Chapter 4
Real-Time Hybrid Simulation of an Unmanned Aerial Vehicle

Markus J. Hochrainer and Peter Schattovich

Abstract This work presents a real-time hybrid simulation for the analysis and optimization of the electronic control unit of a quadcopter. Therefore, the existing physical microcontroller hardware is coupled to a real-time computer model used to simulate the flight. This requires the numerical solution of nonlinear equations of motion including coordinate transformations. Knowing the flight dynamics, the simulated measurements of a virtual inertial measurement unit are determined and fed back to the physical flight control unit to calculate the required actuator response for a desired behavior, thereby closing the control loop. This type of hybrid simulation is currently the most efficient method to obtain a desired system performance before carrying out experimental tests with the entire physical system. Furthermore, a virtual reality module for real-time flight visualization was developed for better analysis of different flight scenarios. Since all results show excellent agreement with real flight testing, the work confirms the efficiency of the proposed system. During the tests it was e.g. possible to determine the effect of different inertia measurement unit sensors with specific noise characteristics on the overall flight dynamics and consequently, find the reason for rarely occurring engine failures. In addition, the project shows that complex real-time hybrid simulations on industrial level are possible even with low investment costs.

Keywords Real-time hybrid testing • Hardware in the loop • Real-time simulation • Quadcopter • Direction estimation

4.1 Introduction

Experimental techniques have developed significantly in the last decades, with a distinct focus on hybrid experimental-computational techniques. This process has been driven by a steady progress in model based design as well as physical modeling techniques together with powerful automatic code generation tools, which can be configured to generate fast C and C++ code for use on embedded processors, target rapid prototyping boards, microprocessors or real-time PC based systems. On the other side, there is a tremendous advance of cheap and very powerful embedded systems already including analog and digital interfaces and thus computing power is easily available for almost any level of real-time hardware. This reduces the additional costs for real-time hybrid simulation (RTHS) substantially, and, furthermore, the simulation model stays almost unaffected of the target hardware. As a consequence, hybrid simulation techniques have attracted increased research attention and they can be found in almost any field of experimental testing. This development is further supported by the fact, that modelling and simulation of multi-domain component oriented physical systems is supported by several modeling languages. In this context, all individual component models are based on physical connections, and the level of detail of the simulation can be changed just by exchanging simple components models to more complex ones. A hierarchy of different component models is often readily available from different component libraries.

Depending on the scientific discipline, the coupling of numerical and experimental techniques is known as (real-time) hybrid simulation, hybrid dynamic substructuring or hardware in the loop (HIL) testing. Although these new techniques have led to significant savings, faster product development and reduced design uncertainties, full scale experimental testing of the entire system cannot be eliminated completely. However, modern testing methods are even more demanding for the scientist due to the multidisciplinary nature of work integrating numerical and experimental methods. It requires advanced knowledge in the fields of modeling and simulation, real-time integration, model-order reduction, scalable numerical simulations, measurement and signal processing. Furthermore, since the coupled systems generally result in a closed loop structure, profound understanding of control theory, sensors and actuators is essential, see e.g. [1, 2]. Nevertheless, the research work of the last decade has resulted in a much deeper understanding of hybrid simulation and, consequently, many initial problems

---

M.J. Hochrainer (✉) • P. Schattovich
University of Applied Sciences, Johannes Gutenberg-Strasse 3, 2700, Wiener Neustadt, Austria
e-mail: markus.hochrainer@fhwn.ac.at

M.S. Allen et al. (eds.), *Dynamics of Coupled Structures, Volume 4*, Conference Proceedings of the Society
for Experimental Mechanics Series, DOI 10.1007/978-3-319-54930-9_4

