Hindawi International Journal of Aerospace Engineering Volume 2020, Article ID 1360702, 10 pages https://doi.org/10.1155/2020/1360702



Research Article

Controller for a Low-Altitude Fixed-Wing UAV on an Embedded System to Assess Specific Environmental Conditions

Johannes von Eichel-Streiber, Christoph Weber, Jesús Rodrigo-Comino, and Jens Altenburg

¹Institute for Innovation, Transfer and Consulting, 55411 Bingen am Rhein, Germany

Correspondence should be addressed to Christoph Weber; uav@th-bingen.de

Received 19 February 2020; Accepted 22 April 2020; Published 16 June 2020

Academic Editor: Zongyu Zuo

Copyright © 2020 Johannes von Eichel-Streiber et al. 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.

The use of an appropriate sensor on an unmanned aerial vehicle (UAV) is vital to assess specific environmental conditions successfully. In addition, technicians and scientists also appreciate a platform to carry the sensors with some advantages such as the low costs or easy pilot management. However, extra requirements like a low-altitude flight are necessary for special applications such as plant density or rice yield. A rotary UAV matches this requirement, but the flight endurance is too short for large areas. Therefore, in this article, a fixed-wing UAV is used, which is more appropriate because of its longer flight endurance. It is necessary to develop an own controller system to use special sensors such as Lidar or Radar on the platform as a multifunctionality system. Thereby, these sensors are used to generate a digital elevation model and also as a collision avoidance sensor at the same time. To achieve this goal, a small UAV was equipped with a hardware platform including a microcontroller and sensors. After testing the system and simulation, the controller was converted into program code to implement it on the microcontroller. After that, several real flights were performed to validate the controller and sensors. We demonstrated that the system is able to work and match the high requirements for future research.

1. Introduction

The selection and use of specific sensors devoted to assessing different environmental issues for remote sensing applications are essential for scientists and technicians [1, 2]. However, traditional platforms such as satellites or manned aircraft have some recognized disadvantages to including specific sensors. For instance, satellites often do not meet the desired resolution and can be very expensive [3]. On the other hand, a manned aircraft needs an expensive pilot and produces other costs, e.g., aircraft fuel [4, 5]. Besides these two platforms, there are some more that recently have been considered for the scientific community, for example, helicopter or terrestrial [6, p. 4]. Heipke et al. [7, p. 34]

exposed that there are different platforms (satellite, aircraft, UAS, and terrestrial) divided into different specifications such as costs, operation risk, or resolution. The authors pointed out that the unmanned aircraft systems (UAS) can be considered the platform with the lowest costs. Recently, the small UAS is giving the opportunity to achieve new applications and assess different scenarios with successful results in applied scientific works [8–10]. They are now often used for standard applications such as topographical surveying, inspection flights, mapping, and surveillance [11, p. 119]. Besides these, there are some special applications which need a low-altitude flight such as estimations of plant density [12], rice yield [13], mineral exploration [14], and evaluation of infrastructures as well as meteorological hazard risks [15–

²Department 2-Engineering, Computer Science and Economics, University of Applied Sciences, 55411 Bingen am Rhein, Germany

³Department of Physical Geography, University of Trier, 54296 Trier, Germany

⁴Soil Erosion and Degradation Research Group, Department of Geography, Valencia University, Blasco Ibàñez, 28, 46010 Valencia, Spain

17]. All of these applications used to have a rotary UAS which include a multirotor by means of a quadcopter and hexacopter that are able to fly in a low altitude from 3 to 40 m or up to 300 m.

Accordingly, it is important to remark that rotary UAS matches the requirements for such a low-altitude flight. However, the endurance is mostly less than 30 minutes because they have no wings for uplift; therefore, they are only useful for small areas. In research where larger areas must be studied, fixed-wing UAS with a flight endurance of about 60 min are necessary [7, p. 44], [18]. The controller for a fixed-wing aircraft is well known and described in detail in the recent literature [19, 20], especially for UAS controlling techniques [21-23]. Besides the literature, there are some commercial autopilots on the market (e.g., Pixhawk 4 and Paparazzi [24]). However, in order to use special sensors, it is necessary to develop its own autopilot. Sensors such as special Lidar [25, 26] or Radar [27, 28] can be easily implemented with multifunctionality systems. Lidar is commonly used in order to generate digital elevation models, detect flow patterns, or assess connectivity [29-31] or even to realize a collision avoid system [32]. On the other hand, for a weatherindependent system, Radar is more appropriate [7, p. 53] [33, p. 204].

This is possible when a flight controller is developed for the fixed-wing and rotary UAS. However, nowadays, there is a lack of this development; fortunately, increasing interest to design and apply this to assess environmental issues among others is occurring. Therefore, the main aim of this research is to develop an autopilot with special requirements by means of low-altitude flights included in a small fixedwing UAS equipped with a hardware platform characterized by a microcontroller to run the developed controller program. To achieve this goal, we paid attention to the following: (i) to develop a commercial small fixed-wing unmanned aerial vehicle (UAV) which was equipped with a hardware platform including a microcontroller and several sensors, (ii) to test the UAS including a ground control station and check the integrity of the sensor data to supply the controller, (iii) to analyse the UAV process and develop a simulation including a controller, (iv) to develop a program code that allows to run the simulated controller on the microcontroller, and (v) to test the cascade control loops step by step and discuss the results.

