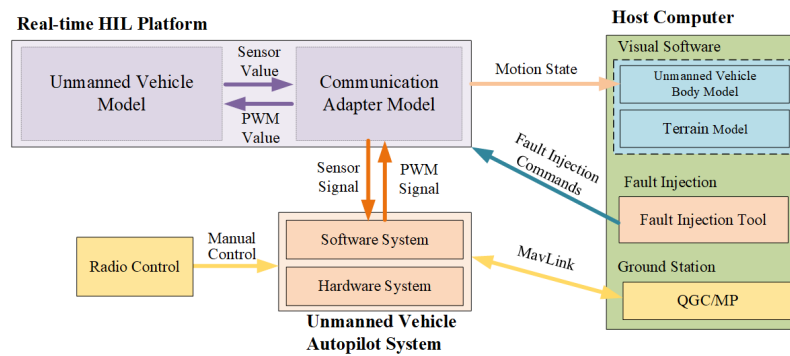


Figure 1. Structure of Hardware-in-the-loop simulation platform.

### 2.1. RHILT

It is primarily divided into two parts: system-on-chip(SoC) and FPGA. The SoC executes a mathematically complex unmanned vehicle model and employs the RT-Linux system to ensure synchronization of the model's clock with the real-world clock. In contrast, the FPGA operates a high-frequency update communication adapter model to ensure the real-time nature of sensor communication. See Section 3 for the details of the unmanned system model involved in this module. As well, the sensor model is discussed in detail in Section 4.

Moreover, the PS sends flight simulation data packets at a higher frequency, including byte streams of the *airframe type, location, attitude*, etc. Concurrently, the PS possesses a fault injection interface that supports online modification of model parameters for the purpose of injecting faults.

### 2.2. UVAS

An UVAS consists of a control hardware system and a control software system, and the most popular open source platforms for the consumer market are listed in [25]. In general, the hardware system is mainly used to provide the computational environment (embedded processor) and sense motion information (sensors), while the software system is mainly used to solve the control laws (Extended Kalman Filter algorithm) to achieve autonomous navigation control. For safety reasons, the software system will also include algorithms for fault-tolerant control, fault diagnosis and health management.

In order to ensure that the autopilot system can work properly on the proposed testbed, it is necessary to block (or remove) the on-board sensors in the hardware system and reconnect the autopilot system to the pins of the sensor chip simulated by the fpga system of the real-time simulation computer in order to receive the digital communication signals from the sensors. The above solution does not require any modification to the software system, which greatly protects the controller software algorithm from being leaked, and is of practical significance for black-box testing of different autopilot products from different UAV companies.

### 2.3. HC

The mainframe is tasked with running the application. The following three applications are included.

- (1) The vision software uses the latest unreal engine 5 to render flight simulation animations, and the simulation model of the UAV frame and terrain environment (including obstacles) can simulate the objective world with a high degree of realism. It receives flight simulation data in real time to determine the UAV attitude and position and other dynamic performance characteristics. At the same time, the fault injection effect can be qualitatively assessed by observing the UAV flight process.
- (2) The fault injection tool is used to trigger fault generation online. In this test method, engineers add fault test cases by setting *fault modes*, *fault durations* and *fault intensities* to simulate various fault cases that may occur in the sensor system.
- (3) On the one hand, the ground station formulates automatic test tasks (such as cruising, level flight, or hovering), and on the other hand, it monitors sensor data in real time to quantitatively evaluate the fault injection effect.

#### 2.4. RC

As an optional tool, the RC has the same mission planning functions as the ground station. Engineers can manually control the movement of the unmanned vehicle to the desired state and complete some difficult flight actions (such as yaw, roll extreme sports, etc.).

### 3. Unmanned Vehicle Model

The unmanned vehicle model mainly describes the dynamic and kinematic motion process of the unmanned vehicle in the working environment. A complex unmanned vehicle model can be regarded as composed of several small subsystems, as shown in Figure 2, including actuator subsystem, environment subsystem, force and moment subsystem, vehicle body subsystem and Sensor Hub.

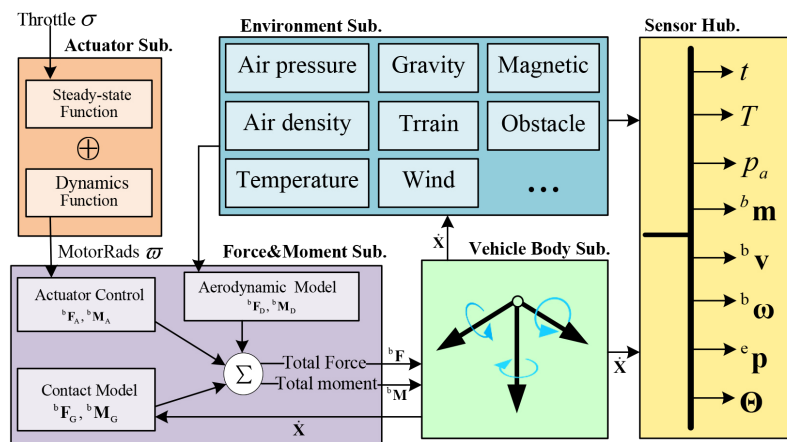


Figure 2. Unmanned vehicle model.

The actuator subsystem is the primary power source of control unmanned vehicles, but the accurate power model is extremely complex. From the control efficiency and the general characteristics of the motor system, the transfer function of the simplified model of the power unit is given as follows

$$\omega_i = \frac{1}{T_m s + 1} (C_R \sigma_i + \omega_b) \quad (1)$$

where  $i$  is the sequence number of the actuator,  $\omega_i$  is the motor speed,  $\sigma_i$  is the throttle command that is equal to the PWM value,  $T_m$  is the motor dynamic response time constant,  $C_R$  and  $\omega_b$  are constant



parameters that can be obtained from the website<sup>1</sup>. Then, the driving force  ${}^bF_A$  and driving torque  ${}^bM_A$  generated by the actuator are

$$\{{}^bF_A, {}^bM_A\} = f(E, \sum_i \omega_i^2) \quad (2)$$

where  $E$  is the control efficiency matrix taking into account the different rack layouts.

The support forces  ${}^bM_G$  and support moments  ${}^bF_G$  from the ground during take-off and landing, as well as the aerodynamic resistance  ${}^bF_D$  and damping moments  ${}^bM_D$  during flight, are also considered in depth. The total force  ${}^bF$  and moment  ${}^bM$  of the unmanned carrier is expressed as

$$\begin{aligned} {}^bF &= {}^bF_A + {}^bF_G + {}^bF_D \\ {}^bM &= {}^bM_A + {}^bM_G + {}^bM_D \end{aligned} \quad (3)$$

The unmanned vehicle provides the necessary force and moment to control the position and attitude of the system through its actuators. A six-degree-of-freedom(6DOF) model can describe the motion of any unmanned vehicle. Define unmanned vehicle flat Earth position as  ${}^e p \triangleq [x \ y \ z]^T$ , flight speed vectors in body-fixed axes concerning body frame as  ${}^b v$ , likewise in the flat Earth reference frame as  ${}^e v$ , angular rates vectors in body-fixed axes as  ${}^b \omega \triangleq [\omega_{x_b} \ \omega_{y_b} \ \omega_{z_b}]^T$ , body euler angle as  $\Theta \triangleq [\phi \ \theta \ \psi]^T$ . The dynamic and kinematic characteristics of the unmanned vehicle can be described as follows

$$\dot{X} = F(t, X, \sigma) \quad (4)$$

where  $t$  is real time factor,  $X \triangleq [{}^e p \ {}^e v \ {}^b \omega \ \Theta]^T$ ,  $\dot{X}$  is the next state estimate,  $\sigma$  is the PWMs vector from the output of the autopilot system.