have been overcome leading to wide spread use of RTHS. From a historical perspective several hybrid techniques have been developed in the field experimental mechanics often with a distinct focus on earthquake engineering. Conventional experimental testing was either quasi-static or dynamic (shake table testing). Both types can be categorized as open loop, as all loads applied to the device under test (DUT) are predetermined, and feedback from the DUT is not required in the experiment. Quasi-static tests are based on the application of slowly varying loads to determine the nonlinear behavior of structural members, whereas shake table tests are required whenever a component shows a loading rate dependence. Since the loading capacity of quasi static test rigs is generally significantly higher than that of dynamic ones, quasi-static testing allows larger DUTs to be studied. If the loading is depending on the DUT's response (forces or displacements), the method is denoted hybrid simulation, since it combines both, numerical simulation of a substructure and physical testing. Typically, the behavior of the components tested physically is fairly complex and difficult to predict, whereas the numerical model is rather simple permitting highly reliable estimates. Hybrid simulations can be pseudo-dynamic or dynamic. Pseudo-dynamic hybrid simulations are typically displacement controlled with a feedback of measured physical forces to a numerical model which accounts for all dynamic effects like masses or (visco-) elastic components. The fact that all equipment necessary for hybrid simulation is available in a dynamics laboratory, is one of the salient feature of pseudo-dynamic hybrid simulation. If the DUT has dynamic properties which significantly influence the overall behavior, the hybrid simulation must be performed in real-time. This often requires complex control mechanisms and places high demands on equipment and simulation model (real-time simulation). The importance of hybrid simulation is strongly associated with experimental testing, which is, at the moment, the only reliable method to confirm, develop and improve numerical simulation models. Although well established in the field of civil engineering, RTHS has become very attractive for many other disciplines because it permits reliable testing of individual components while taking into account the complex interaction with the overall system. In automotive and aeronautic industry real-time hybrid simulation/testing is well established but known as hardware in the loop simulation/testing. The approval of components and modules is responsible for an ever increasing demand for this kind of testing in recent years. Depending on the actual application, the requirements and challenges of real-time hybrid simulations are very different. When testing mechanical systems quasi-static methods are often appropriate for nonlinear members of complex structures, however, in most other applications this type of time scaling is not possible and accordingly real-time hybrid testing is generally applied. This is particularly true if a system's electronic control unit (ECU) is tested because potential data transmission, digital interfaces, timing aspects, the control loop as well as possible analog filtering is hardly possible with quasi-static testing.

Apparently, the RTHS or HIL philosophy allows reliable component testing without any risk. This is a significant advantage in the development of aircrafts, because each test flight has the potential threat of aircraft crashes. However, as in any other discipline the computer models used for HIL-testing are a simplification of the real physical structure and therefore experiments and tests with the real system cannot be replaced completely yet. On the other side, the HIL philosophy enables investigations which are hardly possible with traditional methodologies, e.g. the repeatable injection of fault signals or the temporal/permanent failure of sensors if an ECU is selected as DUT. Therefore HIL currently seems to be the most versatile and efficient method for obtaining a desired system performance before performing experimental test with the entire physical system, see e.g. [3, 4] for aircraft applications. In the project presented, the copter was developed without proper simulation and since the performance did not meet the expectations, RTHS was used to analyze the system before redesigning it.

4.2 Concept

The primary prerequisite for any HIL testing is a complete separation of all physical components from the simulated numerical model. In case of the quadcopter the physical components include the control unit, power amplifier, BLDC engines and remote control. In the current project, the coupling between physical and simulated model is based on simulated sensor values and estimated aerodynamic lifting forces (rotor thrusts). The ECU position controller processes simulated accelerations and angular velocities generated by the real-time simulation of the quadcopter flight dynamics. The simulation model, on the other hand, receives the current engine speeds for individual rotor thrust determination. Thus, the HIL control loop is closed as illustrated in Fig. 4.1. The host-PC is primarily used for the configuration and control (HIL control/configuration) of the hybrid testing. However, since sufficient computational resources are available the host PC is also used for the online visualization (VR-model/data analysis) and analysis of the experimental results. Via a standard Ethernet interface the host PC communicates with the target PC, which performs all real-time simulations (flight dynamics) and derives the simulated measurements (IMU simulation). It is connected to the ECU, the physical section of the experiment, by a digital UART interface (RS232). The central component of the ECU is an embedded system based on a 16-bit microcontroller of type Microchip PIC 24FJ256GB106 which is already set up to perform both, the calculation of