2. Unmanned Aircraft System

To test the controller, we use a commercial small fixed-wing UAV "Yuki-Yuki Fly" (Figure 1). UAV means the aircraft itself, and UAS is defined as the whole system including a ground station for control. The UAV is characterized to have a wingspan of about 1.6 m and a weight of about 1.6 kg, including the battery. The advantages of this UAV are enough space to mount the newly developed system and the availability for a low price on the market (about €200). Furthermore, this UAV is built with expanded polypropylene (EPP) which is easy to be repaired or changed after a crash or accident.



FIGURE 1: Fixed-wing UAV with modified parts. Pitot tube (a), PCB stack inside (b), and telemetry antenna (c).

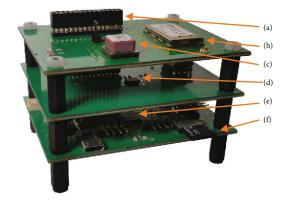


FIGURE 2: Printed circuit boards are stacked and interconnected with a 30-pin header. 30-pin header (a), telemetry (b), GNSS (c), IMU (d), μ C (e), and SD card (f).

In this research, we modified the aircraft by implementing in the electronic system a small platform to attach the printed circuit boards (PCB). Three PCB are stacked and connected via a 30-pin header to each other. The size of the complete stack is $8\times6\times5$ cm. With this concept, it is possible to replace a PCB in a case of hardware updates or failures. Figure 2 shows the PCB stack.

The bottom PCB contains the microcontroller (μ C) which is used for the controlling algorithm, sensor data processing, and telemetry connection. The μ C is an ARM 32-bit Cortex-M4 CPU from STMicroelectronics with 180 MHz CPU frequency, 128 kB SRAM, and 512 kB flash memory. The program code was written in the programming language "C", and no additional operating system is used. The program implements all the necessary real-time criteria for the system. This means if the program code takes too much time in case of a failure, the system will restart immediately. The whole restart takes about 60 ms. At this time, the aircraft keeps the last settings and flies further this way. To control the aircraft, this PCB has connectors on the front and on the rear. These are used to connect the servos for setting the flaps, elevator, and rudder as well as the motor controller. Additionally, this PCB has a voltage down converter, which is necessary to supply the μ C and the sensors with 3.3 V or 5 V. Usually, an electronic-driven UAV has three to six lithiumion polymer (LiPo) battery cells in series. The abbreviations

for those configurations are 3s and 6s batteries. However, that means it has a voltage of about $12.6\,\mathrm{V}$ to $25.2\,\mathrm{V}$ and needs to be converted down. Last but not least, an SD card slot, which is located on the front, is included to read and write data. Examples are reading waypoints from the SD card and writing a log file of the flight.

The middle PCB in the stack contains an inertial measurement unit (IMU). This is considered the most important sensor and used for basic controlling. For mounting, it is essential to check the position of the IMU in relation to the UAV. The difference between the two needs to be corrected with an offset. The used sensor is a 9 degrees of freedom (DOF) IMU "Bosch BNO055". The major advantage of this sensor is the integrated sensor fusion functionality. It combines the raw sensor values from the accelerometer, gyroscope, and magnetometer into an orientation vector that contains the Euler angles: roll, pitch, and yaw. All three axes are limited and need to be taken into account in the program code: roll from -90° to +90°; pitch from +180° to -180°; and yaw from 0° to 360°. The data output rate is set to 33.3 Hz, which is connected to the μ C in the lower stack via a universal asynchronous receiver transmitter (UART) interface. After disclosing a bug in the communication via UART to the manufacturer, they recommended using the Inter-Integrated Circuit (I²C) interface because it will not be fixed. The bug triggers sometimes a bus overrun error. Because of the relatively low communication speed, the bug occurs only rarely. Therefore, we decided to work around this bug instead of implementing a whole new I²C interface which is highly time-consuming. With this solution, the IMU is fully integrated and not limited in its use.

The top PCB in the stack is assembled with a Global Navigation Satellite System (GNSS) sensor and the telemetry. In order to receive a clear GNSS signal, it is located on the top of the stack. The sensor has a built-in embedded antenna. Therefore, no extern GNSS antenna is necessary, which allows saving weight and building space. The sensor is a "Telit SE868" and can receive multiple GNSS services. The most common one is the Global Positioning System (GPS) from the United States [34]. Furthermore, it can receive Galileo signals from the European Union [34] and "Globalnaja nawigazionnaja sputnikowaja sistema"-Global Navigation Satellite System (GLONASS) from Russia [34]. Besides that, it receives satellite-based augmentation system (SBAS) data to get a more accurate position. The maximum output data rate that will be used is 10 Hz. Just as with the IMU sensor, a UART interface is used to receive the data from the GNSS sensor. The data frames are standardized following the National Marine Electronics Association (NMEA) messages [35]. These are checked, analysed, and parsed by the μ C. To reduce the traffic on the interface, only the important NMEA messages are enabled. In our case, we used the GGA (positioning fix data) and VTG (course and speed over ground) messages. Additionally, we use the ZDA (time and date) messages but this data will only be sent every five cycles. The GGA data includes standard position information such as latitude and longitude. The VTG message contains the speed and the course. Last but not least, the ZDA message provides