Inspired by the works presented in [26], the rigid motion law of the unmanned vehicle can be expressed by differential equations, including force and moment equations, navigation equations, and motion equations.

$$\begin{cases} {}^e \dot{p} = {}^e v = R_b^e \cdot {}^b v \\ {}^b \dot{v} = -[{}^b \omega]_{\times} \cdot {}^b v + {}^b F/m \\ \dot{\Theta} = W \cdot {}^b \omega \\ J \cdot {}^b \dot{\omega} = -{}^b \omega \times (J \cdot {}^b \omega) + {}^b M \end{cases} \quad (5)$$

where  $R_b^e$  is the rotation matrix from the body-fixed frame to the earth inertia frame,  $[{}^b \omega]_{\times}$  is  ${}^b \omega$  skew symmetric form,  $W$  is a matrix shorthand,  $J$  is the moment of inertia,  ${}^b F$  and  ${}^b M$  are the vectors of external forces and moment, respectively, satisfying the following:

$$R_b^e = \begin{bmatrix} C_\theta C_\psi & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ C_\theta S_\psi & S_\psi S_\theta \sin \phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix},$$

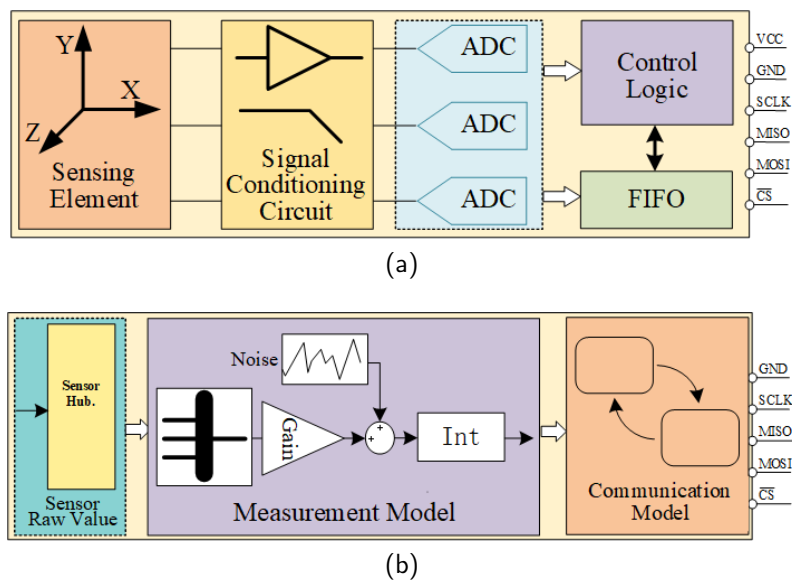
$$W \triangleq \begin{bmatrix} 1 & T_\theta S_\phi & T_\theta C_\phi \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi / S_\theta & C_\phi / C_\theta \end{bmatrix}, [{}^b \omega]_{\times} \triangleq \begin{bmatrix} 0 & -\omega_{z_b} & \omega_{y_b} \\ \omega_{z_b} & 0 & -\omega_{x_b} \\ -\omega_{y_b} & \omega_{x_b} & 0 \end{bmatrix}$$

<sup>1</sup> URL: <https://www.flyeval.com/index.html>

where  $S(\cdot)$ ,  $C(\cdot)$  and  $T(\cdot)$  denote  $\sin(\cdot)$ ,  $\cos(\cdot)$  and  $\tan(\cdot)$ , respectively.

#### 4. Sensor Digital model

In consumer electronics, manufacturers favor high-performance MEMS sensors. For example, IMU sensors use the piezoelectric or Coriolis effect to detect changes in motion, and the workflow as shown in Figure 3a includes signal amplification, digitization, and data communication control logic. In order to simulate actual flight testing of an autopilot system on the HIL platform, a sensor simulation workflow as shown in Figure 3b is recommended. This simulation generates raw sensor data, adjusts the data to meet the requirements of the autopilot software through scaling and conversion, and simulates the data transfer using appropriate communication protocols.



**Figure 3.** Sensor workflow in the real-world and the HIL platform. (a) Sensor workflow of autopilot system in the real-world. (b) Sensor simulation workflow in the HIL platform.

##### 4.1. Sensor Data Source

The types of sensor data are diverse. The first type of sensor data is to perceive its own attitude, such as acceleration and angular velocity, which can be calculated by the 6DOF model; The second type of sensor data is related to working environment factors, such as temperature, air pressure and magnetic field strength, these require the use of standard modules with high reliability; The third type of sensor data is observation, such as the longitude, latitude and altitude data required for GPS positioning, which need to be converted between different reference systems. The program of calculating the sensor raw data is as shown in Algorithm 1. It is worth mentioning that the Flat Earth to LLA Model, WGS84 Model, ISA Atmosphere Model and World Magnetic Model from lines 5 to 8 in Algorithm 1 are standard library models [27].

**Algorithm 1:** Calculate sensor data**Input:** The throttle vector  $\sigma$ .**Output:** The acceleration  ${}^b\dot{v}$ , angular velocity  ${}^b\omega$ , temperature  $T$ , magnetic field strength  ${}^b\mathbf{m}$ , air pressure  $p_a$ , longitude  $\mu$ , latitude  $\iota$  and height  $h$ .1 Initialization: Initial temperature of the environment  $T$ , the GPS location  $\mu_0, \iota_0$  of the birth point of the unmanned vehicle system.

2 This function is executed every 5ms.

3 **Function** Calculate sensor data:4      $[{}^b\dot{v}, {}^b\omega, {}^e\mathbf{p}, \Theta] \leftarrow \text{Unmanned Vehicle Model}(\sigma) \text{ via (1)-(4)}$ 5      $[\mu, \iota, h] \leftarrow \text{Flat Earth to LLA Model}({}^e\mathbf{p}, [\mu_0, \iota_0])$ 6      $g \leftarrow \text{WGS84 Model}([\mu, \iota, h])$ 7      $p_a \leftarrow \text{ISA Atmosphere Model}(h)$ 8      ${}^b\mathbf{m} \leftarrow \text{World Magnetic Model}([\mu, \iota, h, t])$ 

## 4.2. Measurement Model

## 4.2.1. IMU Model

The output of the MEMS IMU are affected by many error sources, such as scale factor errors, biases, and random drift. In the field of inertial navigation, the acceleration and angular velocity measurements of the IMU [28] can be expressed as:

$$\mathbf{u}_a = \mathbf{C}_a \cdot {}^b\mathbf{a} + \mathbf{b}_a + \boldsymbol{\varepsilon}_a \quad (6)$$

$$\mathbf{u}_g = \mathbf{C}_g \cdot {}^b\boldsymbol{\omega} + \mathbf{b}_g + \boldsymbol{\varepsilon}_g \quad (7)$$

where  ${}^b\mathbf{a}$  and  ${}^b\boldsymbol{\omega}$  are the real values of the acceleration and angular velocity in the body frame, respectively. Furthermore,  $\mathbf{C}_a$  and  $\mathbf{C}_g$  are defined as follows:

$$\mathbf{C}_a = \begin{bmatrix} s_x^a & \mu_{xz}^a & -\mu_{xy}^a \\ -\mu_{yz}^a & s_y^a & \mu_{yx}^a \\ \mu_{zy}^a & -\mu_{zx}^a & s_z^a \end{bmatrix}, \mathbf{C}_g = \begin{bmatrix} s_x^g & \mu_{xz}^g & -\mu_{xy}^g \\ -\mu_{yz}^g & s_y^g & \mu_{yx}^g \\ \mu_{zy}^g & -\mu_{zx}^g & s_z^g \end{bmatrix}$$