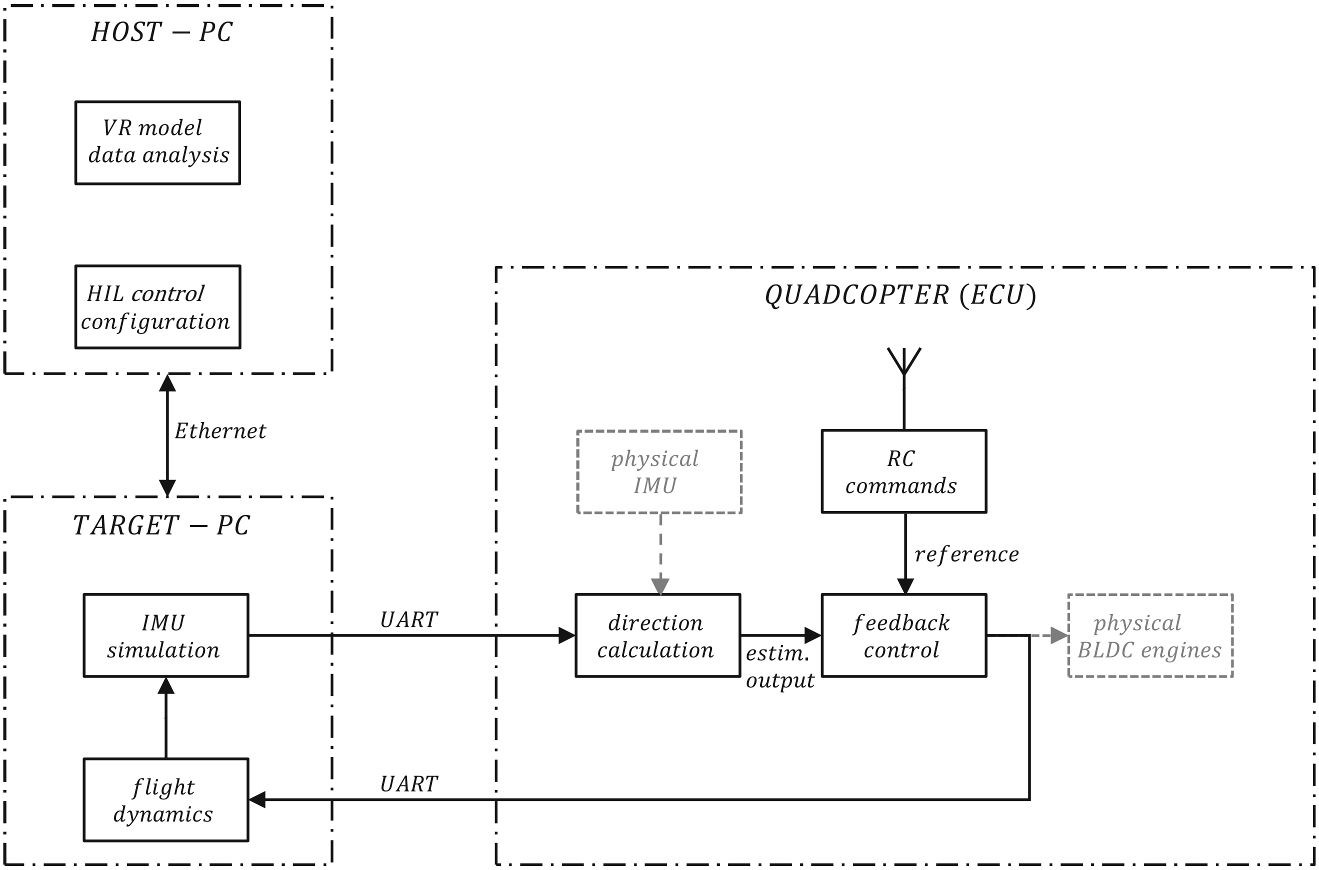

Fig. 4.1 Schematics of the HIL configuration applied

the direction in space (direction calculation) and the stabilization of the inherently instable quadcopter (feedback control). In addition the ECU is responsible for the communication with the RC-unit (RC commands) and the additional UART communication in the HIL setup.

