CEAS Aeronautical Journal (2021) 12:549–558
https://doi.org/10.1007/s13272-021-00509-7

ORIGINAL PAPER

Development of a hardware-in-the-loop demonstrator for the validation of fault-tolerant control methods for a hybrid UAV

F. Prochazka[1] · S. Krüger[1] · G. Stomberg[2] · M. Bauer[1]

Received: 28 April 2020 / Revised: 11 February 2021 / Accepted: 31 March 2021 / Published online: 5 May 2021

Abstract

Many aircraft are inherently over-actuated with regard to their input variables. This can be particularly advantageous in the context of unmanned aerial vehicles (UAV), where actuator functions can fail in critical situations. In such cases, the redundant actuators can be used to further fulfil the control strategies used and thus increase the operational safety. Within such an active fault-tolerant control system, a fault detection and isolation (FDI) module is required. To evaluate such safety–critical systems, hardware-in-the-loop simulations (HIL) are a necessary step prior to real flight tests. These simulations can verify the correct implementation of the flight controller on the target hardware as well as the real-time capability of the algorithms used. Particularly in the context of active fault-tolerant control, investigations concerning the robustness of the used FDI module with regard to real, noisy sensor signals, which can be generated by a HIL demonstrator, are of utter importance. This paper presents the development of a HIL demonstrator for the validation of fault-tolerant control methods for a hybrid UAV. This includes a detailed description of the demonstrator's design, control and interfacing between the integrated subsystems. As an application example, a hybrid UAV model will be shortly presented, which, in addition to the primary aerodynamic control surfaces, can also use four lift rotors to control the aircraft during cruise and is therefore inherently over-actuated. Finally, a closed-loop real-time simulation of the UAV model on the HIL demonstrator is presented on the basis of the exemplary simulation of an actuator failure and subsequent reconfiguration by the fault-tolerant flight control law.

Keywords UAV · Hardware-in-the-loop · Fault-tolerant control · Flight control

1 Introduction

With an increasing number of unmanned aerial vehicles (UAVs) predicted for the near future [1], the integration of unmanned aerial systems (UAS) in the civil airspace will become a topic of major relevance. In [2], it is stated that for such levels of UAS integration different safety and reliability requirements will need to be fulfilled. This includes the flight control system's (FCS) robustness in the face of failure occurrences, e.g., to faults in the primary actuators. In [2], the need to validate implemented autopilot algorithms using sophisticated hardware-in-the-loop (HIL) simulations is also highlighted. This indicates a direct approach towards established clearance procedures for UAV flight control laws in civil aviation [3]. Therefore this paper presents the development of a HIL demonstrator which makes it possible to include commercial of-the-shelf (COTS) components, which are critical for the stability and performance of the FCS, directly into a real-time closed-loop HIL simulation. The integrated subsystems include a typical RC servo actuator, a consumer level inertial measurement unit (IMU) and a *Pixhawk 4* flight controller, which is the de-facto standard flight control computer (FCC) for sophisticated UAV applications [4]. A very interesting type of upcoming UAV configurations is a hybrid system UAV, which combine's the capabilities of a multicopter to hover with the far more energy-efficient cruise flight properties of a fixed-wing aircraft. As the detailed mathematical modelling procedure of such a hybrid configuration from wind tunnel and flight tests has been presented in [5], here only a brief description of a generic hybrid UAV model is given. Furthermore, real-time

✉ F. Prochazka
prochazka@fsr.tu-darmstadt.de

S. Krüger
krueger@fsr.tu-darmstadt.de

[1] Institute of Flight Systems and Automatic Control, Technical University of Darmstadt, Darmstadt, Germany

[2] Institute of Energy Systems, Energy Efficiency and Energy Economics, TU Dortmund University, Dortmund, Germany

HIL simulation results for a fault-tolerant controller based on the UAV model are shown, which enables the UAV to follow a desired tracking command even in the face of a total failure of an aerodynamic control surface.

This paper is organized as follows. In "Sect. 2" the developed HIL demonstrator's subsystems and interfaces are presented. The integration of the flight controller as well as the design of a three-axis rotary stage are described in detail. "Section 3" describes the hybrid UAV model as well as a servo actuator subsystem, which have been used for the FTC controller design and HIL integration. Furthermore, exemplary experimental simulations results are given. "Section 4" concludes the paper and gives an outlook on future research.

2 Hardware-in-the-loop demonstrator

In this section, the developed hardware-in-the-loop demonstrator is described in detail. "Section 2.1" begins by introducing the HIL target-machine and the interfaces to the integrated hardware under test. The integration procedure and latency evaluation of the *Pixhawk 4* flight control computer (FCC) is presented in "Sect. 2.2". "Section 2.3" describes the conceptual design and construction of the 3 degrees of freedom (3DOF) rotary stage and presents a validation procedure for the integrated IMU sensor.

2.1 Real-time target and interfacing

A *Speedgoat* Baseline Education real-time target machine forms the core of the HIL demonstrator. The target machine is used in combination with an FPGA-based I/O module (IO397) and runs *Simulink RealTime*. The target machine comes with an Intel 2 GHz quad core CPU,4 GB RAM and 32 GB SSD for data logging. It further offers a wide range of interfaces, thus enabling the integration of different hardware components using the *Simulink* simulation software. Figure 1 gives an overview over the different hardware components and their interfaces that are installed in the demonstrator. The host PC deploys compiled code from Simulink to the target-machine. Developed control laws are compiled using *MATLAB's* Embedded Coder and are deployed on the FCC unit. The host PC also serves as a data sink for visualizing the UAVs motion in *X-Plane* during the real-time simulation. The aircraft's equations of motion are solved on the *Speedgoat* target and simulated sensor outputs are sent to the FCC unit via a UDP server. In typical flight control systems, the aircraft's angular rates are measured by an IMU and are fed to the controller. Because these signals correspond to the innermost control loop and are crucial for closed-loop stability a typical low-cost IMU for UAV applications has been integrated in the setup to be able to evaluate the stability and performance of developed control laws using real gyroscope measurements. For this purpose, the *Speedgoat* has been configured as an EtherCAT master node and it sends the calculated angular rates of the UAV to the 3DOF rotary stage as reference values for the three axes' motor controllers. The *Nanotech* CR-E-1–21 motor controllers are installed as EtherCAT slaves in a daisy-chain configuration with a sampling rate of 1 kHz and actuate the rotary stage's frames in such a way that the mounted IMU is stimulated to measure the commanded angular rates. This real sensor data is then fed to the flight controller via a serial I2C interface. The controller calculates desired control inputs and feeds them back to