where the diagonal elements  $S_a \triangleq [s_x^a \ s_y^a \ s_z^a]$  and  $S_g \triangleq [s_x^g \ s_y^g \ s_z^g]$  of the  $\mathbf{C}_a$  and  $\mathbf{C}_g$  matrix are the acceleration and angular velocity scale factor errors respectively.  $\mu_{xz}^a, \mu_{xy}^a, \mu_{yz}^a, \mu_{zx}^a, \mu_{yx}^a, \mu_{xy}^g, \mu_{xz}^g, \mu_{xy}^g, \mu_{yz}^g, \mu_{zx}^g, \mu_{yx}^g$  are the six installation error angles between the accelerometer frame  $a$  and the body frame  $b$ ;  $\mu_{xz}^g, \mu_{xy}^g, \mu_{yz}^g, \mu_{zx}^g, \mu_{yx}^g, \mu_{xy}^g$  are the six installation error angles between the gyro frame  $g$  and the body frame  $b$ ;  $\mathbf{b}_a$  and  $\mathbf{b}_g$  denotes the acceleration zero bias and angular velocity zero bias respectively;  $\boldsymbol{\varepsilon}_a$  and  $\boldsymbol{\varepsilon}_g$  denotes the acceleration rate noise and angular velocity rate noise respectively, assumed to be Gaussian white noise [29] with characteristics

$$\begin{aligned} E[\boldsymbol{\varepsilon}_a] &= \mathbf{0}_{3 \times 1} \\ E[\boldsymbol{\varepsilon}_a^2] &= \sigma_{a,noise}^2 \cdot f_s \cdot \mathbf{I}_{3 \times 3} \end{aligned} \quad (8)$$

where  $\sigma_{a,noise}$  is the white noise variance coefficient,  $f_s$  is the sampling rate. The zero bias  $\mathbf{b}_a$  and  $\mathbf{b}_g$  will change with time and can be established as a first-order Markov process

$$\dot{\mathbf{b}}_a = -\frac{1}{\tau_a} \mathbf{b}_a + \mathbf{n}_{b_a} \quad (9)$$



with characteristics

$$\begin{aligned} E(\mathbf{b}_a) &= \mathbf{0}_{3 \times 1} \\ E(\mathbf{b}_a^2) &= \sigma_{a,bias}^2 \cdot f_s \cdot \mathbf{I}_{3 \times 3} \\ \mathbf{n}_{b_a} &= \sqrt{\frac{2f_s \cdot \sigma_{a,bias}^2}{\tau_a}} \boldsymbol{\vartheta}_{a,bias} \cdot \mathbf{I}_{3 \times 1} \end{aligned} \quad (10)$$

where the noise driving the bias  $\boldsymbol{\vartheta}_{a,bias} \sim N(0,1)$ , time constant  $\tau_a$ , bias variation  $\sigma_{a,bias}$  can be determined by Allan variance and autocorrelation analysis.

Considering the lever-arm effect due to the fact that the accelerometer is not at the center of gravity, the accelerometer true vector is

$${}^b\mathbf{a} = {}^b\dot{\mathbf{v}} + {}^b\boldsymbol{\omega} \times ({}^b\boldsymbol{\omega} \times \mathbf{d}) + {}^b\dot{\boldsymbol{\omega}} \times \mathbf{d} - \mathbf{g} \quad (11)$$

where,  $\mathbf{d}$  represents the distance between the IMU installation position and the center of gravity position. Beside,  $\mathbf{u}_g = {}^b\boldsymbol{\omega}$ .

The dynamic characteristics of a tri-axis sensor are generally expressed by a second-order programming transfer function[30] as

$$G_i(s) = \frac{w_{i,n}^2}{s^2 + 2\xi_i w_{i,n}s + w_{i,n}^2} \cdot e^{-\tau s} \quad (12)$$

where,  $w_n$  denotes the natural frequency,  $\xi$  denotes the damping ratio, and  $\tau$  denotes the time constant.

#### 4.2.2. Compass Model

Because they are both three-axis sensors, magnetic compasses have similar measurement models and dynamic transfer functions to accelerometers or magnetometers, which are expressed as

$$\mathbf{u}_m = \mathbf{C}_m \cdot {}^b\mathbf{m} + \mathbf{b}_m + \boldsymbol{\varepsilon}_m \quad (13)$$

where  $\mathbf{C}_m$  denotes scale factors and misalignment error,  $\mathbf{b}_m$  is the zero bias,  $\boldsymbol{\varepsilon}_m$  is the rate noise, which have the same statistical characteristics as (8) and (10). And it has the same dynamic characteristics as (12).

#### 4.2.3. Barometer Model

The barometer is a typical uniaxial sensor which measures physical quantities without multi-channel characteristics, and its measurement model is expressed as follows

$$u_p = p_a + b_p + \varepsilon_p \quad (14)$$

where  $u_p$  is the pressure altitude measurement value,  $p_a$  is the true air pressure height,  $\varepsilon_p$  is the barometer drift error which has the same Gaussian white noise characteristics as (8), and  $b_p$  is the barometer wind disturbance error [31] which are expressed as

$$b_p = \frac{T_0}{\beta} \left[ \left( \frac{P'_s}{P_0} \right)^{-\frac{\beta R}{g}} - \left( \frac{P'_s - \frac{1}{2}\rho v^2}{P_0} \right)^{-\frac{\beta R}{g}} \right] \quad (15)$$

where  $T_0 = 288.15K$  (the capital letter  $K$  is the temperature unit of Kelvin) is the standard sea level temperature,  $\beta = -6.5K/km$  is the temperature lapse rate constant,  $P_0 = 1013.25hPa$  is the standard sea level pressure,  $g$  is the gravity acceleration constant,  $R = 287.05287m^2/K \cdot s^2$  is the air constant,  $\rho = 1.23kg/m^3$  is the standard air density,  $P_s$  is the measured air pressure value after wind interference,  $v$  is the wind speed.

#### 4.2.4. GPS Receiver Model

In contrast to the aforementioned three types of MEMS measurement sensors, GPS receivers are classified as calculation sensors which receive multiple satellite signals and then calculate the position of the node in the earth Latitude-Longitude-Altitude (LLA) global coordinate.

Define the node position measurement with respect to the Earth fixed coordinate system as  $\mathbf{u}_{GPS} \triangleq [\mu \ \iota \ h^T]$ . The measurement model for calculating the position of a node by the GPS receiver [32] is expressed as

$$\mathbf{u}_{GPS} = f_{LLA}({}^e\mathbf{p}) + \boldsymbol{\varepsilon}_{GPS} \quad (16)$$

where  $f_{LLA}(\cdot)$  denotes the transformation from Flat Earth to LLA,  $\boldsymbol{\varepsilon}_{GPS}$  is the measurement noise induced by the communication channel with the satellite, which can also be described by the same Gaussian white noise model as (8).

Define the node velocity measurement with respect to the Earth fixed coordinate system as  ${}^e\mathbf{v}_{GPS} = [{}^e\mathbf{v}_N^{GPS} \ {}^e\mathbf{v}_E^{GPS} \ {}^e\mathbf{v}_D^{GPS}]^T$ . The measurement model for calculating the velocity of a node by the GPS receiver is expressed as

$${}^e\mathbf{v}_{GPS} = \mathbf{R}_b^e \cdot {}^b\mathbf{v} + \boldsymbol{\varepsilon}_v \quad (17)$$

where  $\boldsymbol{\varepsilon}_v$  is the measurement noise of the velocity caused by the flying perturbation.

#### 4.3. Communication Adapter Model

In the autopilot system, critical sensors perform efficient data transfer with the processor via board-level communication buses, which involve communication interface protocols such as serial peripheral interface (SPI), inter-integrated circuit (IIC), and universal asynchronous transceiver (UART). In the proposed HIL platform, the communication adapter model simulates the digital control logic unit of the actual sensor chip to enable the conversion of sensor data into binary digital signals suitable for board-level transmission.

Here, we take the sensor products in Pixhawk4 as an example to describe the model of the sensor adapter in detail. The names of the various types of sensor products in the Pixhawk4 autopilot system are as follows: the IMU is ICM20689, the magnetic compass is IST8310, the barometer is MS5611 and the GPS receiver is Ublox M8N GPS. The simulation model of the above four sensor sensors based on the finite state machine(FSM) modeling approach as shown in Figure 4. The model is divided into two major phases: data processing and analogue communication. The data processing phase includes functions such as data type conversion, byte splitting, and FIFO storage, while the analogue communication phase covers response mechanisms such as chip identification, programmable register configuration, and data transmission. For this complex control logic process, the model adopts a finite state machine-based approach to modeling, which ensures the high efficiency of the communication process and the reliability of the system.