4.3 System Modelling

When compared to helicopters, one outstanding advantage of quadcopters is the simple and robust design of entire drive line. The only requirement is that all rigid rotors can operate at different speeds. Although the number of rotors can vary, it is typically increased in pairs thereby generating redundancy and a higher level of reliability, e.g. in case of hexa- or octocopters. From all multirotor aircrafts available, the quadcopter is most popular and therefore a quadcopter setup was chosen for the original project. When compared to conventional helicopters, quadcopters have no moving parts, no cyclically adaption of the angle of attack, no governor and no need for a tail rotor. However, this is at the price of four engines, typically brushless DC (BLDC) drives, which are alternately rotating in opposite direction. When spinning at the same angular velocity, all reaction torques fully compensate, and the resulting lifting force is adjusted by the engine speed. Yaw (without a change of the cumulative thrust) results from a symmetric thrust-offset between the counter rotating blade pairs. Roll and pitch are adjusted by inversely changing the thrust of two opposing rotors, while keeping the total reaction moment and lifting force constant. The increasing popularity of quadcopters is also due to their simplicity with respect to control: Any complex flight maneuver results from superimposing the rotor speed adaptions of the corresponding basic flight operations (yaw, pitch, roll and altitude adjustment). It is important to recognize that the rotor thrusts always point in the direction of the local z-axis. Since the drive engines are directly connected to fixed pitch rotor blades, there are no moving parts, and consequently, the copter motion must be controlled by tilting the entire aircraft. Due to their geometric design, most quadcopter are inherently unstable unless the center of mass is located very low with respect to the distance of $2l$ between a pair of rotors. Therefore,

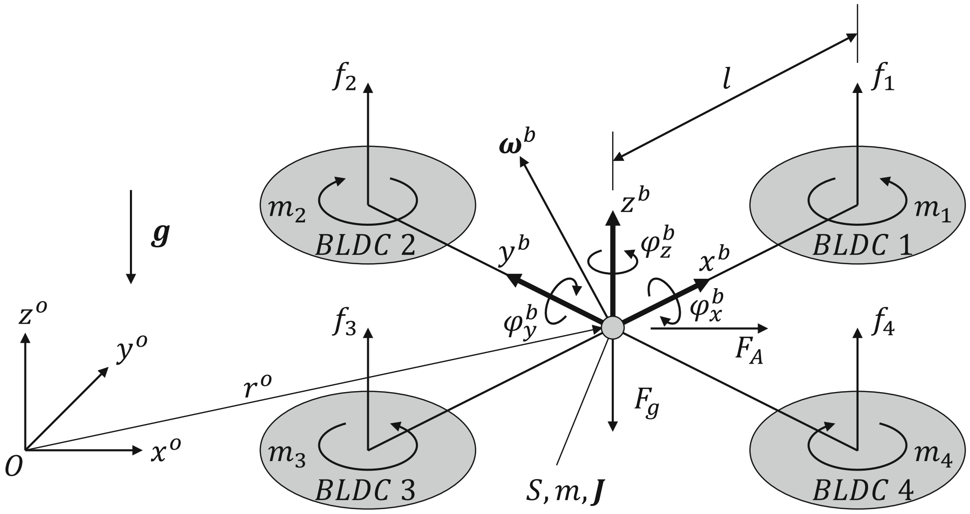

Fig. 4.2 Free body diagram of the quadcopter

a permanent feedback control stabilization is essential which is generally based on local IMU measurements. This inherit instability, on the other side, is the reason for the comparably fast flight dynamics. However, with rising copter and blade size, the rotor's moment of inertia is growing fast and rotor speed corrections take significantly longer, which negatively impacts control.

To obtain a mathematical model of the quadcopter motion, it is assumed rigid and fully symmetric with mass m, moments of inertia collected in the tensor $\boldsymbol{J}$ and the center of mass S. The forces acting on the rigid body are the four thrusts $\boldsymbol{F}_i$, $i = 1\,..\,4$, its corresponding reaction torques $\boldsymbol{M}_i$, the aerodynamic drag force $\boldsymbol{F}_A$ and the gravitational force $\boldsymbol{F}_g = m\boldsymbol{g}$, with $\boldsymbol{g}$ denoting the constant of gravity, see Fig. 4.2 for the free body diagram.