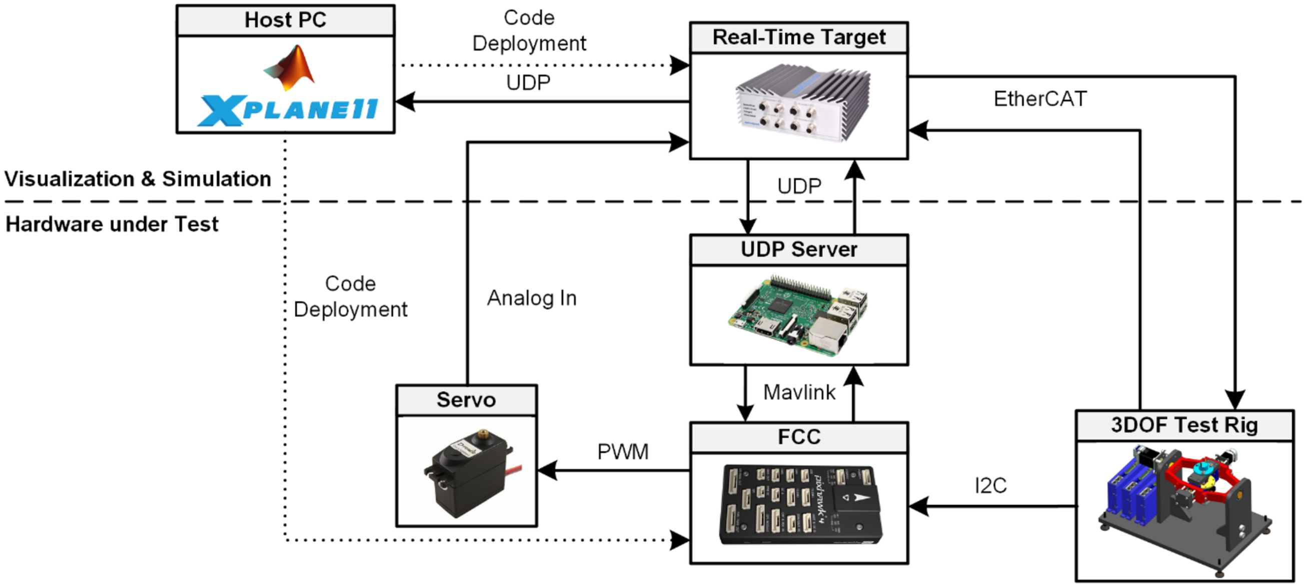

Fig. 1 Hardware-in-the-loop components and interfaces

the target-machine, where actuator dynamics are simulated. The FCC unit also commands a *D-Power* DS-570BB MG servo actuator. Actuators of this type are commonly used in UAVs to drive the primary control surfaces. The servo consists of a DC motor, a gearbox, a potentiometer and a circuit board. The potentiometer measures the output shaft position and the circuit board creates a feedback loop to match the output shaft position with the demanded value. The position is encoded in a pulse-width modulated input signal and the servo is powered by a supply voltage of 5 V. The servo provides an output torque of up to 0.70 Nm and meets the load requirements of the investigated UAVs for true airspeeds up to approximately 40 m/s. To obtain real-time measurements of the elevator deflection, the actuator was modified and additional wires were soldered to the circuit board to rout the potentiometer voltage to the real-time targets analog I/O channels.

2.2 Flight control computer

A *Pixhawk 4* flight management unit (FMU) is incorporated into the demonstrator to test the feasibility of the employed control schemes [4]. The real-time capable FMU is designed for use in UAVs where it reads sensor data, commands actuators, communicates with the pilot and runs the control logic. Several methods of integrating the *Pixhawk 4* into HIL demonstrators have been presented in the past. In [6], the *Pixhawk* is connected to a PC via a serial connection and the PC runs *jMAVSim* to simulate a hexacopter. In [7, 8], the *Pixhawk* is connected to a PC via a serial connection and the PC runs dynamics simulations in C-code that have been generated from Simulink using the *Matlab Embedded Coder*. In [9], the *Pixhawk* is again connected to a PC via a serial connection and *Gazebo* is used to simulate the plant on the PC. The HIL demonstrator presented in this contribution however relies on the *Speedgoat* target machine to interface further hardware components such as sensors and actuators. The approaches mentioned above can therefore not be applied here as the *Pixhawk*'s commands have to be processed in *Simulink* directly. Even though *Mathworks* provides the *Pixhawk Support Package* to transfer Simulink models on to the *Pixhawk 4*, the real-time communication with a Simulink real-time target machine is not supported. Therefore, a *Raspberry Pi 3B* is used to handle the communication between the target machine and the *Pixhawk*. The *Raspberry Pi* runs the *MavLinkBridge* software [10]. The software forwards *MAVLink* messages from the target machine to the *Pixhawk*. For the opposite direction, *MAVLink* messages from the *Pixhawk* are decoded and the payload is sent to the target machine via UDP messages. Two serial connections are established between the *Raspberry Pi* and the *Pixhawk* to use one connection for sending messages in each direction.

