SYSTEM IDENTIFICATION DRIVEN HIGH-PERFORMANCE CONTROLLER DESIGN FOR AN AGILE SMALL ROTORCRAFT UAV

P. Petit, S. Seher-Weiß, J. Wartmann and A. Donkels
German Aerospace Center (DLR), Institute of Flight Systems, Brunswick, Germany

Abstract

This paper presents an efficient methodology for improving an existing legacy controller for the MIDIARTIS, a very agile 10 kg rotorcraft Unmanned Aerial Vehicle (UAV) by leveraging the same legacy controller as a basis for system identification flight tests. The legacy autopilot was modified to enable the injection of system identification signals into the control reference values and to actuator signals during flight. This allowed the excitation of low, middle and high frequencies of the rotorcraft UAV with an active controller stabilizing the UAV either in hover condition or on a trajectory during forward flight. The resulting data was then used to identify a physical 9-degrees of freedom model, accounting for rotor flapping and engine states. Based on this model, an improved flight controller was developed which is shown to greatly improve agility and precision of the rotorcraft UAV.

NOTATION

a_x, a_y, a_z	linear accelerations, m/s^2
a, b, c	feedforward filter parameters
$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$	state-space matrices
e	controller error
g	gravity constant, m/s^2
k_P, k_I, k_D	PID controller gains
p, q, r	angular rates, rad/s
s	laplace variable
u, v, w	aircraft-fixed airspeed components, m/s
$\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{y}$	control, state, and output vector
z	discrete-time laplace variable
β_{1c}, β_{1s}	longitudinal/lateral tip-path-plane flapping angle, rad
δ_x, δ_y	longitudinal/lateral cyclic controls, %
δ_p, δ_0	pedal and collective controls, %
ϕ, θ, ψ	attitude angles, rad
Λ	outer loop command vector
τ_f	flapping time constant, s
T_s	sampling time, s
ω	angular velocity, rad/s
ω_n	natural frequency, rad/s
ζ	damping constant
ref	reference
SID	system identification
TC	turn coordination

DoF	Degrees of Freedom
GCS	Ground Control Station
GBN	Generalized Binary Noise
JIO	Joint Input-Output
UAV	Unmanned Aerial Vehicle
QGC	QGroundControl

1. INTRODUCTION

Increased automation tendencies for manned and unmanned helicopters including automatic and precise slung load positioning, aerial refueling, or automated landings on moving platforms, pose a significant challenge for the flight control systems of these vehicles. All these applications demand precise and fast control responses to successfully complete the tasks.

A very popular approach for flight control algorithms used in many integrated autopilot systems, is the use of a cascaded flight control structure. The faster rotational dynamics are handled by an inner feedback loop which receives command values from an outer loop, designed for handling the slower, for example, translational, dynamics. Cascaded flight controllers offer a clear structure, high flexibility and can be easily adapted to implement different flight command modes such as Attitude Command / Attitude Hold (ACAH), Translational Rate Command (TRC), etc. and are well proven in countless applications [1, p. 322], [2, 3]. Additionally, the cascaded control system can be easily hand-tuned without extensive experience, which further motivates the usage of this control setup in real-world scenarios.

A good example is the popular Pixhawk PX4 autopilot control software, which was also used for the work of this paper. The PX4 autopilot uses this control structure for different vehicle types such as various aerial marine and land vehicles [3]. However, due to the underlying time separation principle, which states that the inner loop has to be significantly faster than the outer loop, the overall dynamics of the complete controller is limited by the performance of the inner loop.

This can be a significant problem if the performance of the existing autopilot is insufficient for the required task or, if new outer loop modes shall be implemented that require higher accuracy, control bandwidth or a better disturbance rejection of the existing inner loop. In this case, it is important to realize that the existing controller is not a liability but rather an advantage in retuning or redesigning a controller with improved performance.

To illustrate this point, this paper shows the design process of redesigning an existing flight controller by leveraging the existing controller to assist in the identification of accurate system models using the example of the MIDIARTIS UAV. This paper is structured as follows: First, the MIDIARTIS UAV, its legacy flight controller and its modification for system identification purposes are presented as well as the associated flight tests. Then, the system identification methodology and the (re)-design of the improved inner loop flight control structure are described. The paper concludes with the validation of the improved controller

2. EXPERIMENTAL SETUP

The experiment consists of the MIDIARTIS vehicle and the ground control infrastructure, both of which were modified to enable flight tests for system identification. This section will describe the base system and their modification.

2.1. MIDIARTIS

The MIDIARTIS flight test vehicle, shown in Figure 1, is a compact and easy to operate battery-electric single main rotor helicopter drone with a weight of about 10 kg offering exceptional agility as it is a converted aerobatic radio-controlled helicopter.