Following the kinematics of rigid bodies, newton's law of inertia for moving reference frames renders an equation of motion in the moving reference frame (local coordinates superscript b)

$$m\dot{\boldsymbol{v}}^b + m\boldsymbol{\omega}^b \times \boldsymbol{v}^b = \sum \boldsymbol{F}_i^b + \boldsymbol{F}_g^b + \boldsymbol{F}_A^b, \tag{4.1}$$

where $\boldsymbol{\omega}^b = \left(\dot{\varphi}_x^b,\ \dot{\varphi}_y^b,\ \dot{\varphi}_z^b\right)^T$ denotes the angular velocity vector. The conservation of angular momentum renders a nonlinear differential vector equation,

$$\boldsymbol{J}^b\dot{\boldsymbol{\omega}}^b + \boldsymbol{\omega}^b \times \boldsymbol{J}^b\boldsymbol{\omega}^b = \sum \boldsymbol{M}_i^b. \tag{4.2}$$

Defining the local coordinates by the principal axes of inertia, the inertia tensor becomes diagonal, $\boldsymbol{J} = diag(J_x,\ J_y,\ J_z)$ and three nonlinear differential equations of motion with constant coefficient, known as Euler's gyroscopic equations are obtained, see e.g. [5, p. 420, 6]. From the equations of motion 1–2 the quadcopter dynamics is completely determined and for known individual engine thrust forces $\boldsymbol{F}_i^b = (0, 0, f_i)^T$ together with the resulting moments $\boldsymbol{M}_i^b = \boldsymbol{r}_i^b \times \boldsymbol{F}_i^b + (0, 0, m_i)^T$ the differential equations can be solved.

Apparently, the flight position must be given in absolute coordinates (superscript o) $O = (x^o, y^o, z^o)$, see Fig. 4.2. All rotations are based on the proper Euler angles $\boldsymbol{e}^o = (\phi,\ \theta, \psi)^T$, which describe elemental rotations in a defined order: first the rotation ψ about the z-axis, then the rotation θ about the already rotated y-axis, and finally another rotation ϕ about the x-axis, see e.g. [7]. The transformation of vector quantities between different frames of reference is carried out by a matrix multiplication with the orthogonal rotation matrix $\boldsymbol{R}(\boldsymbol{e})$, see again [7]. Since the equations of motion are given in local coordinates (moving frame of reference) the time dependent relation between the angular velocity vector $\boldsymbol{\omega}^b$ and the Euler angles $\boldsymbol{e}^o$

$$\dot{\boldsymbol{e}}^o = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{pmatrix} \boldsymbol{\omega}^b = \boldsymbol{\Gamma}\,(\boldsymbol{e})\,\boldsymbol{\omega}^b \tag{4.3}$$