The integration of the *Pixhawk* FCC via a Raspberry Pi and the *MavLinkBridge* software is validated by real-time simulations with a sample rate of 100 Hz. The real-time target machine simulated a linearized UAV plant and a custom flight controller was employed on the *Pixhawk*. A timestamp was sent to the *Pixhawk* in addition to the system states and the command input. The *Pixhawk* computed the control signals and returned them together with the timestamp to the target machine. The timestamps were logged and analyzed to yield the overall latency including all delays due to communication as well as the time taken to compute the control signals. Figure 2 shows a histogram of the measured latencies for five simulations taking two minutes each. The mean latency was 6 ms and a maximum value of 9 ms has not been exceeded. These results agree well with results from [8] where a round-trip maximum latency of 8.53 ms between a PC and a *Pixhawk* including the time spent to execute one simulation step on the PC are reported. If Simulink blocks were available to handle the communication via the *MAVLink* protocol, the *Raspberry Pi* could be removed from the setup in the future and the latency could possibly be reduced further.

2.3 Three degrees of freedom stage

The mechanical part of the HIL test rig allows to excite the IMU in all rotational degrees of freedom like a three-axes gimbal (3-DOF-Stage). The aim is to simulate the roll, pitch and yaw rates $\omega(t) = [\, p(t) \;\; q(t) \;\; r(t)\,]^{\mathrm{T}}$ of a hybrid vertical take-off and landing (VTOL) UAV as accurately as possible. This results in special requirements for the design of the actuators, as the transmission bandwidth between commanded and resulting angular velocity of the 3-DOF-Stage has to be substantially higher than the aircraft's rotary dynamics.The sensor selection has a major influence on the design, since the size and weight of the sensor determines the dimensioning of all three axes. Before choosing the IMU, requirements regarding the rotational rates are to be defined. Therefore the UAVs maximum rotational rates in Table 1 are

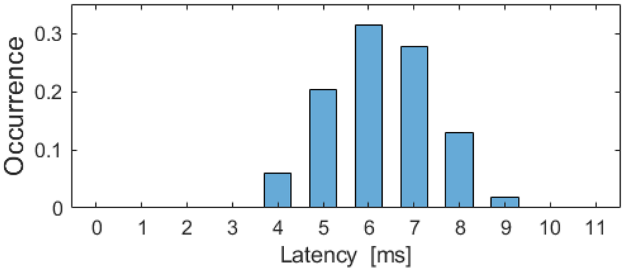

Fig. 2 Latency histogram of combined FCC computing and transmission time

Table 1 Max. angular rates of VTOL UAV and chosen IMU

Axis	Max. angular rate (with margin) VTOL UAV	Max. angular rate IMU
Roll	± 120 °/*s*	± 250 °/*s*
Pitch	± 135 °/*s*	± 250 °/*s*
Yaw	± 45 °/*s*	± 250 °/*s*

approximated using a simulation model of the VTOL UAV and adding an additional safety margin.

It is noteworthy that by including the lift motors as control inputs the maximum resulting pitch rate was higher than for the conventional aircraft mode. The derived requirements shall be met by both IMU and 3-DOF-Stage.

The selection of the IMU takes place under consideration of the current standards in the UAS field. On one hand, the IMU should achieve the highest quality measurement results possible, on the other hand, it should reflect as realistically as possible which type of IMU is used in COTS UAS. For the Micro or Mini UAV class, MEMS IMUs are generally used [11] for cost and weight reasons. At the time of writing, the Invensense MPU-9250 IMU was chosen. It describes a sensor package, which contains three gyroscopes and three accelerometers (Invensense MPU-6500) as well as three additional magnetometers (AsahiKasei AK8963).

The axes are generally [12] arranged in such a way that three rotatable frames are nested into each other so that all axes of rotation are orthogonal to each other and intersect at the center (IMU position) as depicted in Fig. 3.

The rotational rates of roll, pitch and yaw frame are described by $\dot{\varphi}(t) = [\ \dot{\varphi}_x(t)\ \ \dot{\varphi}_y(t)\ \ \dot{\varphi}_z(t)\ ]^T$. Since in aerodynamic flight for a typical aircraft, the roll rate p has the highest dynamic and the yaw rate r the lowest, the yaw rate is mapped in the innermost (yaw) frame and the roll rate in the outermost (roll) frame. This makes it possible to freely design the roll rate actuator, since it has no influence on the remaining rotating frames. The frames and the associated actuators are thereby designed from the inside out.

First, the yaw frame including the actuators is designed according to the size and weight of the used IMU.

Second, the requirements for the dimensions of the pitch frame and the associated actuators are derived, since the pitch actuator must also move the yaw frame including the actuator. Last, the roll actuator is selected, which must move the pitch and yaw frames.

In addition, counterweights are attached to the rotating frames to minimize the required holding moments for the actuators as well as possible imbalances. Since the actuators are procured as a finished product, they must be selected iteratively based on their weight and the available actuating force. To be able to simulate steady turning flight scenarios, the range of motion for the inner frame has to be unlimited. To allow for arbitrarily large angles in the yaw axis, a slip ring must be provided to be able to supply and read out the installed IMU.

The final design is shown in Fig. 3. Due to the low weight of the IMU, the yaw frame (marked yellow) was implemented as a yaw disc which is mounted directly on the yaw motor's shaft without bearing. The sensor cables (IMU) are led into the slip ring, which is fixed on the pitch counterweight. The pitch counterweight and yaw motor are both attached to the pitch frame (marked blue). The pitch frame is U-shaped and rotatably mounted on the roll frame (marked red). The pitch motor and roll counterweight are also mounted here.

The roll-frame is rotatably mounted on the base (marked grey/black) and is driven by the roll motor which is also mounted on the base. On each axle there is a servo stepper