The MIDIARTIS is configured as dual-bladed main rotor with additional Bell-Hiller flybars and conventional tail rotor. The flybars consist of aerodynamically active paddles which serve as a mechanical link between the swashplate and the main rotor. Conclusion of [2] on data provided in [4] are that the paddles introduce a stabilizing effect at low frequency dynamics, which increases the phase margin for pilot inputs in the crossover frequency range. Furthermore, flybars are of considerable weight, producing a gyroscopic effect which further stabilizes the rotor disc against wind gusts and turbulence. Side-effects are an increased response time constant of the rotor reducing the damping of coupled fuselage and rotor-flap dynamics. The MIDIARTIS has been converted to a research testbed for the purpose of low-cost data generation for system identification, demonstration of novel flight control and autonomy architectures enabling dynamic highly automated missions.

Figure 1: The MIDIARTIS flight test vehicle

The original PX4 flight control software used for MIDIARTIS flight tests has been modified to accommodate a custom cascaded flight control structure displayed in Figure 2. The legacy flight control architecture consists of a total of three PID-cascades, first an outer loop which implements a position hold or velocity hold/path following mode depending on user selection, then an angle-loop and finally an angular velocity feedback-loop. The PID-parameters of the complete feedback controller setup were tuned based on a preliminary linear system identification model of the MIDIARTIS in hover condition and were then fine-tuned by hand in flight. This control structure and tuning proved to be robust as it was able to reliably stabilize the MIDIARTIS from hover up to 25 m/s forward velocity with one set of controller gains. However, this controller did not provide the needed close-loop performance for highly agile operations, concerning bandwidth and tracking accuracy with a rise times to angle step commands of $t_{90} \approx 1$ s and a position hold accuracy in hover of around 1 m. Therefore, this flight controller was deemed to be very robust but too inaccurate for challenging tasks such as precision landings etc.. Further analysis has shown that the shortcoming is mainly attributed to the inner loop cascade being too slow and unable to handle quicker commands from the outer loops.

For the flight tests described in this paper, the legacy flight controller and the ground control software were modified such that excitation signals can be added to the control loop at several points, either under manual or under automatic control, depending on the type of signal selected. These signals are marked in red in Figure 2.

The following types of system identification excitation signals can be injected, either right before the actuator, or at the actuator and command input combined:

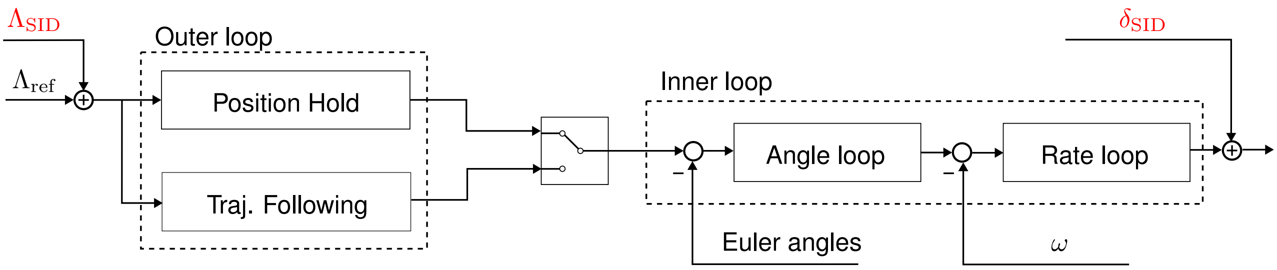


Figure 2: Structure of the legacy controller (system identification signals are marked in red with the subscript $_{SID}$)

1. Single steps,
2. Doublets,
3. (Modified) 3-2-1-1 and 2-3-1-1 multi-steps,
4. Generalized Binary Noise (GBN) signals,
5. Frequency sweeps.

Single step, doublets, and modified 3-2-1-1 respectively 2-3-1-1 excitation signals are relatively short inputs which switch back and forth between two levels, commonly used for a short excitation of a vehicle. Frequency sweeps are the standard excitation signal for system identification with a duration between about 10 s and 120 s depending on the slowest eigenvalue of the system to be identified [5]. They are single axis sinusoidal excitation signals which rise or fall in frequency during their engagement. GBN input signals are computer-generated signals switching randomly between two specific signal levels to excite the system in a dedicated frequency range [6]. Independent GBN signals are well-suited for multi-axis excitation.