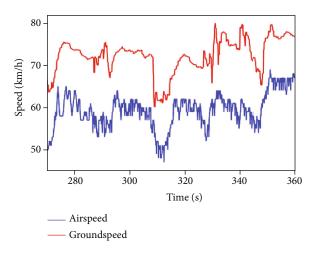


FIGURE 3: Comparison of the GNSS groundspeed (red) and the pitot tube airspeed (blue) in the case of wind.

the time and will be used to write the log files. Besides these messages, the GNSS sensor provides more information, but this is not necessary for the development of such a system.

The last important sensor is the pitot tube to measure the airspeed. In contrast to the GNSS speed, often called ground-speed, the airspeed represents the real speed in the air with consideration of the wind. Figure 3 is a comparison of the two speeds in the case of a tailwind.

The speed in the air is slower as on the ground and must be taken into account by the controller. To fly a fixed-wing aircraft, a minimal airspeed is required; otherwise, a stall will be caused and the aircraft will fall down. The tube for the air inlet needs to be placed away from the aircraft-generated turbulences. Therefore, we placed it on the outer end of the wing. The output of the sensor is the speed measured in km/h and can be used by the μ C directly.

The last part to complete the UAS is the telemetry. The first telemetry connection is used to control the aircraft manually. It is a commercial remote "Futaba T14SG" and communicates through 2.4 GHz. Usually, the manual control is stabilized by the "basic controller" for the roll and pitch axes. Complete control without any controller is possible but not recommended because of high demand by the pilot. Because of the commercial telemetry connection and therefore a limited use, a second connection "Telit LE70-868" was installed on the top PCB (Figure 2). This one has a higher range because of better antennas and a frequency of 868 MHz. The antenna of the aircraft is attached on the bottom side of the fuselage (see Figure 1). Through this connection, it is possible to control the UAV. This includes the possibility to (i) monitor the sensors and UAS status, (ii) set up the controller, (iii) verify the results with graphs, (iv) set up different flight modes, and (v) start or stop the autonomous navigation modes as well as the fly home function. This compact ground control station has a 10" touch screen and is supplied via 5 V USB power.

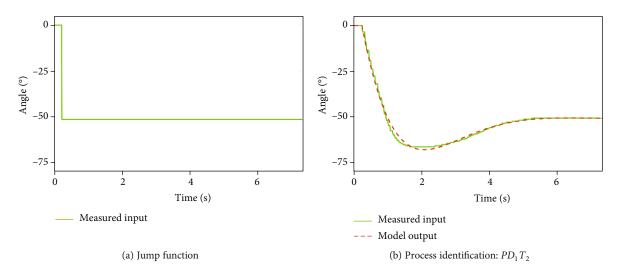


FIGURE 4: Process identification of the longitudinal axis. (a) shows the stimulation of the process, and (b) confirms the measured output and the process modulation.

3. Controller and Design of an Autopilot

The implemented sensor units permanently supply the μ C with clear information about flight attitude, angular velocity, airspeed, and groundspeed, as well as air pressure, humidity, temperature, and the current GNSS position. The sampling rates of the individual sensors were set to suitable values. For example, the sampling rate of the GNSS sensor as it was mentioned above was set to 10 Hz. The IMU sensor was set to 33.3 Hz. The accuracy and reliability of the individual sensors from the automotive industry can be considered of the highest security, because of the requirements in terms of testing. When selecting each sensor, a great importance was attached to the reliability, as the sensor data influence the individual control and thus the flight behaviour.

To develop a suitable autopilot for the fixed-wing airframe, it is necessary to know the plane's propulsion characteristics. With the available sensor data, process identification could be performed. With the knowledge of the process parameters, a suitable control algorithm was selected and adjusted later. An example of process identification is presented in Figure 4, which shows a step response of the longitudinal axis.

Figure 4(a) shows a jump function which stimulates the process. The deflection was estimated as 50 degrees, and a left curve was initiated. The measured step response was then compared to different differential equations by means of numerical optimization. A PD_1T_2 process behaviour showed the best agreement, which is shown in Figure 4(b). A PD_1T_2 transfer function for the longitudinal axis was calculated as follows (equation (1)):

$$G_{PD_1T_2}(s) = \frac{K(T_Z \cdot s + 1)}{s^2 \cdot T + s \cdot 2 \cdot d \cdot T + 1}.$$
 (1)

A process behaviour of the longitudinal axis reflects an oscillating system of the second order. With the knowledge about the process behaviour, a basis for the controller synthe-

sis is created, which will be discussed in more detail later. The behaviour of the lateral and vertical axis was carried out analogously. The result was a differential equation (2) of the second order:

$$G_{PT_2}(s) = \frac{K}{s^2 \cdot T + s \cdot 2 \cdot d \cdot T + 1}.$$
 (2)

Several step and ramp responses were recorded around the working point from real flights. The working point, in this case, is the stable position of an aircraft. No deflection in the longitudinal and lateral axes was found (0°) . The small-and large-signal behaviour was an essential factor to rebuild a realistic model. Then, smaller jumps were recorded and analysed after to determine the process behaviour.