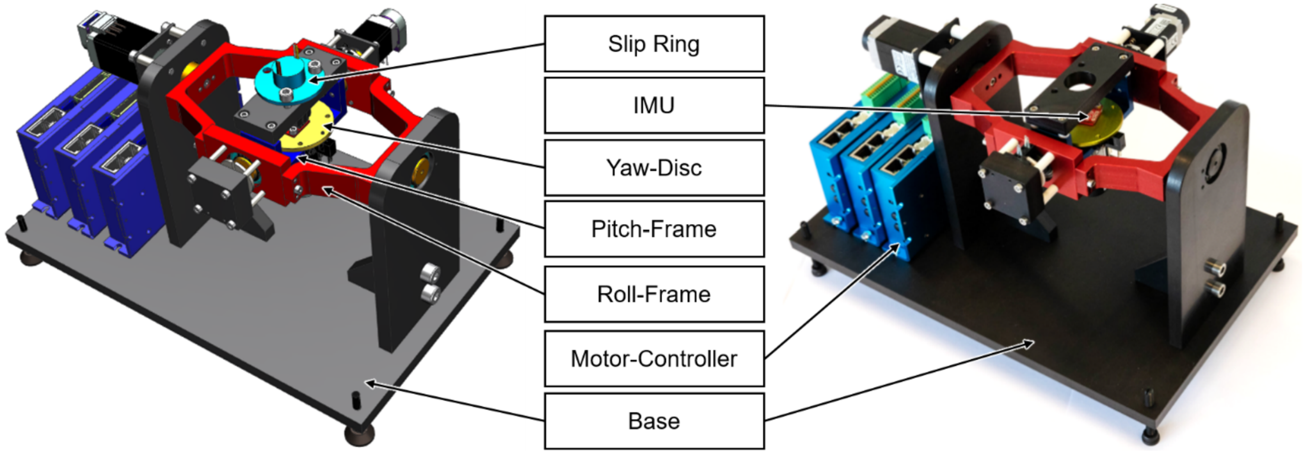

Fig. 3 Introducing the mechanical 3-DOF-Stage both as CAD drawing and finished product

motor by Nanotech which comes with a pre-installed incremental encoder to measure all frames position (angle) and rates ($\varphi_{meas}(t)$, $\dot{\varphi}_{meas}(t)$).

The needed motor torque is calculated by simulating the maximum rotational acceleration of the UAV model and the 3-DOF-stages moment of inertia of each axis. The resulting requirements and chosen actuator specifications are displayed in Table 2. To define the absolute position, optical sensors and light barriers are placed on each axis so that the absolute position can be determined by a reference run.

The angular velocity is used for internal control in the motor controllers. With the angular rates $\dot{\varphi}_{meas}(t)$ provided by the motor controllers and the measured motor currents $i(t)$ a 2-stage cascade controller is implemented on the motor controller for each frame.

In the innermost cascade, the motor torque $m(t)$ is controlled by setting the needed current $i(t)$. The rotational rates $\dot{\varphi}(t)$ are controlled in the outer cascade. The controller structure is shown in Fig. 4. The rotational rates of the model are measured on the IMU $\omega_{meas}(t)$ and transferred to the FCC (*Pixhawk4*) in the finished setup. The FCC calculates the needed commands $\mathrm{u}_{\mathrm{cmd}}(t)$ and transfers them to the model running on the real-time computer (*Speedgoat*). The UAVs reaction to these inputs is simulated and transformed into reference variables $\dot{\varphi}_{\mathrm{cmd}}(t)$, latter being used as the commanded inputs for the 3-DOF-stage.

To test the interfaces between the real-time computer and the 3-DOF stage, the *Speedgoat* is used to command pitch rate steps to the motor controller. Different step amplitudes are tested and the outputs of the pitch motor encoder (via *EtherCat*) and the measured rotation rate of the gyro around the Y-axis (via I2C) on the real-time computer are compared. In Fig. 5 two-step responses are plotted for 120°/s (top) and 60°/s (bottom) angular rates. As the gyro measurements are in accordance with the motor controller's readings, it can be concluded that the setup works properly in real-time. The motor controllers settling time of around 100 ms is sufficient to reproduce the rotary dynamics of a small-sized hybrid UAV. Furthermore, the settling time can be improved by increasing the max. acceleration in the motor controllers' settings.

Table 2 Required and chosen actuator specifications

Axis	Required torque	Required acceleration	Max. torque
Roll	16.3 Ncm	2280 °/s^2	18 Ncm
Pitch	1.74 Ncm	1320 °/s^2	2.2 Ncm
Yaw	0.0015 Ncm	240 °/s^2	2.2 Ncm