2.2. Ground Control Station

An instrumental part of the flight test setup is the modified Ground Control Station (GCS). For this, the QGroundControl (QGC) open-source software was used employing the Mavlink protocol to interface open-source autopilots like PX4 and Ardupilot [7]. The QGC user interface is designed to access high-level autopilot functions used for compiling complex missions and triggering automated vehicle functions.

However, the system identification methodology as proposed in this paper requires additional user interfaces to easily access and modify the system identification control loops and to increase the overall transparency of the autopilot system as the control algorithms themselves are newly designed.

The focus of the user interface extensions lies on fast modification of the maneuver used for identification, which includes the choice of the system excitation signal (e.g. steps or sweeps), the control axis, as well as their parametrization (e.g. times or frequencies, and amplitudes).

The quick access ensures an optimal use of the limited flight time of the MIDIARTIS and allows to complete a set of desired maneuvers under uniform environmental and flight conditions. Figure 3 shows the implemented GUI; an additional side window consisting of buttons, sliders, and textfields to modify a system identification maneuver. It furthermore features a system identification timer to provide feedback on the duration of flown maneuver.

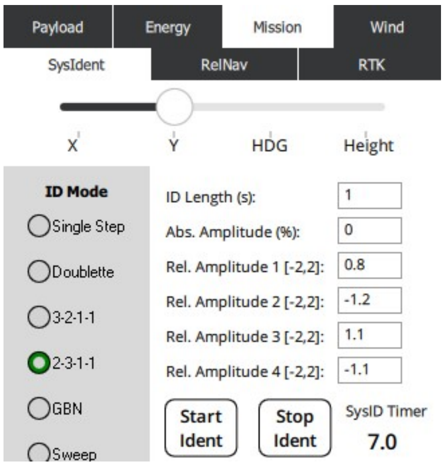


Figure 3: QGC GUI for selecting system identification mode and parameters

Real-time data of vehicle dynamics can also be observed in a separate window, providing a fully configurable set of time-series plots for any recurring status information received from the UAV including scaling as well as a data cursor and a stop button to freeze the graphs allowing deeper analysis of some data.

3. SYSTEM IDENTIFICATION

A series of system identification flight tests were executed in September of 2024 over the span of four separate days with a total flight time of around 3 h. The goal was to generate data for system identification and model validation in hover condition, 10 m/s and 20 m/s forward flight.

3.1. Flight tests

For the system identification of the MIDIARTIS, three types of excitation signals were used:

- Modified 2-3-1-1 signals,
- Frequency sweeps,
- Generalized Binary Noise (GBN).

The multi-step excitation signals were injected for the purpose of bare-airframe model validation, and were activated without active flight controller. The fundamental time interval Δt of the 2-3-1-1 signal has to be chosen such that the frequency range of interest is excited (see ch. 2 of [8]). Due to the fast eigendynamics of the midiARTIS, a value of Δt = 0.5 s was chosen, resulting in an overall length of 3.5 s for the whole maneuver. Only a basic stability augmentation in form of a PI-controller stabilizing the yaw rate r was active during these maneuvers to reduce the workload for the safety pilot. Other axes did not have to be stabilized as the flybar system offers natural stability for the pitch and roll axis. Thus, the MIDIARTIS was controlled by the safety pilot during these tests. Consequently, only the actuator injection point inputs δ_{SID} as marked in Figure 2 are active during the activation of these signals. The safety pilot was instructed to fly the aircraft in trim condition and to avoid any manual inputs during the excitation.

GBN and sweep signals on the other hand are injected for a length of up to 100 s. This long activation duration poses a challenge at higher speeds, as the vehicle quickly leaves the range where the pilot is able to effectively control the vehicle. E.g. if at a speed of 20 m/s the excitation signal duration is 100 s, the vehicle covers a distance of around 2 km during the engagement of the excitation signal. If this distance is flown in a straight line, the safety pilot cannot observer nor control the vehicle over the complete trajectory of the vehicle without additional help. It was therefore decided to fly the longer system identification signals *with* activated legacy controller. The controller stabilized the MIDIARTIS on a racetrack course and performed the system identification maneuvers (e.g. GBN and sweep inputs) automatically with the autopilot engaged. This method offers the advantage that the pilot only supervises the flight during system excitation which increases accuracy and repeatability of the flight data while limiting the maximum distance to the vehicle.