While working out the physical behaviour of an aircraft in the air, the coupling of the axes to each other was noticed. For example, if the airframe was rolling around the longitudinal axis, a moment around the lateral axis was also generated. A coupling from the lateral axis to the longitudinal axis could not be demonstrated, which will be proven later. In order to react against these coupling effects, some more tests were also carried out in these areas and integrated into the controller algorithm. The coupling and decoupling structure of the lateral and longitudinal axes is shown in Figure 5.

In Figure 5, the lateral and longitudinal axes and their coupling are shown on the right side. If a coupled system exists, a decoupling controller can also be designed. With the knowledge of the complete process behaviour, a suitable control loop system can be then developed.

A *PID* controller structure was selected as the transfer function for the roll and pitch controllers. The decoupling system is a simple *P*-controller. The *PID* controller registers the advantage of controlling nonlinear systems and can be easily integrated into an embedded system. To implement the controller in the program code, the controller must be described mathematically first. The structure of a *PID* controller in the frequency domain is shown in Figure 6.

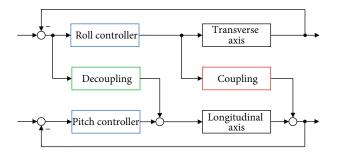


FIGURE 5: Decoupling structure of the lateral and longitudinal axes. After initiating a left or a right turn flight, the nose of the airframe is dropping. The controllers for stabilization are marked in blue, the physical coupling of the system in red, and the corresponding decoupling in green.

The general notation of a *PID* controller applies the following equation:

$$G_{PID}(s) = \frac{u(s)}{e(s)} = K_P + \frac{K_I}{s} + K_D \cdot s. \tag{3}$$

For the implementation of the C code, the transfer function of the PID controller must be converted into a difference equation. To do this, the transfer function must be first transformed into the z area by using the shift rule (equation (4)) [36]:

$$G_{PID}(z) = \frac{u(z)}{e(z)} = K_P + \frac{K_I}{z - 1} + K_D \cdot \frac{z - 1}{z}.$$
 (4)

This is followed by the retransformation into the time domain:

$$u_{k} = u_{k-1} + e_{k} \cdot (K_{P} + K_{I} + K_{D}) + e_{k-1}$$

$$\cdot (-K_{P} - 2 \cdot K_{D}) + e_{k-2} \cdot K_{D}.$$
(5)

This equation can be implemented in the microcontroller. Here, u_k represents the current controller output variable and u_{k-1} the preceding output variable. In order to ensure that the integral part of the regulator does not overflow and that the aircraft is not controllable in the worst case, an additional antiwindup limit was added.

With the control algorithm and the implemented sensor technology, the control loop was closed for the individual airframe axes. The closed control loop is shown in Figure 7.

With the conventional remote control, the pilot provides a command variable in the form of pulse width. This pulse width is detected by the receiver in the aircraft and transmitted to the microcontroller. The program code then allows converting the pulse width into an angle. For example, for a minimum/maximum joystick deflection to the left/right, an angular deflection for the roll controller of -50°/+50° is calculated.

The minimum/maximum angle can be defined by the source code itself. The reference variable, thus specified, is compared to the angles of the IMU sensor and is passed on to the controller algorithm as a control difference. This controller is able to calculate a manipulated variable to guide the

aircraft to the desired position. In this case, the basic controller includes a roll and pitch control, which ensures that the aircraft is stabilized in the air. Disturbances such as gusts should be compensated with this controller system. With this algorithm, the pilot can permanently set a command variable and his desired flight behaviour.

After the implementation of the closed control loop, it was still necessary to define the controller parameters. The determination of these parameters can be performed by different methods. Due to the knowledge of the process behaviour, numerical optimization was used in this design in order to find the global minimum of a specially defined quality function. The quality function can be defined according to energy consumption, speed, and another behaviour. In our case, the focus was placed on fast control with as little overshoot as possible in the controlled variable. Overshoot can nevertheless occur due to external disturbances such as gusts. Therefore, a quadratic optimization method was used following equation (6) [36]:

$$J = K_e \cdot \int_0^T e^2(t) \, dt + K_u \cdot \int_0^T u^2(t) \, dt. \tag{6}$$

Equation (6) describes the quality function, which is calculated from the integral of the control deviation (e) and the output variable (u). The parameters K_e and K_u represent the factors considering how much the quality of each part has weight. If the user wants to obtain smaller differences in the control deviation, the parameter K_e should be increased. This quality is minimized by numerical optimization. This means that the control deviation and output variable should be kept to a minimum. In most cases, these are contradictory statements. If the control deviation has to be reduced, more energy must be applied by the controller, which results in an increase in the output variable of the energy consumption.

The numerical optimization for the roll control resulted in controller settings (see Table 1).