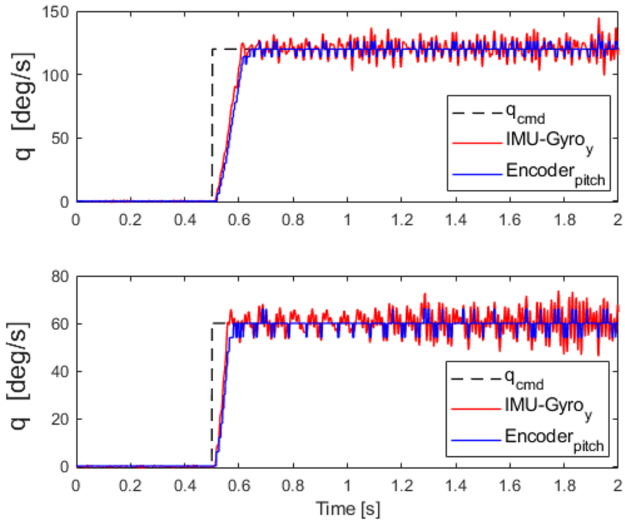

Fig. 5 Pitch rate step response of 3-DOF-Stage and IMU

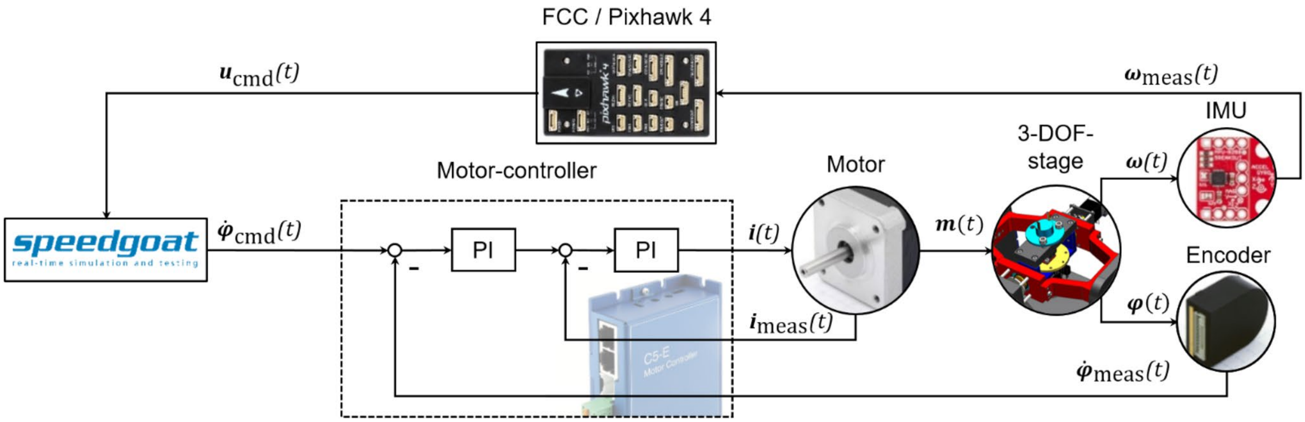

Fig. 4 3-DOF-stage cascaded control loop and hardware under test integration

3 Simulations and experimental results

In this section, exemplary simulation results are presented for an actuator fault scenario and a fault-tolerant controller which has been designed for a longitudinal hybrid UAV model. In "Sect. 3.1", the utilized UAV model is shortly introduced and the modelling of the servo actuator dynamics is presented. In "Sect. 3.2", the overall simulation structure for HIL testing is shown and the closed control loop setup is briefly described. HIL simulation results are presented in "Sect. 3.3".

3.1 Hybrid UAV and servo modelling

The hybrid UAV configuration under consideration has been introduced in [13]. As the mathematical modelling procedure for the aircraft is not the main focus of this contribution, the principle procedure is only shortly described. In [5], detailed modelling results from wind tunnel and flight tests with the institute's hybrid UAV demonstrator are presented. In this work, a dynamical model of the UAV's servo actuator has been developed using the corresponding HIL interface described in "Sect. 2.1". and the results are presented at the end of this section. This model can be used in preliminary controller design stages and simulations prior to HIL tests.

Based on initial results from [13], computational fluid dynamics (CFD) simulations have been performed in *Ansys Fluent 19.0* to identify the aircraft's aerodynamic derivatives as well as cross-coupling effects between the airframe and rotating lift rotors. Figure 6 shows the CAD model of the UAV and a visualization of the simulated airflow around the lift rotors during cruise flight from *Ansys Fluent*.

To evaluate the accuracy of the employed CFD analysis, classical aerodynamic derivatives have been determined from the numerical simulations and compared to results from typical aerodynamic analysis tools. AVL (*Athena Vortex Lattice*) and DATCOM. The UAV's CAD model has also been integrated in the *X-Plane* flight simulator software and derivatives have been estimated from simulated flights using an Extended Kalman Filter (EKF). The achieved results are in good accordance and also allow for an estimate of derivative uncertainty levels to be included in future robust controller design procedures. Figure 7 shows the exemplary comparison between CFD results and the mentioned tools for the $C_{L'}$, $C_{L\alpha}$, $C_{m\alpha}$ aerodynamic coefficients. When superpositioning the aerodynamic forces of the fixed-wing-structure and the thrust of the rotors, the cross-coupling between these two must be considered. The incident flow from fixed-wing-flight influences the thrust generated by the rotors while the turning rotors themselves influence the incident flow of the trailing structure. To distinguish between these two effects, the thrust and the aerodynamic forces have been stored separately in *Ansys* while varying the influential parameters rotational speed, airspeed and incident flow direction. After identifying main influences many more operating points were measured, thus generating many data points to quantify the observed effects. Subsequently regression analysis was carried out to extend the preliminary models of aerodynamic derivatives and rotor thrust. These models have then been integrated in a typical 6 DOF rigid body model [14].

To model the dynamics of a servo similar to the ones integrated in the hybrid UAV at hand a linear time-invariant model of order three including motor dynamics, armature current dynamics and feedback control has been identified in [15] using step input signals. Here, the dynamics of the armature current and the servo feedback control are neglected, which leads to a second order system. In addition, the angular velocity of the output shaft is limited as in [16] to account for saturation effects of the real actuator. This leads to the model description

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & -2D\omega_0^2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ K\omega_0^2 \end{pmatrix} \eta_d \tag{1}$$

$$|x_1| \leq \eta_{\max}$$

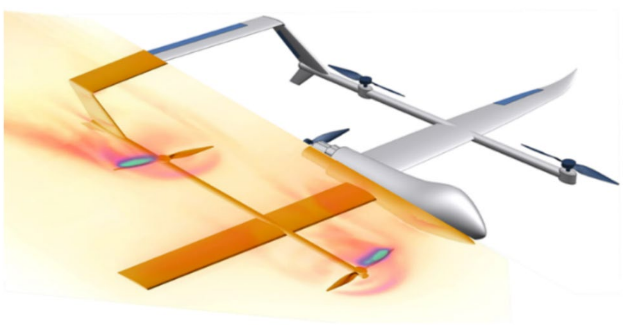

Fig. 6 CFD Visualisation and hybrid VTOL UAV

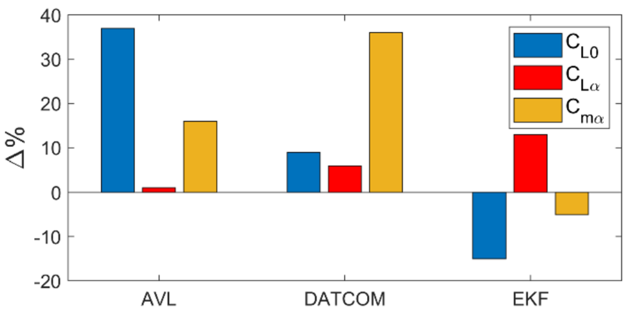

Fig. 7 Percentual deviation of aerodynamic derivatives for other tools compared to CFD results