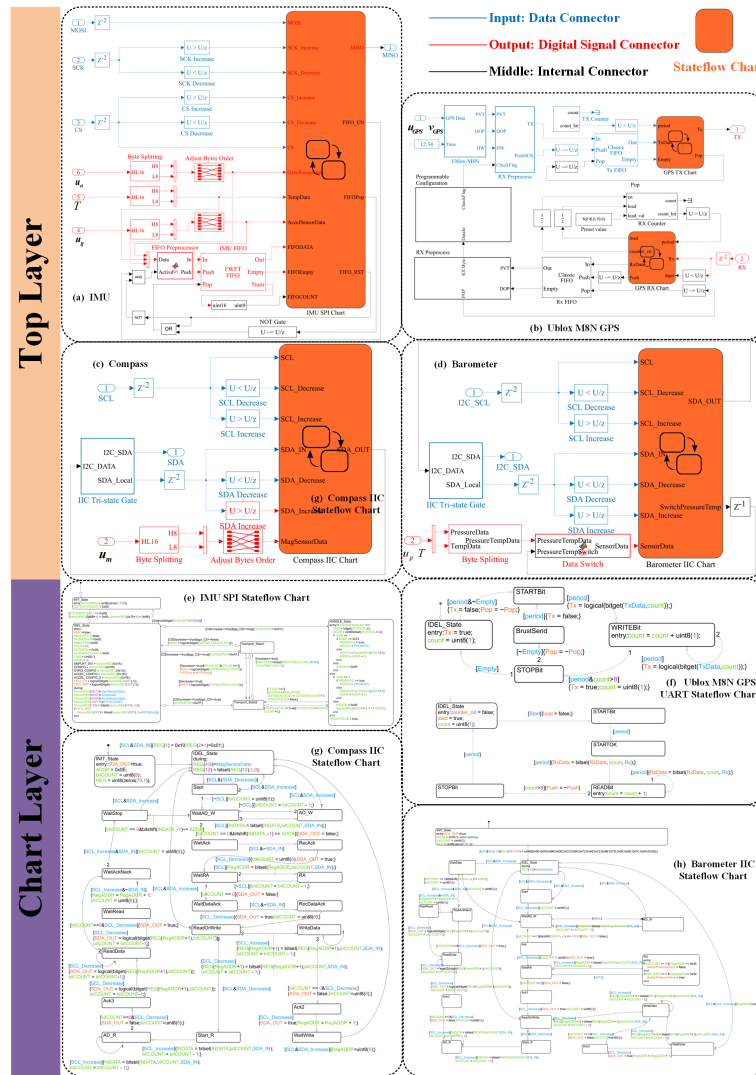


Figure 4. FPGA implementation of Sensor Adapter Model.

## 5. Sensor Fault Model

Fault is a common phenomenon in which the characteristics or parameters of an electronic equipment system deviate beyond the acceptable range, thereby degrading the performance of the system. The possible malfunction behavior of the autopilot system is concentrated on the sensors.

### 5.1. Measurement Model Fault Scheme

#### 5.1.1. Scale Fault

As the sensor is exposed to harsh environmental conditions (such as high temperature, high humidity, strong magnetic field, strong corrosion, etc.), the performance of the sensor material may be changed and degraded, resulting in distortion in the output of electronic devices such as transistor resistance, capacitance and inductance in the chip circuit. The scale fault output is shown as sensor data scaling, the fault scheme  $\Delta S_j$  is represented as

$$\Delta S_j = S_j \cdot \Delta s_j \quad (18)$$

where  $\Delta s_j$  indicates the proportional fault factor,  $j$  can be replaced by the characters  $a, g, m, p, GPS$ .

### 5.1.2. Bias Fault

Sensor bias faults often arise from several causes, such as component aging or damage, temperature effects, power supply variations, mechanical stress or damage, calibration mistakes, and environmental factors like magnetic fields, humidity, or chemical interference. These factors, individually or in combination, cause the sensor output to deviate from the true value by a fixed amount over the entire measurement range, the fault scheme  $\Delta b_j$  is represented as

$$\Delta b_j = b_j + \Delta b_j \quad (19)$$

where  $\Delta b_j$  indicates the bias fault factor.

### 5.1.3. Drift Fault

Sensor drift faults are the gradual deviation of a sensor's output from the true value over time, caused by material aging, environmental influences like temperature and humidity, and mechanical wear of components like ADCs and amplifiers. Drift represents a continuous, cumulative change in output, distinct from fixed offsets or proportional changes. For example, a temperature-affected sensor may gradually change its output due to changes in material properties with temperature, or a sensor subjected to mechanical stress over a long period of time may experience a gradual deviation of the output signal from the initial calibration point due to small changes in physical form. The measured signal  $u_j(t)$  of a drift fault scheme is expressed as

$$\Lambda(u_j(t)) = u_j(t) + \Delta\kappa(t - t_s) \quad (20)$$

where  $\Delta\kappa$  is the drift fault factor,  $t_s$  is the occur time.

### 5.1.4. Noise Fault

Sensor noise faults involve undesired random fluctuations in output, caused by external electromagnetic interference, internal electronic noise like thermal or shot noise, and environmental factors such as mechanical vibrations and temperature changes, particularly in sensitive sensors. The noise fault scheme  $\Delta\epsilon_j$  is represented as

$$\Delta\epsilon_j = \Delta\epsilon_j(N) \quad (21)$$

where  $\Delta\epsilon_j$  is the noise fault gain factor,  $N$  is the noise fault model.

### 5.1.5. Spike Fault

Sensors with microstructures like cantilevers and membranes can degrade or fail from external shocks, causing ruptures or detachment from anchor points, potentially leading to short circuits or performance issues. For instance, excessive pressure can rupture micro-membranes, and strong impacts can misalign internal elements, like causing a mass block in accelerometers to touch electrodes. Severe impacts may also damage tiny circuits and solder joints, affecting signal transmission and processing. And the spike fault can be regarded as a special form of scale fault, when the amplitude  $\Delta s_j$  is as large as possible and the spike fault duration time  $\Delta t_j$  is as small as possible, it can be described as follows

$$\Delta s_j \rightarrow \infty, \Delta t_j \rightarrow dt \quad (22)$$

where  $dt$  indicates a very small time.

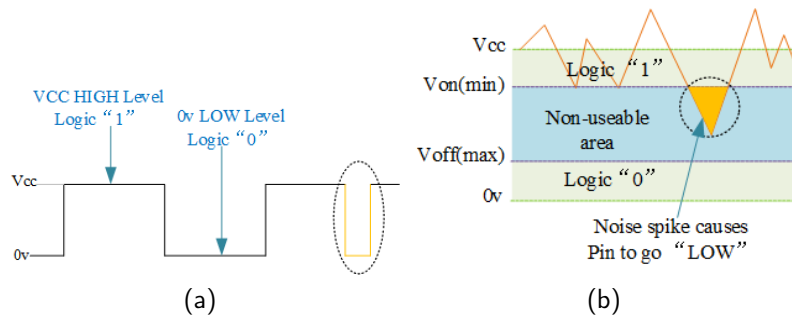
## 5.2. Communication Model Fault Scheme

### 5.2.1. Noise Fault

Due to the short distance communication, the sensor communication circuit uses a baseband digital circuit, that is, transmits high-level or low-level signals representing binary digital logic "0" or "1". The widely used transistor-transistor-logic(TTL) and complementary metal-oxide-semiconductor(CMOS) digital integrated circuit(IC) devices are affected by the characteristics of materials, effect of proton radiation [33], high-energy particles, etc., so the communication circuit does not have the ideal electrical signal characteristics, but parasitics a considerable amount of interference signals. The Figure 5 illustrates the general interference phenomena in digital communication circuits. For different digital circuits, the low-level interval of  $0V - V_{off(max)}$  is generally defined as digital logic "0", and the high-level interval of  $V_{on(min)} - V_{cc}$  is defined as digital logic "1". However, there is a "Non-useable area" between  $V_{off(max)}$  and  $V_{on(min)}$ , and the level signal falls in this area. Digital circuit logic interpretation will become unpredictable. Figure 5a shows a segment of digital signal that is disturbed by noise. The noise signal in Figure 5b is superimposed on the  $V_{cc}$  supply voltage, the voltage signal is kept above the minimum level  $V_{on(min)}$ , and the digital circuit logic interpretation will not be affected. But when the noise level becomes sufficiently large and the noise spike causes the high voltage level to drop below this lowest level, the digital circuit may interpret this spike as a logical "0". When the noise impact fault occurs, the level signal of the pin is a randomly varying 0 or 1, and the fault model is as follows

$$\hbar_N(x(n)) = rand([0, 1]). \quad (23)$$

where  $x(n)$  is the sensor chip pin digital signal,  $n$  is the sampling time.