In Figure 8, the commissioning of the roll and pitch controllers is shown. In the simulation, these control parameters can be adapted to different flight scenarios. These parameters are then implemented in C code. By means of a ground station, the parameters can be adjusted afterwards and also during the flight. By simulating the closed control loop, the first control parameters could be calculated, which were subsequently tested in commissioning. The commissioning of the stabilization control is an essential step, which is the most important component of aircraft control. Figure 8 shows the control behaviour of the roll and pitch controllers.

The dynamics of the aircraft is strongly dependent on the speed relative to the air of the airframe. In Figure 8, the airspeed of the aircraft is 72 km/h. The airspeed is not yet regulated in the current state of the autopilot and must be controlled manually by the pilot. The commissioning of the basic controller according to Figure 7 has been carried out successfully. As already shown in Figure 5, there is coupling from the longitudinal axis to the lateral axis, but not the other way around. After the first test flights, a decoupling *P*

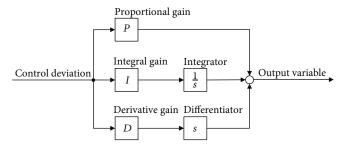


FIGURE 6: Structure of the PID controller in the frequency domain. The controller is set using the parameters P, I, and D.

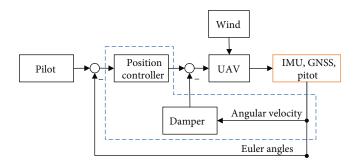


FIGURE 7: Structure of the closed control loop system. The IMU is shown on the right side in orange and measures the Euler angles every 30 ms. The pilot commands the set value with a commercial remote. The difference between the set value and the actual value is offered by the controller, which calculates itself a control signal to get the airframe to the desired position. The damper damps the own movement of the airframe, and the complete control system is also sensitive to disturbances such as wind gusts. The internal control algorithm is called "basic controller" and is framed in blue.

Table 1: Parameters of the controller by numerical optimization.

Parameter	Value
K_{p}	0.81
K_I	0.33
K_D	0.04

-controller is implemented. In Figure 9, the controller behaviour is depicted.

In a left or right curve, the pitch angle is deflected and must be corrected. If the pitch angle is deflected, the roll angle is not affected.

In order to be able to control the airspeed, course, and altitude as well, the closed control loop of Figure 7 must be extended. The desired multivariable control was implemented using a cascade control loop. In Figure 10, the extended control loop is shown as a cascade control loop.

Before describing the cascade regulation in more detail, it is necessary to deal with a flight route and main elements that have to be conducted. The user can plan the desired flight route in advance using map software. Depending on the software, different parameters can be configured for the flight. The default output parameters are the geographical points with the corresponding altitude at which the airframe should fly to. The points are stored as a list on an SD card and connected to the microcontroller before flying. The navigation system reads the SD card and stores the GNSS list in the flash of the microcontroller. If the navigation control is switched on, the module calculates the course

to be flown to reach the next navigation point. In addition, the navigation system calculates the distance of the aircraft to the next point in real time. If the aircraft is in a defined epsilon area, the target is considered to be reached and the next position is approached. In case of emergency, the pilot can set a different course during a flight and has priority over the navigation system.

The desired course is thus set by the pilot or by the navigation system as a command variable and then compared with the current course of the aircraft. The difference in the course is passed on to the autopilot, which calculates to what extent the aileron and rudder must be deflected to bring the aircraft on the desired course. The autopilot also calculates the amount of airspeed required to fly the aircraft efficiently. The output of the autopilot is given by 4 command variables: the airspeed, roll, pitch, and heading angle of the aircraft. The internal basic controller from Figure 7 includes the airspeed and heading control after the expansion. To control these effects, a GNSS, pitot tube, and IMU sensor are needed.

When designing the cascade control systems, care was taken to ensure that the internal control loops are designed quickly, while the external control loops ensure stationary accuracy. This rule must be observed to keep the overall system controllable and stable. If the internal control is too slow, the desired control quality will not be achieved. The course controller is a *P*-controller and the altitude controller a *PI*-regulation. In addition, it is necessary to pay attention to not have multiple integral parts in the controller loops. For example, the altitude controller should provide the control variable for the pitch controller. The pitch controller will then adjust the elevator to cause movement around the lateral

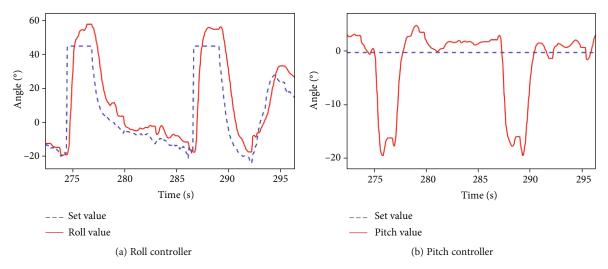


FIGURE 8: Behaviour of the roll and pitch controllers. In (a), the pilot changes the set value from -20° to +40°. The roll controller follows the command and tries to reach the desired position. Because of the coupling system, the pitch angle in (b) changes too and the airframe is dropping. The pitch controller and the decoupling controller are reacting to this and set the airframe in the right position.