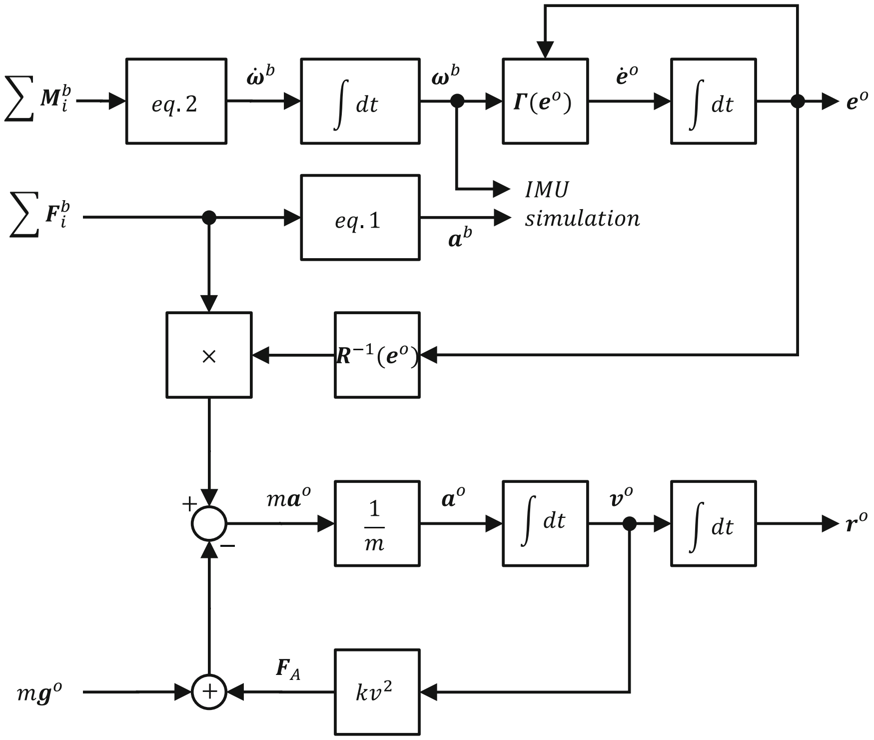

Fig. 4.3 Signal flow of real-time simulation

is required for the numerical integration of the equations of motion. For feedback stabilization the vast majority of quadcopters use IMU (gyro and accelerometer) measurements, which are predominantly processed by complementary filtering to determine the actual copter orientation. However, this measurement principle is only valid for stationary flight conditions, and this is the reason why it is essential to simulate the aerodynamic drag force $F_A^0 = v^o\,|v^o|\,k$, with k describing the drag constant. An overview of the calculations necessary (signal flow) to obtain the actual position in global coordinates is given in Fig. 4.3. For simulated IMU measurements the determination of ω^b is straightforward, the calculation of a^b requires the correction of the centrifugal forces, and both vector quantities must, of course, be scaled according to the datasheet's sensor hardware specification, see [8].

4.4 Implementation and Setup

Having derived the equations of motion and the signal flow diagram for solving them numerically, refer Fig. 4.3, it is straightforward to implement the numerical model in a signal flow based simulation environment. The numerical integration is performed using an explicit Runge-Kutta method of order four (RK4) with a constant integration step size (fundamental sample time) of $T_S = 5ms$. Within modern simulation environments the model can be compiled and transferred to the target computer hardware which is connected to the physical ECU system. Besides standard interfaces (Ethernet, RS232) there are no special hardware requirements in the current project, because the real-time ability is provided by the operating system. When the simulation is running on the target PC, the flight trajectories are determined from the four rotor speeds which are periodically transmitted from the physical ECU. Once the flight trajectory is known, the IMU sensor values can be derived directly from the local acceleration and angular velocity. Because the simulated sensor output represents a perfect measurement, it must be degraded by superposition of various sensor errors e.g. sensor noise, offset, temperature drift, nonlinearities, before transmitting it to the physical ECU for further processing. In the current project it was essential to keep the existing ECU firmware almost unchanged. However, minor modifications of the communication routines were required for data exchange with the real-time target PC. Before starting RTHS testing, it is, however, vital to identify all mechanical system parameter (mass, inertia tensor, geometric dimensions), as well as the drive characteristics at a high level of confidence. Consequently, all not directly measureable parameter were determined by experimental testing: The inertia