**Figure 5.** Noise fault in digital logic circuit. (a) A disturbed digital signal. The ideal case is a high level, which becomes low when disturbed. (b) Schematic of noise interference in digital signals.

### 5.2.2. Open Circuit Fault

Sensor open-circuit faults typically result from a break in the circuit path, caused by physical damage to wires or solder joints, environmental corrosion, or damage to internal components from overheating or voltage issues, preventing current flow. Sensor open circuit faults can be represented as

$$\hbar_O(x(n)) = x(n) \& \Delta_{open} \quad (24)$$

where  $\&$  denotes AND operation sign, when  $\Delta_{open} = 1$  there is no fault, while when  $\Delta_{open} = 0$  it indicates an open-circuit fault.

### 5.2.3. Short Circuit Fault

Sensor short-circuit failures occur when an unintended conductive path redirects current flow, caused by insulation damage, moisture intrusion, external conductive materials like metal shavings, or

internal component damage due to over-voltage, overheating, or physical shock. Sensor short circuit faults can be represented as

$$\hat{h}_S(x(n)) = x(n) | \Delta_{short} \quad (25)$$

where  $|$  denotes OR operation sign, when  $\Delta_{short} = 0$  there is no fault, while when  $\Delta_{short} = 1$  it indicates an open-circuit fault.

### 5.3. Fault Inject Progress

Above all, we obtain the sensor measurement model fault parameters and communication model fault parameters, and the sensor measurement values with fault modes are further expressed as

$$\begin{cases} u_o^*(t) = \Lambda S_j \cdot u_j(t) + \Lambda b_j + \Lambda \varepsilon_j \\ u_o(t) = \Lambda(u_o^*(t)) \end{cases} \quad (26)$$

where  $u_o^*(t)$  represents the node measurement value before the drift fault processing unit in the signal chain of the measurement model. Likewise, the sensor communication digital signal with failure mode is further expressed as

$$\begin{cases} x(n) = \hat{h}_N(x^*(n)) \\ x(n) = \hat{h}_O(x^*(n)) \\ x(n) = \hat{h}_S(x^*(n)) \end{cases} \quad (27)$$

where  $x^*(n)$  represents the measurement value of the intermediate node in the communication model signal chain.

In addition, according to the form of fault existence, it can be divided into intermittent faults and permanent faults, which shows that faults have a time dimension. Therefore, the fault duration time  $\Delta t_d$  needs to be fully considered when designing the failure mode, indicating that the sensor is in an unhealthy state during this period.

We can develop a test case library necessary for the fault injection process based on engineers' development and testing experience and civil aircraft airworthiness test items and evaluation indicators. In order to form a standard test process, we stipulate that each test case command sequence needs to set the fault injection *trigger time*  $t_s$ , *test stop time*  $t_e = t_s + \Delta t_d$ , *expected sensor fault components*, *fault impact factors* as well as *flight tasks*, etc. Therefore, we can achieve the purpose of injecting faults online by modifying the fault model parameters. And the software and hardware platforms involved, are shown in Table 1.

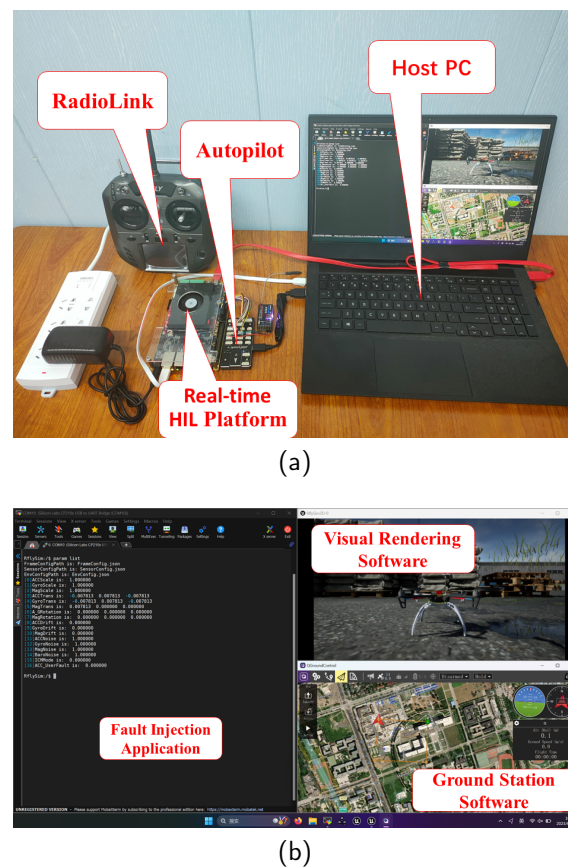
**Table 1.** HIL platform component list.

Devices	Performance, purpose and access
RHILP	AXU2CGB-E MPSoCs: <a href="https://www.xilinx.com/products/boards-and-kits/1-1jl2mmo.html">https://www.xilinx.com/products/boards-and-kits/1-1jl2mmo.html</a> MPSoC(2x ARM Cortex™-A53 processors): Running Figure 2 Model FPGA(XCZU2CG-1SFVC784E): Running Figure 4 Model
HC	GPU : NVIDIA Geforce RTX 3070 CPU : 12th Gen Intel® Core™ i7-12700F Rflysim3D: <a href="https://rflysim.com/en/4_Pro/Customizationof3DScenarios.html">https://rflysim.com/en/4_Pro/Customizationof3DScenarios.html</a> QGgroundControl: <a href="http://qgroundcontrol.com/">http://qgroundcontrol.com/</a> Mission planner: <a href="https://ardupilot.org/planner/">https://ardupilot.org/planner/</a> Fault Injection Application: (introduction in this article)
UVCS	Pixhawk4: <a href="https://docs.px4.io/v1.12/en/flight_controller/pixhawk4.html">https://docs.px4.io/v1.12/en/flight_controller/pixhawk4.html</a> CubeOrange: <a href="https://cubepilot.org/">https://cubepilot.org/</a> ZY-H7: <a href="https://doc.rflysim.com/hardware/2pixhwak/zy_H7.html">https://doc.rflysim.com/hardware/2pixhwak/zy_H7.html</a> PX4 Autopilot(PX4): <a href="https://px4.io/">https://px4.io/</a> ArduPilot Mega(APM): <a href="https://www.ardupilot.co.uk/">https://www.ardupilot.co.uk/</a>



## 6. Experiments and Verification

In this section, we design an MPSoC-FPGA-based HIL platform shown in the Figure 6 to fulfill the functional requirements of the testbed. First, we develop and validate the unmanned vehicle model, sensor model and fault model in Simulink and Stateflow environments based on the MBD modeling approach. Then the embedded code of each model is generated by the automatic code generation tool, followed by adding the fault injection interface and simulation data sending interface program in the IDE. Finally, the executables are cross-compiled and imported into the testbed device to run. In terms of performance, the platform uses two ARM-A53 1.2GHz cores to ensure that the unmanned vehicle model, the sensor measurement model and its part of the fault model computation results are accurate and up to 2KHz running frequency, and uses FPGA 100MHz to ensure that the sensor communication model and its part of the fault model ultra-high update frequency requirements.