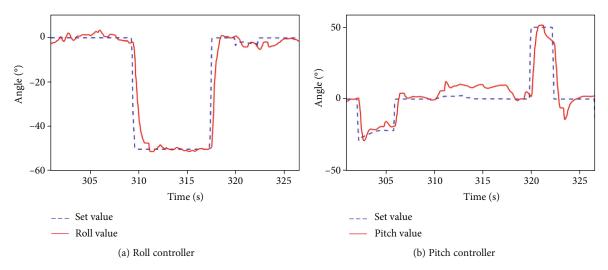


FIGURE 9: Behaviour of the roll and pitch controllers. During this time, the roll angle from (a) shows no deflection angle. In (b), the pilot changes the set value from 0° to -25°. The pitch controller is reacting to this command and reaches the commanded angle. While the airframe is flying, a left flight turns the pitch angle positive, which means that the pitch controller and the decoupling system are too strong. At 320 seconds, the airframe gets a command to fly upwards; the roll angle shows no deflection from this command.

axis. The process is of the second order and therefore capable of oscillation. In order to not stimulate the process of oscillation, the integral component in the pitch controller is deleted and, subsequently, implemented in the altitude controller.

The course and altitude controls were the first to be put into operation. The speed to be flown was still specified by the pilot and in most cases reached 65 km/h. As this is an oscillation system, it was decided to not operate several control loops at the same time. Regarding the computing time for the calculation of the course and the required flight altitude, approx. 540 microseconds was measured. For the first tests, the airframe should fly a square, which defines four navigation points. The desired altitude is about 360 absolute heights. The test field is 300 meters high that represents a rel-

ative height of 60 meters. In Figure 11, the square with the four navigation points is shown. The commanded way between these four points is also drawn.

The cascade control loop is able to control the airframe from one navigation point to another. Every time the airframe reaches the epsilon area of the current destination, the navigation system calls the next GNSS point the airframe has to be approached. If the airframe finishes the route, the user has two possibilities: (i) repeating the route or (ii) flying back to the home spot. The user himself generates the home spot. The epsilon environment for this measurement is 15 meters and can be modified in the software. When planning the flight route to be conducted, the minimum and maximum possible curve radius must be taken into account so that

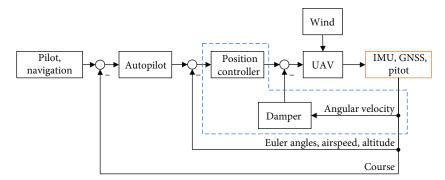


FIGURE 10: Development of the cascade control loop. To control the navigation course, airspeed, altitude, and Euler angles, an IMU, GNSS, and pitot tube sensor are needed. On the left side, the navigation system or the pilot commands the course the airframe should fly. The control deviation is transferred to the autopilot which calculates the needed roll angle to reach the desired course. The autopilot also commands the airspeed and required altitude. The commands were compared to the sensor data. The internal basic controller in the blue dashed frame ensures that the airframe reaches the final position.

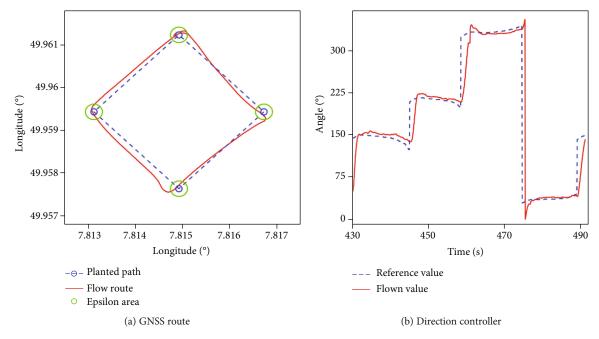


FIGURE 11: The desired route and the navigation points (red) the airframe has to reach (a). If the airframe attains the epsilon area (green), the navigation unit commands to fly to the next destination. The flown route is shown in black. In (b), the commanded course (blue) and the flown curse (red) are contrasted. The planned altitude (blue) and the actual altitude (red) are shown.

the aircraft can reach the individual points. If a navigation point is not reached, the aircraft is turned around and tries to fly this point again. To reduce the turn radius and to fly smaller turns, the rudder can be mixed with the roll controller. The rudder is additionally operated with each turn.

In the current example, it is also noticeable in Figure 11(a) that a constant wind from the northeast flows onto the aircraft and deflects it. The control algorithm reacts to this and adjusts its control variables accordingly. In Figure 11(b), it can be noted that when the reference variable changes, the aircraft has reached its stationary final value after approx. 2.5 seconds and is on the desired course. Due to the mentioned disturbances caused by wind influences, there are minimal overshoots in control behaviour. In order to reach the desired predetermined flight altitude, the altitude

controller influences the pitch controller, which adjusts the elevator. This adjustment alone is not sufficient as shown in Figure 12.

This is because the airspeed of the aircraft has a significant influence on the lift. As already told, the wind influences cause larger jumps in the airspeed. If the aircraft is too fast, the altitude will automatically increase when the aircraft is in a neutral position. In order to be able to control the altitude reliably and precisely, the airspeed must also be controlled.