$|x_2| \leq \dot{\eta}_{\max}$

for the elevator deflection $x_1 = \eta$ and the desired deflection η_d. Load effects are accounted for in [16] by limiting the angular velocity to a maximum load scenario whereas [15] neglects load effects. To further investigate the effects of load on the gain K, the eigenfrequency ω_0^2 and the damping D of the servo, a test setup has been constructed which can simulate four different load scenarios. To this end, an exchangeable torsion spring is attached to the elevator hinge such that four different stiffness values can be used for experiments. The spring torque is proportional to the elevator deflection and represents the aerodynamic load. Neglecting transient aerodynamic effects and assuming sea level conditions, the setup allows to apply load conditions similar to true airspeeds of 0, 13, 18 or 26 m/s. Input signals with multiple steps and variable step heights between 1° and 5° are used for the system identification. Figure 8 shows measurement data for different loads together with the identified model for idle load. The model parameters were identified as $K = 1$, $D = 0.9158$, $\omega_0 = 26.3799$ rad/s and $\dot{\eta}_{\max} = 4.86$ rad/s. The elevator deflection is limited to $\eta_{\max} = 30°$. In addition to the shown data, step heights up to 30° were investigated. It was found that the reaching time is significantly influenced by the step height. However, Fig. 8 shows that load effects for airspeeds up to 18 m/s have little influence on the servo motor step response. The influence of aerodynamic loads on the servo motor is thus negligible for large parts of the flight envelope and hence the idle case is used for further HIL simulations. Furthermore, Fig. 8 shows that the actuator is subject to a finite positional resolution of 0.2°. Additional experiments yielded a dead band of 0.5°.

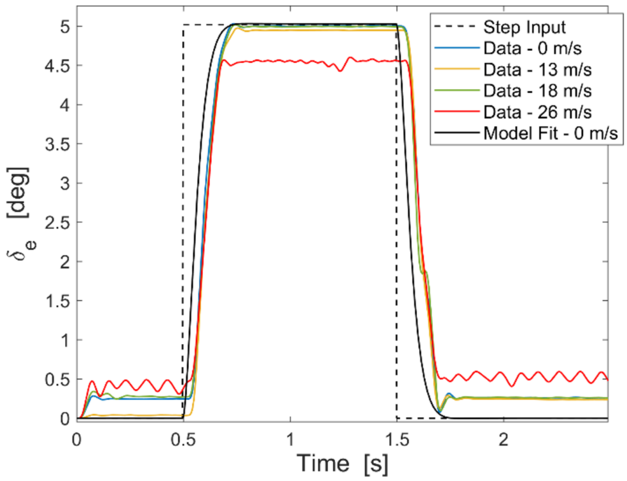

Fig. 8 Servo response under varying loads

3.2 Closed control loop setup

All simulation models and interfaces to HIL hardware components have been implemented in Simulink. For simulations on the target machine, a fourth order Runge–Kutta solver with a fundamental step size of $\Delta t = 0.001s$ is used. Figure 9 shows the general structure of the simulation setup. The flight controller (FCC), aircraft model including actuator and sensor models as well as environmental disturbances like wind gusts and turbulences can be simulated on the target machine. The FCC module, elevator servo actuator and IMU can alternatively be replaced by the corresponding real hardware components.

The general structure of an active fault-tolerant control system is shown in Fig. 10. When a fault occurs, e.g., a loss of effectiveness of a control surface, a module for fault detection and diagnosis (FDD) identifies and isolates the fault**.** Based on this information, the baseline control law is reconfigured online to accommodate for the fault acting on the system. For such an approach, it is generally necessary that the system is over-actuated, i.e., it has more control inputs than needed to control the closed loop's output variables.

The implemented FTC controller has the same structure as the setup presented in [13] and is described there in detail. It employs the nonlinear dynamic inversion (NDI) technique [14] for the baseline controller. The NDI approach is suitable

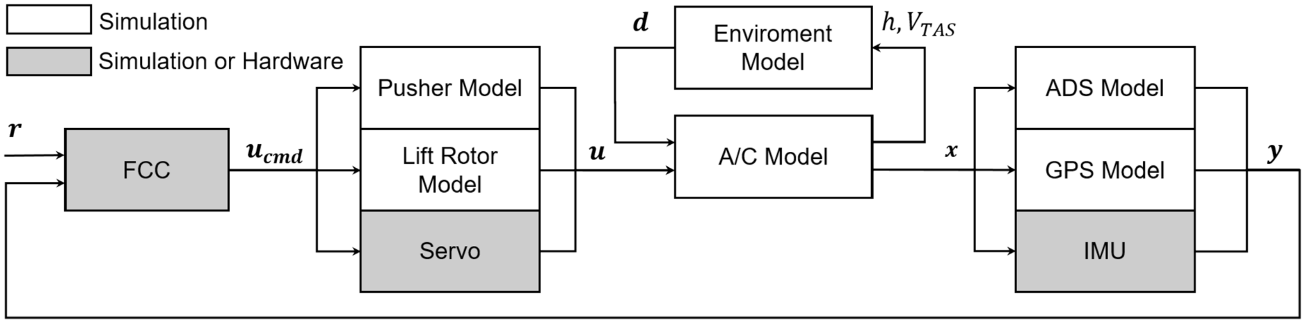

Fig. 9 Simulation model structure and hardware implementation for HIL testing

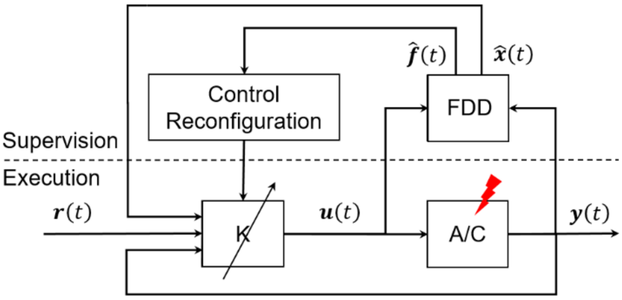

Fig. 10 Fault tolerant control loop with active FDD and control reconfiguration

for a wide range of operation points and can naturally be combined with a method for control allocation (CA) to distribute virtual control inputs among redundant actuators. An optimization-based CA approach has been implemented, which redistributes control inputs after a fault has been detected based on a Quadratic Programming routine [17]. This has the advantage that the baseline controller needs not be reconfigured. By choosing appropriate weighting functions, it is also possible to prioritize among redundant inputs in a no-fault scenario. The outer control loop consists of a stabilizing proportional integral controller to counter the effects of imperfect modelling accuracy and a first order reference model, generating desired aircraft responses from pilot inputs.