**Figure 6.** HIL platform. (a) Hardware physical platform of HIL. (b) Software of HIL.

### 6.1. HIL Platform Function Verification

The most distinctive feature of the proposed HIL platform is to simulate the working mode of actual sensor products. In the case of the flight autopilot system, data interaction with sensors through digital communication is the same as the workflow of natural flight. The solution adopted in this paper removes the limitation of general hardware in the loop simulation scheme, so it has broader support. The functional verification of platform suitability was carried out from three aspects: flight control software platform, flight control hardware platform and unmanned system type.

We selected a number of representative hardware and software platforms that are widely used in scientific research and commercial fields for experimental verification. As shown in Figure 7, we selected two hardware platforms, PX4 and APM, and three hardware platforms, Holybro Pixhawk 4, CubePilot Cube Orange, and FEISILAB ZYH7. The two critical points of the HIL platform to adapt different kinds of flight controllers are as follows: the software platform modifies configuration options

to select the access mode of sensors as external, and it is necessary to shield internal sensors. The hardware platform needs to reserve the necessary communication interface, optional with multi-function carrier board.

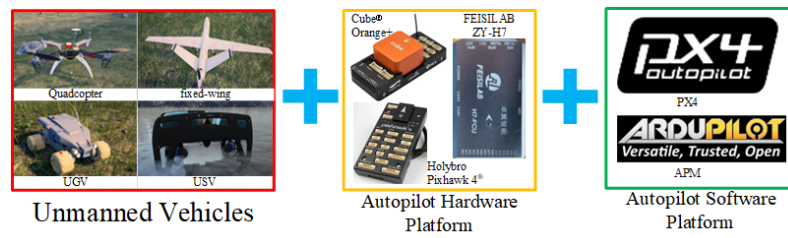


Figure 7. Validated HIL platform.

We established a variety of analytical models of unmanned systems, including multi-rotor model, fixed wing model, unmanned ship model and unmanned vehicle model, and completed the adaptation verification of the above four unmanned systems. This work shows that the proposed platform can be easily compatible with a variety of typical unmanned systems.

## 6.2. Simulation Reliability Assessment

To verify the reliability of the HIL platform, experiments and simulations are performed with accelerometer temperature drift fault test presented in Figure 8a. In the horizontal static environment, we obtained the acceleration values of x-axis, y-axis and z-axis of the ICM20689 sensor onboard the Pixhawk4 autopilot under the temperature variation range of 17°C – 50°C are presented as the ‘EXP Data’ curve in Figure 8b, from which it can be clearly observed that the accelerometer appears drift fault phenomenon with the change of temperature.

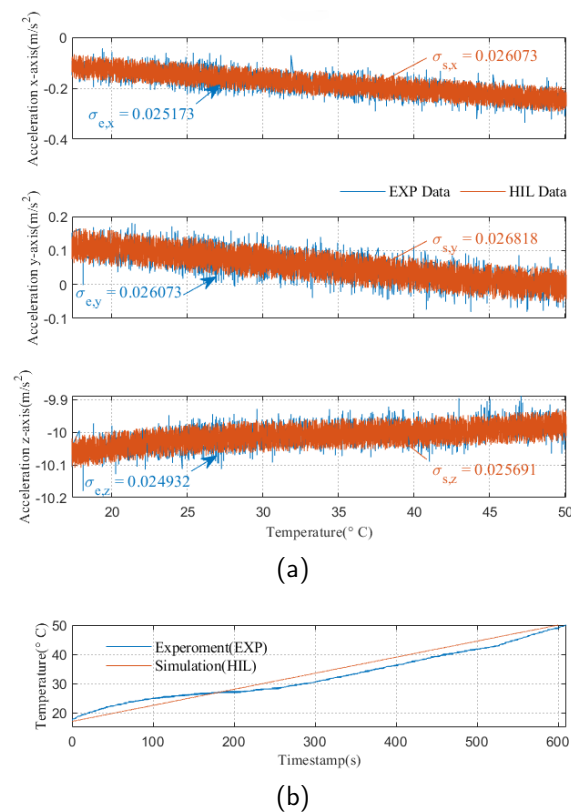


Figure 8. Accelerometer temperature drift fault test. (a)Temperature variation curve. (b)Acceleration curve.

From the data curve, the curves of simulation and experiment are highly coincident, which reveals the performance ability of HIL platform to simulate real fault behavior. Further, we adopted the evaluation method in literature [34] to quantitatively analyze the HIL reliability. We choose the performance credibility index  $\eta_p$  here. The root mean square error  $\sigma$  is selected as the performance parameter, the threshold parameter  $\varepsilon_p \approx 10\% \cdot |p_e|$  and the the simulation error for each axis is  $e_p = |\sigma_e - \sigma_s|$ . In the expectation of achieving a desired passing mark  $\eta_t \geq 0.6$ , the credibility evaluation index  $\eta_p$  was calculated as followed

$$\eta_p^i = \frac{0.75 \cdot \varepsilon_p^i}{\sqrt{\left(0.75 \cdot \varepsilon_p^i\right)^2 + e_p^{i\,2}}} \tag{28}$$

where  $i$  represents the x-axis, y-axis and z-axis. Then, the simulation credibility for each period  $\eta_p$  is obtained with the results listed in the Table 2. By combining accelerometer x-axis, y-axis, and z-axis credibility indices, the average simulation credibility is  $\bar{\eta}_t = 92.1251\%$ . Since the obtained credibility index  $\bar{\eta}_t$  is far above the passing mark  $\eta_p \geq 0.6$ , the simulation results can be considered as credible as experimental results.

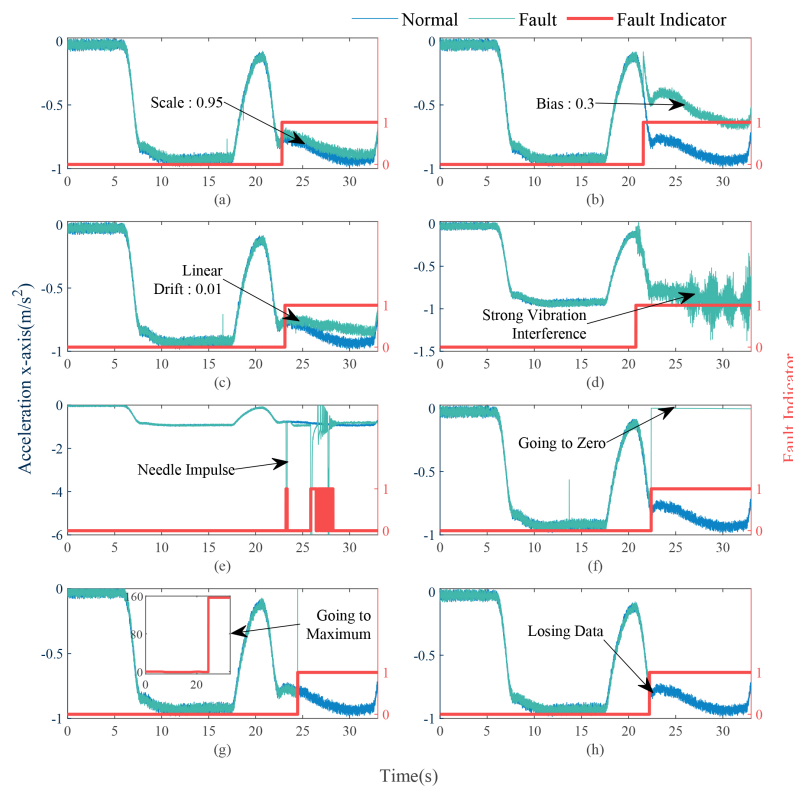
**Table 2.** Unssessment indices obtained for temperature drift fault test.