4. Conclusion, Challenges, and Future Research

In this paper, hardware design and a control system for a fixed-wing aircraft with a span of 1.6 m were developed. A

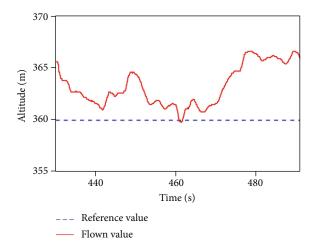


FIGURE 12: Altitude controller with the rudder as the only actuator. To control the altitude, the rudder itself is not enough; therefore, the airspeed has to be controlled, and if the airframe has to rise, the airspeed should not break down.

high-quality sensor system was of utmost importance for the selection. The subsequent implementation of the components was performed on the embedded system in the programming language "C". Subsequently, process identification of the fixed-wing aircraft was carried out and a simulation was created on this basis. Thereby, a coupling of the systems was identified and a subsequent decoupling was developed. A stabilization control of the longitudinal and lateral axes was developed for the aircraft. By means of quadratic and numerical optimization, the emphasis was placed on the smallest possible deviation in the control difference and manipulated variable. The required PID control algorithm was implemented in the program code, and the control parameters used can be adapted to the external conditions in real time. After successful commissioning, the implemented control system was extended to a cascade control system to enable multivariable control. Thus, the foundation for the control of heading, altitude, and airspeed was laid. Flying of defined routes was successfully tested, and the behaviour of the switched-on controllers was investigated in the laboratory.

The commissioning of the airspeed control is still pending for future investigations. This control system should also enable the desired flight altitude to be set even better. In the future, it should be possible to test the controller algorithm in other aircrafts and integrate additional sensors such as Radar and Lidar into the aircraft. Additionally, we want to change the GNSS sensor to a measuring unit with an external antenna and an internal battery. The battery ensures that the sensor can be used immediately at the start-up. Subsequent tests will be performed to prove an increased quality of the sensor data. The main advantages of the designed autopilot are the low costs, the high quality of the hardware, the possibility to adjust the control characteristics in real time, and the communication between the pilot and the aircraft via a ground control station. The effectiveness of the system was validated during autonomous flights of a fixed-wing aircraft and further optimized in the laboratory.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research work was supported by "European Regional Development Fund" (EFRE) in context to the aim "Investment in Growth and Employment" (IWB) in Rhineland-Palatinate, Germany.

References

- [1] L. Royimani, O. Mutanga, J. Odindi, T. Dube, and T. N. Matongera, "Advancements in satellite remote sensing for mapping and monitoring of alien invasive plant species (AIPs)," *Physics and Chemistry of the Earth, Parts A/B/C*, vol. 112, pp. 237–245, 2019.
- [2] Y. Huang, Z. Chen, T. Yu, X. Huang, and X. Gu, "Agricultural remote sensing big data: management and applications," *Jour*nal of Integrative Agriculture, vol. 17, no. 9, pp. 1915–1931, 2018
- [3] A. Matese, P. Toscano, S. di Gennaro et al., "Intercomparison of UAV, aircraft and satellite remote sensing platforms for precision viticulture," *Remote Sensing*, vol. 7, no. 3, pp. 2971–2990, 2015.
- [4] J. Xue and B. Su, "Significant remote sensing vegetation indices: a review of developments and applications," *Journal of Sensors*, vol. 2017, 17 pages, 2017.
- [5] A. Khaliq, L. Comba, A. Biglia, D. Ricauda Aimonino, M. Chiaberge, and P. Gay, "Comparison of satellite and UAV-based multispectral imagery for vineyard variability assessment," *Remote Sensing*, vol. 11, no. 4, p. 436, 2019.
- [6] J. S. Aber, I. Marzolff, and J. B. Ries, Small-Format Aerial Photography: Principles, Techniques and Geoscience Applications, Elsevier, Amsterdam, 1st edition, 2010.
- [7] C. Heipke, W. Freeden, and R. Rummel, Eds., *Photogrammetrie und Fernerkundung*, Springer Spektrum, Berlin, 1st edition, 2017.
- [8] A. S. Laliberte, C. Winters, and A. Rango, "UAS remote sensing missions for rangeland applications," *Geocarto International*, vol. 26, no. 2, pp. 141–156, 2011.
- [9] A. D. Tamminga, B. C. Eaton, and C. H. Hugenholtz, "UAS-based remote sensing of fluvial change following an extreme flood event," *Earth Surface Processes and Landforms*, vol. 40, no. 11, pp. 1464–1476, 2015.
- [10] C. H. Hugenholtz, B. J. Moorman, K. Riddell, and K. Whitehead, "Small unmanned aircraft systems for remote sensing and earth science research," *Eos, Transactions Ameri*can Geophysical Union, vol. 93, no. 25, pp. 236–236, 2012.
- [11] J. S. Aber, I. Marzolff, J. B. Ries, and S. W. Aber, Small-Format Aerial Photography and UAS Imagery: Principles, Techniques and Geoscience Applications, 2019.
- [12] X. Jin, S. Liu, F. Baret, M. Hemerlé, and A. Comar, "Estimates of plant density of wheat crops at emergence from very low