3.3 Simulation results

For the simulation results presented here, the longitudinal nonlinear UAV simulation model has been trimmed and linearized at an operation point of height h = 50 m and true airspeed $V_{TAS} = 30$ m/s.

The longitudinal states are $x(t) = [\,u\ \ w\ \ q\ \theta]^T$ with the horizontal and vertical velocity components (u, w), the pitch rate (q) and the Euler pitch angle (θ). The control inputs are $u(t) = [\,\delta_e\ \ \delta_t\ \ \omega_f\ \omega_b]^T$ with the elevator (δ_e), pusher throttle (δ_t) and the front and back lift rotor's angular velocity (ω_f, ω_b). The controlled variable is the pitch rate (q). The control law is executed at 100 Hz. Two simulation scenarios are presented. For both scenarios. a doublet to the pitch rate with an amplitude of ±40 deg/s is commanded. The step inputs are pre-filtered by a second order reference model with a damping ratio of $D = 0.85$ and an undamped eigenfrequency of $\omega_0 = 2$ rad/s. The resulting reference commands are fed to the inversion based control law and control allocation module. The weighting function for the control inputs has been chosen so that both the elevators and lift rotors are utilized as longitudinal control inputs.

In the first scenario, no fault in the actuators is simulated. Here, the differences between a software-in-the-loop (SIL) simulation, where the UAV dynamics, flight control law and the actuator model developed in Sect. 3.2 are run on the real-time target machine, and a HIL simulation including the real servo, flight controller and IMU gyros should be investigated. Figure 11 shows the resulting pitch rates q(t) and actuator states. The dead band behavior of the real servo is visible and due to the noisy potentiometer measurement from the A/D converters, which are fed back to the CA module, the allocated angular rates of the (simulated) front and back lift rotors are also significantly less smooth and actuated differently than in the SIL simulation. Furthermore, in the HIL setup, the measured pitch rate is disturbed by sensor noise effects from the real IMU. Nevertheless, there is almost no difference in the closed-loop performance. Further, because of the different servo input the CA module actuates the back lift rotors in a slightly different manner.

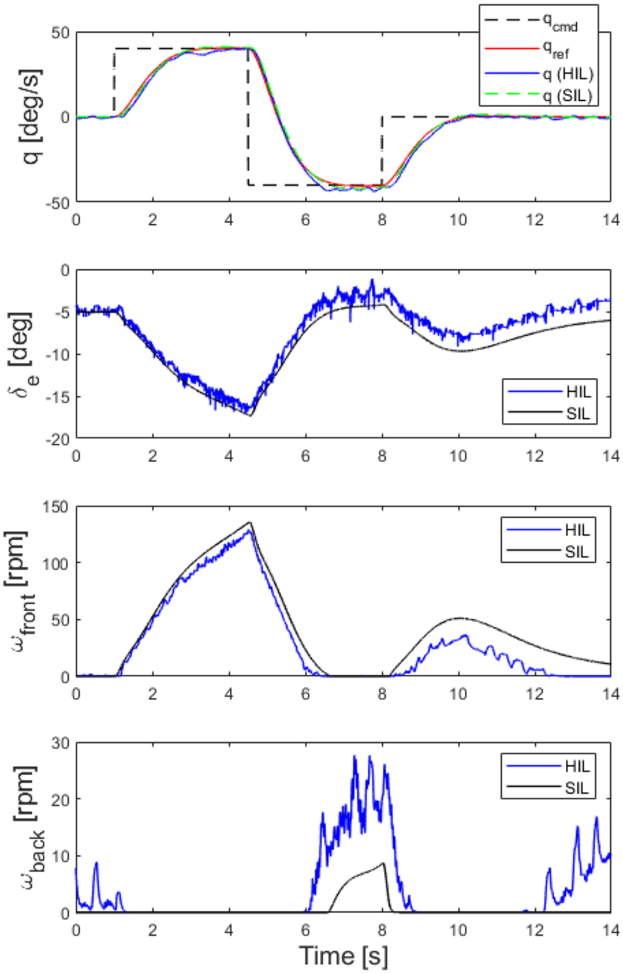

Fig. 11 Comparison of controlled variable $q(t)$ and control inputs $\delta(t), \omega(t)$ for a SIL and HIL simulation

In the second scenario, an elevator jamming is simulated after 1.5 s and the results are compared to the fault-free simulation. Here, only HIL simulation results

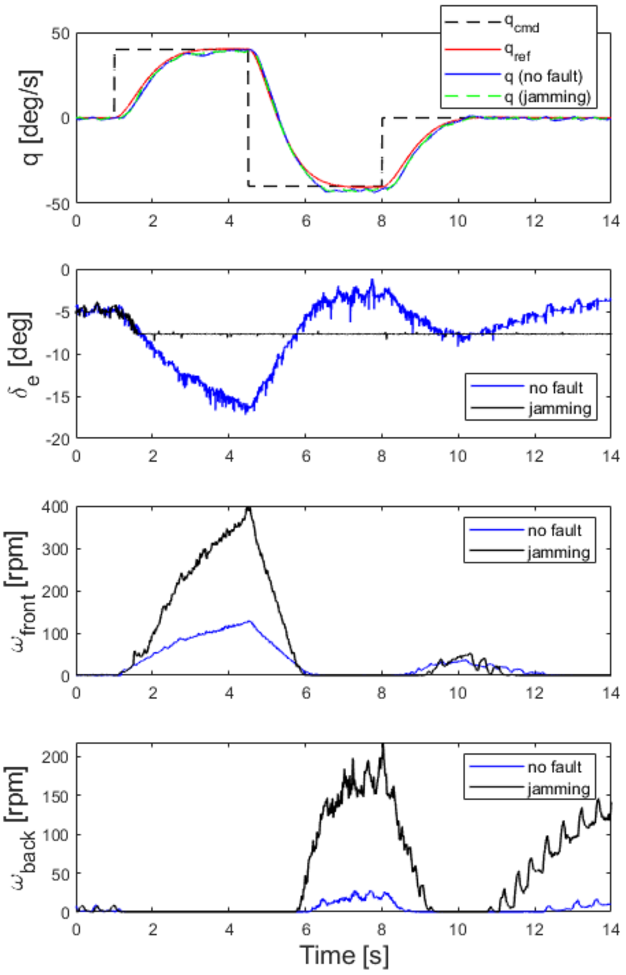

Fig. 12 HIL simulation comparison of controlled variable q(t) and control inputs δ(t), ω(t) for an elevator jamming fault occurrence after 1.5 s