Test Phase	Parameter Error $e_t$	Error Threshold $\varepsilon_t$	Credibility Index $\eta_t$
x-axis	0.000900	0.002517	90.259640%
y-axis	0.000744	0.002607	93.463131%
z-axis	0.000759	0.002493	92.652503%

6.3. Proposed Platform Application

6.3.1. Fault Generation

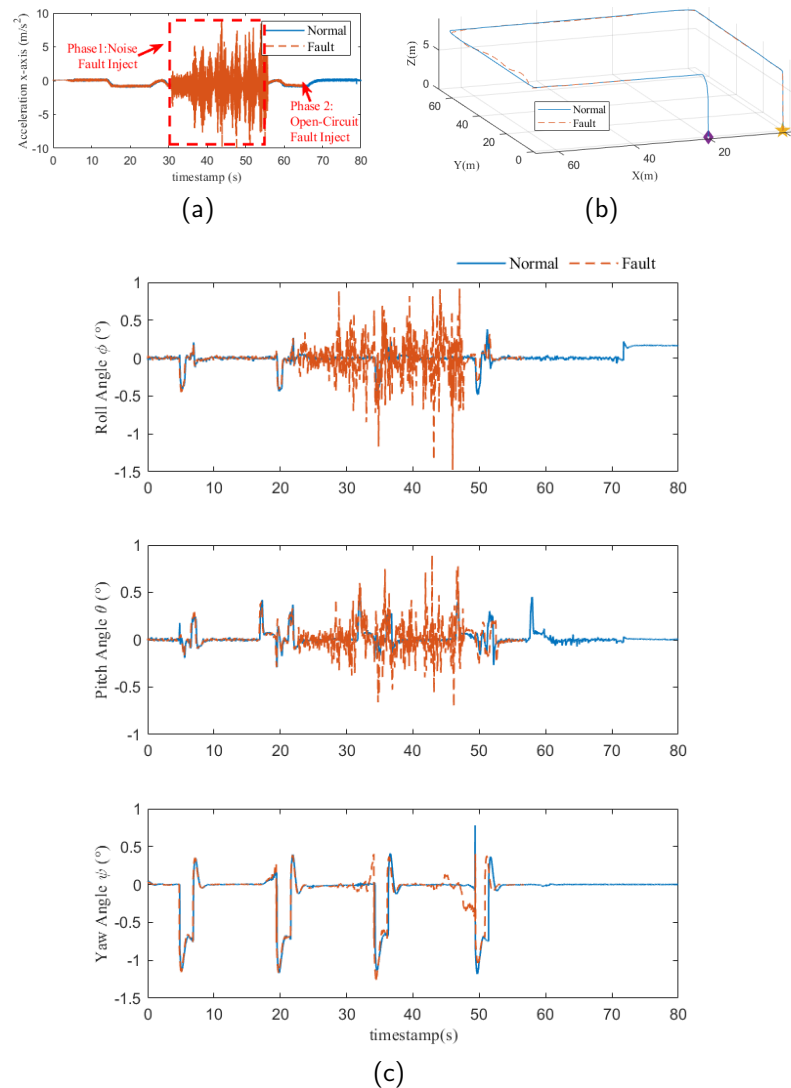
Multiple UAV flight missions are planned in the HIL platform, then injecting scale fault, bias fault, drift fault, noise fault, spike fault, internal open-short circuit fault for sensitive element, and open/short-circuit for the communication circuit into the accelerometer during the flight, as shown in Figure 9, we choose a simple expert model to perform fault identification and health assessment on fault data. It is worth noting that fault diagnosis and health assessment algorithms are not the core content of this article. This application case shows that test cases with multiple sensors, multiple failure modes, and different fault impact levels can be injected into the proposed HIL platform. The outstanding contribution of this work is that we can provide high-quality and large-scale test data for health management research work.



**Figure 9.** Accelerometer x-axis with fault injected. (a) Scale fault. (b) Bias fault. (c) Drift fault. (d) Noise fault. (e) Spike fault. (f) Open fault for sensitive element. (g) Short fault for sensitive element. (h) Open/short-circuit fault for communication circuit.

### 6.3.2. Reliability Testing

Here we present a study case of a UAV reliability flight. The autonomous flight mission shown in Figure 10b is scheduled in the HIL platform, then the accelerometer noise fault and open-circuit fault for communication circuit are injected successively in the flight mission. The data recorded by the HIL platform is shown in Figure 10, and the fault effect is shown in Figure 10a. The position and attitude of the drone are shown in Figure 10b. The test results as shown in Figure 10c presented the noise fault caused the drone body to undergo severe vibration and had a certain impact on the scheduled mission. On the other hand, the drone did not lose control due to severe noise fault, which shows that the tested unmanned system software platform has strong anti-noise performance; However, when the accelerometer suffered an open-circuit fault, the drone crashed immediately, causing huge damage to the fuselage. This shows that the unmanned system under test did not have hardware redundancy design.



**Figure 10.** A quadrotor reliability testing case. (a) Acceleration noise and open-circuit composite fault test case. (b) The quadrotor trajectory. (c) The Euler Angle attitude of the quadrotor.

This application case shows that we can perform reliability testing of unmanned systems on the proposed HIL platform in a low-cost and efficient manner. When the unmanned system model is sufficient and the fault model is rich enough, in theory, the relevant testing work of any unmanned system can be completed on the proposed HIL platform. In addition, since the proposed HIL platform supports complete flight testing and fault injection, it can also perform fault tolerant control of dynamic systems, and the research work [35] has been well verified.

#### 6.4. Videos and Source Code

The video gives an overall introduction to the test platform and an operational demonstration of fault generation, using the PX4 software platform, Pixhawk4 hardware platform and DJi F450 quadcopter drone:

<https://youtu.be/CMeeZYQ1eQ>

We have released the Matlab/simulink code:

<https://github.com/MsXFang/MPSoC-FPGA-HIL-Platform>

Engineers can easily adapt to the target open source platform by simply modifying the unmanned vehicle model and sensor model parameters.

## 7. Conclusions

Aiming at the demand of reliability verification of automatic driving system, this paper proposes an economical and practical HIL platform. In this paper, coarse-grained simulation models of various sensor products for unmanned vehicle and autopilot systems are first given. Then, the multi-mode failure models of various sensors are further given, which can realize credible chip-level failure simulation. Finally, the design and development of the HIL platform is completed on the MPSoC-FPGA development board, and the compatibility, accuracy and functionality of the HIL platform are further verified. The results show that the HIL platform proposed in this paper has outstanding advantages in intelligent unmanned control system. Future work will focus on refining various behavioral-level fault models so as to provide a richer library of fault use cases.

**Author Contributions:** Conceptualization, X.D.; methodology, X.D. and R.F.; software, X.D. and R.F.; validation, X.D. and R.F.; formal analysis, X.D. and R.F.; investigation, X.D. and R.F.; resources, X.D. and R.F.; data curation, X.D. and R.F.; writing—original draft preparation, R.F.; writing—review and editing, X.D. and R.F.; visualization, X.D. and R.F.; supervision, X.D. and R.F.; project administration, X.D.; funding acquisition, X.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Key Project of Research and Development Plan of China under Grant 2021ZD0140301.