- altitude UAV imagery," Remote Sensing of Environment, vol. 198, pp. 105–114, 2017.
- [13] M. N. Reza, I. S. Na, S. W. Baek, and K.-H. Lee, "Rice yield estimation based on K-means clustering with graph-cut segmentation using low-altitude UAV images," *Biosystems Engineering*, vol. 177, pp. 109–121, 2019.
- [14] A. Malehmir, L. Dynesius, K. Paulusson et al., "The potential of rotary-wing UAV-based magnetic surveys for mineral exploration: a case study from central Sweden," *The Leading Edge*, vol. 36, no. 7, pp. 552–557, 2017.
- [15] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma, "Car detection from low-altitude UAV imagery with the faster R-CNN," *Journal of Advanced Transportation*, vol. 2017, 10 pages, 2017.
- [16] A. Nikulin and T. S. de Smet, "A UAV-based magnetic survey method to detect and identify orphaned oil and gas wells," *The Leading Edge*, vol. 38, no. 6, pp. 447–452, 2019.
- [17] D. Wierzbicki, M. Kedzierski, and A. Sekrecka, "A method for dehazing images obtained from low altitudes during highpressure fronts," *Remote Sensing*, vol. 12, no. 1, p. 25, 2020.
- [18] H. González-Jorge, J. Martínez-Sánchez, M. Bueno, and A. P. Arias, "Unmanned aerial systems for civil applications: a review," *Drones*, vol. 1, no. 1, p. 2, 2017.
- [19] B. L. Stevens and F. L. Lewis, "Aircraft Control and Simulation," Aircraft Engineering and Aerospace Technology, vol. 76, no. 5, 2004.
- [20] R. W. Pratt, Ed., Flight Control Systems: Practical Issues in Design and Implementation, Repr, Institution of Engineering and Technology, Stevenage, 2008.
- [21] K. P. Valavanis, Handbook of Unmanned Aerial Vehicle, Springer, New York, 2014.
- [22] R. Lozano, *Unmanned Aerial Vehicles: Embedded Control*, ISTE: Wiley, London; Hoboken, NJ, 2010.
- [23] R. Vepa, Nonlinear Control of Robots and Unmanned Aerial Vehicles: An Integrated Approach, CRC Press/Taylor & Francis Group, Boca Raton, 2017.
- [24] E. Baskaya, G. Manfredi, M. Bronz, and D. Delahaye, "Flexible open architecture for UASs integration into the airspace: Paparazzi autopilot system," in 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), pp. 1–7, Sacramento, CA, USA, September 2016.
- [25] K. Bakuła, A. Salach, D. Z. Wziątek, W. Ostrowski, K. Górski, and Z. Kurczyński, "Evaluation of the accuracy of lidar data acquired using a UAS for levee monitoring: preliminary results," *International Journal of Remote Sensing*, vol. 38, no. 8–10, pp. 2921–2937, 2016.
- [26] W. Yuan, J. Li, M. Bhatta, Y. Shi, P. S. Baenziger, and Y. Ge, "Wheat height estimation using LiDAR in comparison to ultrasonic sensor and UAS," *Sensors*, vol. 18, no. 11, p. 3731, 2018.
- [27] C. Leuschen, R. Hale, S. Keshmiri et al., "UAS-based radar sounding of the polar ice sheets," *IEEE Geoscience and Remote Sensing Magazine*, vol. 2, no. 1, pp. 8–17, 2014.
- [28] A. F. Scannapieco, A. Renga, G. Fasano, and A. Moccia, "Experimental analysis of radar odometry by commercial ultralight radar sensor for miniaturized UAS," *Journal of Intelligent and Robotic Systems*, vol. 90, no. 3-4, pp. 485–503, 2018.
- [29] J. Rodrigo-Comino, S. D. Keesstra, and A. Cerdà, "Connectivity assessment in Mediterranean vineyards using improved stock unearthing method, LiDAR and soil erosion field surveys," *Earth Surface Processes and Landforms*, vol. 43, no. 10, pp. 2193–2206, 2018.

- [30] M. Lang, O. McDonough, G. McCarty, R. Oesterling, and B. Wilen, "Enhanced detection of wetland-stream connectivity using LiDAR," Wetlands, vol. 32, no. 3, pp. 461–473, 2012.
- [31] M. López-Vicente and N. Ben-Salem, "Computing structural and functional flow and sediment connectivity with a new aggregated index: a case study in a large Mediterranean catchment," *Science of The Total Environment*, vol. 651, Part 1, pp. 179–191, 2019.
- [32] S. Ramasamy, R. Sabatini, A. Gardi, and J. Liu, "LIDAR obstacle warning and avoidance system for unmanned aerial vehicle sense- and-avoid," *Aerospace Science and Technology*, vol. 55, pp. 344–358, 2016.
- [33] J. B. Campbell and R. H. Wynne, Introduction to Remote Sensing, Fifth Edition, 5 New Edition, Guilford Publications, New York, 2011.
- [34] K. Elliott and C. J. Hegarty, Understanding GPS/GNSS: Principles and Applications, 3 Revised Edition, Artech House Books, London, 2017.
- [35] NMEA National Marine Electronics AssociationFebruary 2020, https://www.nmea.org/.
- [36] R. C. Dorf and R. H. Bishop, *Modern Control Systems, Thirteenth edition*, Pearson, Hoboken, 2016.