integrating the real subsystems are presented. For FDD purposes, an ideal fault detection is assumed so that the fault information is directly forwarded to the CA module for reconfiguration of the remaining healthy actuators. Figure 12 compares the resulting pitch rate and control inputs. Despite this severe fault, the controller is able reallocate the virtual control input to the pitch axis so that the resulting closed-loop tracking performance remains unaltered. To compensate for the loss of elevator control, the front and back lift rotors need to generate more thrust. As the elevator is stuck at a negative angle relative to the trimmed value, the back rotors have to generate a constant pitch down moment in steady state.

4 Conclusion and future work

In this paper, the development of a HIL demonstrator for the evaluation of flight control laws for a hybrid UAV has been presented. The HIL setup includes a *Pixhawk 4* flight controller and a RC servo actuator, which have been integrated in a Simulink real-time simulation. Furthermore, a rotary 3-DOF-Stage has been designed to stimulate MEMS-based Inertial Measurement Units (IMUs) and feed the IMU's gyroscopic measurements to the real-time target computer. Furthermore, the parameter identification procedure for a generic hybrid UAV model and a real servo actuator integrated in the HIL setup have been introduced. Finally, HIL simulation results for a fault-tolerant control scheme against an actuator fault have been presented.

Future work will include the integration of an attitude estimation scheme and calibration routines into the HIL demonstrator. Furthermore, bias estimation algorithms for the IMU's gyros should be included. The presented fault-tolerant control setup will be extended to an already existing mathematical model of a real hybrid UAV flight demon-strator, which has been built at the Institute of Flight Systems and Automatic Control.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. SESAR Joint Undertaking, "European drones outlook study," Unlocking the value for Europe (2016)
2. Easy Access Rules for Unmanned Aircraft Systems (Regulation (EU) 2019/947 and Regulation (EU) 2019/945) (2019)
3. A. Varga, A. Hansson, and G. Puyou, (eds.): Optimization based clearance of flight control laws. In: A civil aircraft application. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). [Online]. Available: http://dx.doi.org/https://doi.org/10.1007/978-3-642-22627-4
4. Meier, L., Honegger, D., Pollefeys, M.: PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In: 2015 IEEE international conference on robotics and automation (ICRA), pp. 6235–6240 (2015)

5. Prochazka, F., Krüger, S., Ribnitzky, D.: Aerodynamic parameter identification of a hybrid unmanned aerial vehicle by using wind tunnel and free flight tests. In: Deutscher Luft- und Raumfahrtkongress (2020)
6. Wandarosanza, R., Trilaksono, B.R., Hidayat, E.: Hardware-In-the-Loop Simulation of UAV hexacopter for Chemical Hazard monitoring mission. In: 2016 6th International Conference on System Engineering and Technology (ICSET), pp. 189–193. Bandung, Indonesia (2016)
7. Pannocchi, L., Marinoni, M., Buttazzo, G.: Hardware-in-the-loop development framework for multi-vehicle autonomous systems. In 2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 17–22. Coimbra, Portugal (2017)
8. Pannocchi, L., Di Franco, C., Marinoni, M., Buttazzo, G.: Integrated framework for fast prototyping and testing of autonomous systems. J Intell Robot Syst **96**(2), 223–243 (2019). https://doi.org/10.1007/s10846-018-0969-3
9. Nguyen, K.D., Ha, C.: Development of hardware-in-the-loop simulation based on Gazebo and Pixhawk for unmanned aerial vehicles. JASS **19**(1), 238–249 (2018). https://doi.org/10.1007/s42405-018-0012-8
10. Mohamed Abdelkader Zahana, *MavLinkBridge.* [Online]. Available: https://github.com/mzahana/MavLinkBridge Accessed 25 Sept 2019
11. Valavanis, K.P., Vachtsevanos, G.J. (eds.): Handbook of unmanned aerial vehicles. Springer, Dordrecht (2015) [Online]. Available: http://dx.doi.org/https://doi.org/10.1007/978-90-481-9707-1
12. Ahmed, J., Bernstein, D.S.: Adaptive control of double-gimbal control-moment gyro with unbalanced rotor. J. Guid. Control. Dyn. **25**(1), 105–115 (2002). https://doi.org/10.2514/2.4855
13. Prochazka, F., Ritz, T., Eduardo, H.: Over-actuation analysis and fault-tolerant control of a hybrid unmanned aerial vehicle. In: 5th CEAS conference on guidance, navigation & control (EuroGNC), Milano, Italy (2019). https://eurognc19.polimi.it/wp-content/uploads/2019/12/0068_FI.pdf
14. Stevens, B.L., Lewis, F.L., Johnson, E.N.: Aircraft control and simulation: dynamics, controls design, and autonomous systems. Wiley, Hoboken (2015)
15. Wada, T., Ishikawa, M., Kitayoshi, R., Maruta, I., Sugie, T.: Practical modeling and system identification of R/C servo motors. In: 2009 IEEE International Conference on Control Applications, pp. 1378–1383. St. Petersburg, Russia (2009)
16. Holzapfel, F.: Nichtlineare adaptive Regelung eines unbemannten Fluggerätes. Technische Universität München (2004)
17. Härkegård, O.: Backstepping and control allocation with applications to flight control. Zugl.: Linköping, Univ., Diss., (2003); Linköping: Univ, (2003)