The actuator injection signals

$$\delta_{\mathrm{SID}} = \begin{pmatrix}\delta_{x,\mathrm{SID}} & \delta_{y,\mathrm{SID}} & \delta_{p,\mathrm{SID}} & \delta_{0,\mathrm{SID}}\end{pmatrix}^T \tag{1}$$

are used to excite the mid- to high-frequency vehicle dynamics. On the other hand, the command signals in position hold mode

$$\Lambda_{\mathrm{SID}} = \begin{pmatrix}x_{\mathrm{SID}} & y_{\mathrm{SID}} & \psi_{\mathrm{SID}} & h_{\mathrm{SID}}\end{pmatrix}^T \tag{2}$$

or in trajectory following mode

$$\Lambda_{\mathrm{SID}} = \begin{pmatrix}y_{\mathrm{SID}} & V_{\mathrm{SID}} & \psi_{\mathrm{SID}} & h_{\mathrm{SID}}\end{pmatrix}^T \tag{3}$$

respectively cover the low-frequency dynamics as low-frequency (input) disturbances are generally well-suppressed by the active flight controller. Note that the signals Λ_{SID} are added to the standard reference values Λ_{ref} commanded to the outer loop as shown in Figure 2.

The advantage of GBN inputs is that these can be easily used to excite all axis simultaneously which leads to significantly reduced flight test time compared to frequency sweep inputs, as these only excite one axis at a time which quadruples necessary flight test time. This test methodology has been successfully used in system identification flight tests in the past for example on the larger SUPERARTIS testbed [9]. An exemplary set of plots for GBN excitation during one flight with engaged autopilot flying the MIDIARTIS on a racetrack at 10 m/s and 20 m/s can be found in Figure 4. During the shown flight, GBN excitation of 100 s duration was activated at t=310 s, 445 s, and 630 s. The GBN excitation of the speed command V_{ref} is well visible in the plot, as the commanded velocity switches randomly between the aforementioned two levels.

This approach of generating system identification data proved to be highly successful, as the workload for the safety pilot and the monitoring flight engineers were minimal. After triggering either the GBN or sweep excitation, the vehicle automatically hovers or follows its programmed path at the speed set by the ground operators without any intervention required, which made it possible to perform several identification tests during one sortie. After preliminary evaluation of the data, the GBN excitation was deemed to be favourable as it allows the simultaneous excitation of all axes in contrast to single axis sweeps. While in theory it is possible to implement sweeps which also excite all axis simultaneously, such multi-axis sweeps require careful engineering to eliminate undesired cross correlations between the axes.

3.2. Frequency Response Generation

Frequency responses form the database for an identification with the frequency response method. For identification with other methods, frequency responses are used to assess model fidelity in the frequency domain. Frequency responses are usually generated from frequency sweep excitation maneuvers but can also be generated from maneuvers with GBN excitation.

In general, the frequency response matrix $H(j\omega)$ of a Multi-Input/Multi-Output (MIMO) system is calculated as

$$H(j\omega) = \left[\frac{y}{u}(j\omega)\right] = G_{uu}^{-1}(j\omega)G_{uy}(j\omega) \tag{4}$$

where G_{uu} is the matrix of auto- and cross-spectra between the control inputs and G_{uy} are the cross-spectra between the control inputs $\boldsymbol{u}$ and the outputs $\boldsymbol{y}$ (see chapter 9 of [5]).

When high correlation exists between the control inputs $\boldsymbol{u}$ (e.g. due to controller crossfeed), the matrix G_{uu} will be nearly singular and cannot be inverted. Ref. [5] gives the following guideline for the average cross-control co-