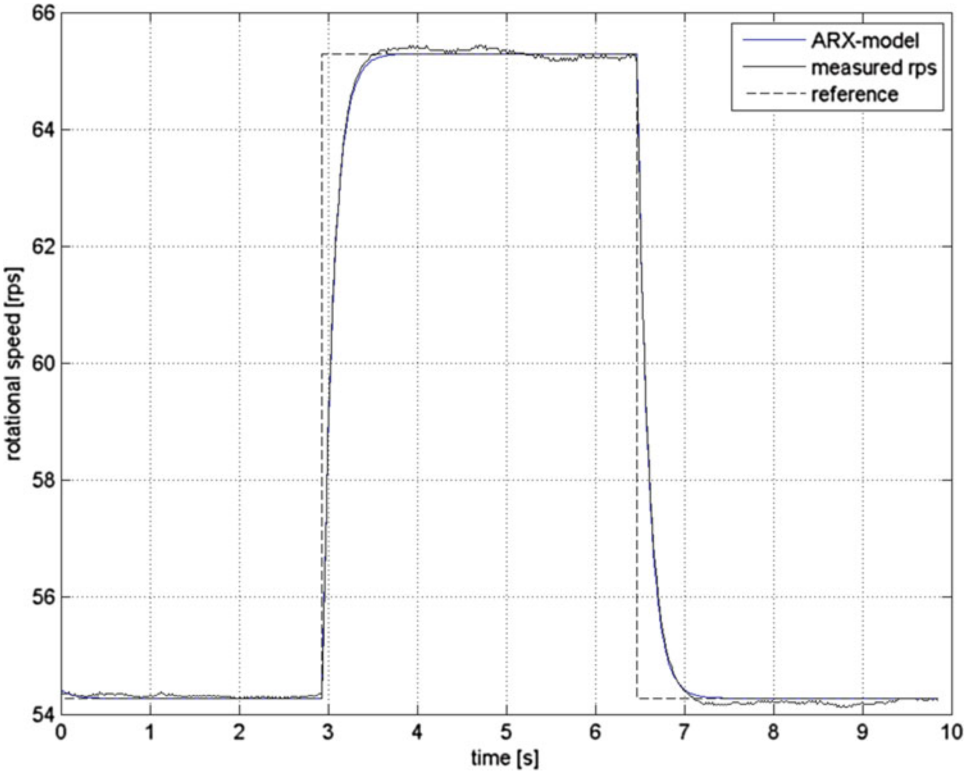

Fig. 4.4 Step response of the entire drive line

tensor by vibration tests, the speed-duty-cycle characteristics of the BLDC engine by rpm measurements and the quadcopter yaw using a pendulum type set-up. All experiments were repeated several times, with the desired parameter being identified using least squares methods. A rather important component of the entire identification process was the determination of the overall drive dynamics comprising of the power amplifier and the BLDC engine with rigidly attached rotors in the relevant rpm range, see Fig. 4.4. The overall drive line dynamics is obtained from an ARX model of first order, again, by least squares identification.

4.5 Experimental Results

The developed, calibrated and sufficiently tested HIL experiment allows to simulate, validate and analyze the dynamic performance of the quadcopter in any possible flight situation. Nevertheless, all experiments have shown, that the assessment of the system behavior is hardly possible using standard graphical interfaces like time-plots, time histories or numerical displays. Therefore, a flight animation was developed by linking 3D graphics objects of a virtual reality environment to the numerical simulation results. Only this way it was possible to develop a fundamental understanding of the system behavior, since a realistic visualization of the flight is essential if an operator is controlling the system using the standard RC control, see Fig. 4.5. Even in case of automated testing with predefined flight maneuvers, the visualization is crucial to correctly interpret the dynamic behavior.

Using the presented RTHS, the focus of the research and development work can again be put on the improvement of the IMU sensor measurements within the ECU firmware. Accordingly, the analysis concentrated on real physical flight situations which initially triggered the HIL testing: little stability margins during hovering, irregular crashes due to suspected engine shut down as well as firmware bugs in the processing of measured data. The inspection of the HIL flight stability has confirmed the shortcomings of the current ECU with respect to limit cycle vibrations around all axes during stationary hover flight. This deficient performance became even worse with increased simulated IMU sensor noise, see Fig. 4.6. The HIL experiment has proven undoubtedly, that this behavior was due to numerical effects when calculating the copter direction.

Fig. 4.5 Real-time visualization of a quadcopter flight at the university campus in the virtual reality environment, the copter is guided by a remote control

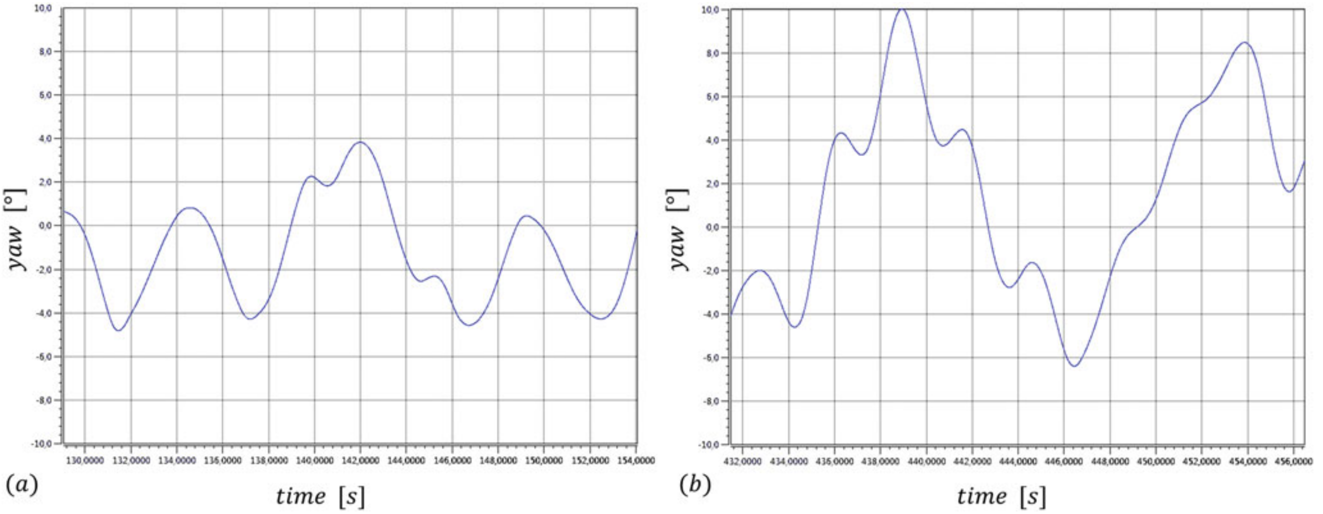

Fig. 4.6 Yaw vibrations during hovering for different IMU sensor configurations (**a**) low sensor noise (**b**) increased sensor noise