**Data Availability Statement:** The datasets analyzed in this study are publicly available. Additionally, for convenience, we have provided download links for our modified datasets on GitHub repository: <https://github.com/MsXFang/MPSoC-FPGA-HIL-Platform>.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Dai, X.; Quan, Q.; Ren, J.; Cai, K.Y. An Analytical Design-Optimization Method for Electric Propulsion Systems of Multicopter UAVs With Desired Hovering Endurance. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 228–239.
2. Asignacion, A.; Satoshi, S. Historical and Current Landscapes of Autonomous Quadrotor Control: An Early-Career Researchers's Guide. *Drones* **2024**, *8*.
3. Baldi, S.; Roy, S.; Yang, K.; Liu, D. An Underactuated Control System Design for Adaptive Autopilot of Fixed-Wing Drones. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 4045–4056.
4. Shen, K.; Wang, M.; Fu, M.; Yang, Y.; Yin, Z. Observability Analysis and Adaptive Information Fusion for Integrated Navigation of Unmanned Ground Vehicles. *IEEE Trans. Ind. Electron.* **2020**, *67*, 7659–7668.
5. Raber, G.T.; Schill, S.R. Reef Rover: A Low-Cost Small Autonomous Unmanned Surface Vehicle (USV) for Mapping and Monitoring Coral Reefs. *Drones* **2019**, *3*.
6. Xing, L.; Johnson, B.W. Reliability Theory and Practice for Unmanned Aerial Vehicles. *IEEE Internet Things J.* **2023**, *10*, 3548–3566.
7. He, K.; Yu, D.; Wang, D.; Chai, M.; Lei, S.; Zhou, C. Graph Attention Network-Based Fault Detection for UAVs With Multivariant Time Series Flight Data. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–13.
8. Yang, P.; Geng, H.; Wen, C.; Liu, P. An Intelligent Quadrotor Fault Diagnosis Method Based on Novel Deep Residual Shrinkage Network. *Drones* **2021**, *5*.
9. Ma, Z.; Gong, H.; Wang, X. Fault-Tolerant Event-Triggered Control for Multiple UAVs with Predefined Tracking Performance. *Drones* **2024**, *8*.
10. Nguyen, N.P.; Pitakwatchara, P. Attitude Fault-Tolerant Control of Aerial Robots with Sensor Faults and Disturbances. *Drones* **2023**, *7*.
11. Zhongyi, C.; Zezhou, W.; Keqiang, X.; Shaoliang, L.; Bingye, Y. Health index construction and remaining useful lifetime prediction of aviation products based on multi-source degradation data fusion. 2022 Glob. Reliab. Progn. Health Manag. Conf., PHM-Yantai, 2022, pp. 1–5.
12. Xu, J.; Wang, Y.; Xu, L. PHM-Oriented Integrated Fusion Prognostics for Aircraft Engines Based on Sensor Data. *IEEE Sensors J.* **2014**, *14*, 1124–1132.
13. Wang, Y.; Wang, X.; Shen, L. Approximate Optimal Curve Path Tracking Control for Nonlinear Systems with Asymmetric Input Constraints. *Drones* **2022**, *6*.



14. Zhao, B.; Xian, B.; Zhang, Y.; Zhang, X. Nonlinear Robust Adaptive Tracking Control of a Quadrotor UAV Via Immersion and Invariance Methodology. *IEEE Trans. Ind. Electron.* **2015**, *62*, 2891–2902.
15. Xiao, K.; Tan, S.; Wang, G.; An, X.; Wang, X.; Wang, X. XTDrone: A Customizable Multi-rotor UAVs Simulation Platform. 2020 4th Int. Conf. Robot. Autom. Sci., ICRAS, 2020, pp. 55–61.
16. Khaliq, S.; Ahsan, S.; Nisar, M.D. Multi-Platform Hardware In The Loop (HIL) Simulation for Decentralized Swarm Communication Using ROS and GAZEBO. 2021 Proc. - IEEE Int. Symp. World Wirel., Mob. Multimed. Networks, WoWMoM, 2021, pp. 310–315.
17. Bahl, N.; Sahu, P.; Sharma, P.; Gautam, P.; Singh, A.; Srivastava, A. Design of an Autonomous UAV for Racing Applications. 2022 8th Int. Conf. Signal Process. Commun., ICSC 2022, 2022, pp. 91–96.
18. Szolc, H.; Kryjak, T. Hardware-in-the-loop simulation of a UAV autonomous landing algorithm implemented in SoC FPGA. 2022 Signal Process. - Algorithms, Architect., Arrange., Appl. Conf. Proc., SPA, 2022, pp. 135–140.
19. Guerra, W.; Tal, E.; Murali, V.; Ryou, G.; Karaman, S. Flightgoggles: A modular framework for photorealistic camera, exteroceptive sensor, and dynamics simulation. *arXiv preprint arXiv:1905.11377* **2019**.
20. Gong, S.; Meng, S.; Wang, B.; Liu, D. Hardware-In-the-Loop Simulation of UAV for Fault Injection. 2019 Progn. Syst. Heal. Manag. Conf., PHM-Qingdao, 2019, pp. 1–6.
21. Khan, A.; Ivaki, N.; Madeira, H. Are UAVs' Flight Controller Software Reliable? 2022 IEEE 27th Proc. IEEE Pac. Rim Int. Symp. Dependable Comput., PRDC 2022. IEEE, 2022, pp. 194–204.
22. Tsmots, I.; Teslyuk, V.; Łukaszewicz, A.; Lukashchuk, Y.; Kazymyrya, I.; Holovatyy, A.; Opotyak, Y. An Approach to the Implementation of a Neural Network for Cryptographic Protection of Data Transmission at UAV. *Drones* **2023**, *7*.
23. Hosseinabadi, A.H.H.; Black, D.G.; Salcudean, S.E. Ultra low-noise FPGA-based six-axis optical force–torque sensor: Hardware and software. *IEEE Transactions on Industrial Electronics* **2020**, *68*, 10207–10217.
24. Dai, X.; Ke, C.; Quan, Q.; Cai, K.Y. RFLySim: Automatic test platform for UAV autopilot systems with FPGA-based hardware-in-the-loop simulations. *Aerosp. Sci. Technol.* **2021**, *114*, 106727.
25. Ebeid, E.; Skriver, M.; Terkildsen, K.H.; Jensen, K.; Schultz, U.P. A survey of open-source UAV flight controllers and flight simulators. *Microprocess. Microsy.* **2018**, *61*, 11–20.
26. Quan, Q., Dynamic Model and Parameter Measurement. In *Introduction to Multicopter Design and Control*; Springer Singapore: Singapore, 2017; pp. 121–143.
27. Agency, D.M. Department of Defense World Geodetic System 1984: its definition and relationships with local geodetic systems. *Defense Technical Information Center* **1991**.
28. Huang, F.; Wang, Z.; Xing, L.; Gao, C. A MEMS IMU gyroscope calibration method based on deep learning. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–9.
29. Trawny, N.; Roumeliotis, S.I. Indirect Kalman filter for 3D attitude estimation. *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep* **2005**, *2*, 2005.
30. Li, H.; Gou, L.; Chen, Y.; Li, H. Fault Diagnosis of Aeroengine Control System Sensor Based on Optimized and Fused Multidomain Feature. *IEEE Access* **2022**, *10*, 96967–96983.
31. Bao, X.; Xiong, Z.; Sheng, S.; Dai, Y.; Bao, S.; Liu, J. Barometer measurement error modeling and correction for UAH altitude tracking. 2017 29th Proc. Chin. Control Decis. Conf., CCDC. IEEE, 2017, pp. 3166–3171.
32. Ahuatzin-Flores, G.; Tang, Y.; Espíndola-López, E. Navigation observer design using vector measurements and a GPS sensor. *IEEE Sensors J.* **2023**.
33. Chen, S.; Zhao, Q.; Cui, J. Effect of Proton Radiation on Mechanical Structure of Silicon MEMS Inertial Devices. *IEEE Trans. Electron Devices* **2022**, *69*, 5155–5161.
34. Dai, X.; Ke, C.; Quan, Q.; Cai, K.Y. Simulation Credibility Assessment Methodology With FPGA-based Hardware-in-the-Loop Platform. *IEEE Trans. Ind. Electron.* **2021**, *68*, 3282–3291.
35. Ke, C.; Cai, K.Y.; Quan, Q. Uniform Fault-Tolerant Control of a Quadcopter With Rotor Failure. *IEEE/ASME Trans. Mechatron.* **2023**, *28*, 507–517.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.