It is essential to note that the limit cycle behavior does not occur in a traditional simulation (model in the loop without HIL coupling). However, the effectiveness and reliability of the proposed set-up is confirmed, since undesired vibrations were clearly observed during real flights.

The hybrid simulation offers the possibility to rigidly attach the quadcopter in the laboratory while energizing the BLDC engines to activate all rotors during the experiment. Only this way the occasional engine stops could be reproduced, and systematic error tracking revealed an emergency shutdown triggered by the overcurrent protection module of the power amplifier unit. As expected, this behavior was directly dependent on the simulated sensor noise level and thus, the test has confirmed that sensor noise reduction by improved filtering is essential, because even under normal flight conditions the IMU is exposed to very high vibration and EMC levels. Similarly, the work has demonstrated that the firmware code segment for evaluating the angular velocity behaves improper when exceeding a critical gyro limit. Consequently, the implementation of either the numerical integration or the complementary filter is incorrect. Finally, the magnetic field sensor used to calculate the local yaw angle was tested, because it has always shown a moderate dependence on rotations about the other axes. So far, this effect has been attributed to interfering magnetic fields in the laboratory or EMC engine noise. The simulation of a perfect earth's magnetic field together with an ideal magnetic sensor, however, has revealed that the effect must again be attributed to the ECU data processing.

4.6 Conclusions

This work presents a real-time hybrid simulation for the analysis and optimization of the electronic control unit of a quadcopter, which was initially performed to analyze and understand the undesired dynamic flight performance. Hence, the existing physical ECU including the embedded system and firmware, the power amplifier and the BLDC engines are coupled to a real-time computer model used to simulate the flight. In summary, the HIL testing compares very well with real flight tests and the detected insufficient flight performance could be attributed clearly to shortcomings of the current firmware. A revision of the current sensor data processing is required with respect to noise and disturbance attenuation. With RTHS accurate testing of all firmware improvements is possible without real flight experiments. The hybrid simulation indicates, that instead of using improved IMU sensor units, a firmware tuning should be sufficient to achieve the desired flight behavior. The work confirms, that proper RHTS allows all flight maneuvers to be simulated and optimized at a very realistic level. Consequently, the HIL methodology opens opportunities and enables developments, which are hardly possible with traditional methods, or only with a significantly greater effort.

References

1. Saouma, V., Sivaselvan, M. (eds.): Hybrid Simulation: Theory, Implementation and Applications. Taylor & Francis Ltd (2008), London, UK
2. Bursi, O. S., Wagg, D.: Modern Testing Techniques for Structural Systems, Dynamics and Control. CISM International Centre for Mechanical Sciences, vol. 502, Springer, New York, Vienna (2008)
3. Gans, N.R., Dixon, W.E., Lind, R., Kurdila, A.: A hardware in the loop simulation platform for vision-based control of unmanned air vehicles. Mechatronics. **19**(7), 1043–1056 (2009)
4. Karpenko, M., Sepehri, N.: Hardware-in-the-loop simulator for research on fault tolerant control of electrohydraulic actuators in a flight control application. Mechatronics. **19**(7), 1067–1077 (2009)
5. Ziegler, F.: Mechanics of Solids and Fluids, 2nd reprint of second edition. Springer, New York, Vienna (1998)
6. Li, Z., Murray, R.M., Sastry, S.S.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton (1994)
7. Fu, K.S, Lee, C.S.G., Gonzales, G.C.: Robotics, Control, Sensing, Vision and Intelligence. McGraw-Hill International Editions (1987)
8. Schattovich, P.: Hardware in the Loop Studies of an Unmanned Aircraft, in German, Master Thesis, University of Applied Sciences, Wiener Neustadt, Austria (2016)