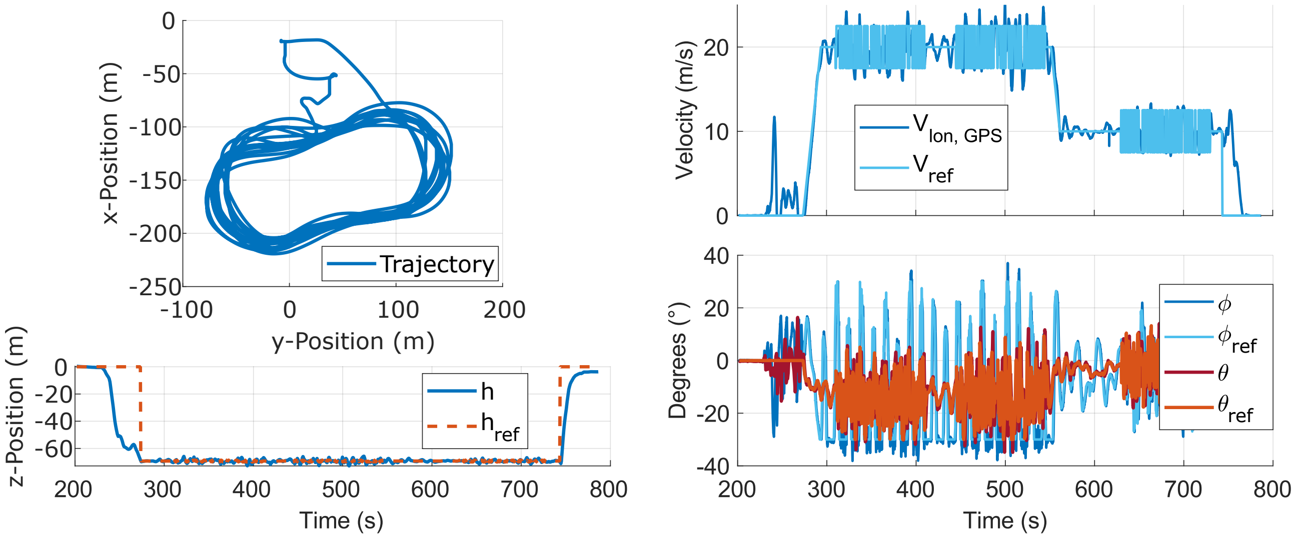


Figure 4: Flight data of the MIDIARTIS with GBN excitation during forward flight on a racetrack

herence $\gamma^2_{u_k,u_l}$ under which the matrix inversion in (4) can be performed:

(5) $$\left(\gamma^2_{u_k,u_l}\right)_{\text{ave}} < 0.5 \quad \text{for} \quad k \neq l.$$

If (5) is not satisfied but the average secondary autospectra G_{u_k,u_l} are small compared with the average primary autospectrum G_{u_k,u_k},

(6) $$\left(G_{u_k,u_l}\right)_{\text{ave}} - \left(G_{u_k,u_k}\right)_{\text{ave}} \leq -20\,\text{dB} \quad \text{for} \quad k \neq l$$

then the secondary input can be ignored and the Single-Input/Single-Output (SISO) solution for the frequency response can be used.

In case of the midiARTIS, the built-in stability augmentation of the yaw axis in the legacy flight control system is always active when the UAV is controlled by the safety pilot. The effect of this stabilization is shown in Figure 5 for some 2-3-1-1 multistep inputs.

Since all maneuvers for system identification were flown either with manual stabilization or with active control system, the above mentioned conditions had to be checked before the frequency response generation could be performed. Figure 6 shows that high correlation exists between δ_0 and δ_p, and that the difference in amplitude for the autospectra is also small. For δ_y and δ_p, as well as δ_0 and δ_y, the cross correlation is near the limit. All other control signal combinations are uncritical and thus not shown in the figure.

Thus, the desired bare-airframe frequency responses could not be determined directly from the sweep maneuvers, but the Joint Input-Output (JIO) method from [10] had to be used to calculate a valid frequency response matrix. This method requires a set of uncorrelated control inputs for which the commanded system identification control inputs ($\delta_{x,\text{SID}}, \delta_{y,\text{SID}}, \delta_{p,\text{SID}}, \delta_{0,\text{SID}}$) could be used. No critical cross correlations were present for the

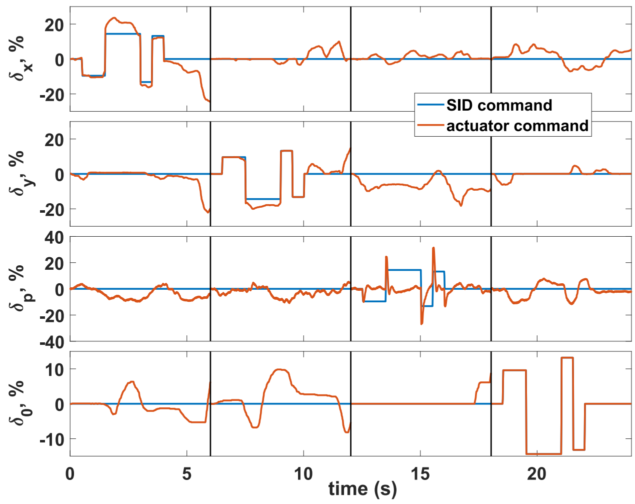


Figure 5: Commanded system identification controls and resulting actuator commands

GBN maneuvers, so that the use of the JIO method was not necessary for these tests.

Figure 7 shows the difference in the resulting frequency response for r/δ_0 when the response is either calculated directly or when using the JIO method. The direct calculation leads to smaller amplitude and very volatile phase values. For comparison, the figure also includes the frequency response that results from the maneuvers with GBN excitation. It can be seen that the frequency response generated from the sweeps using the JIO method compared very well to the frequency response from the GBN maneuvers.

3.3. Modeling

It had been planned to perform the system identification with the frequency response method based on the

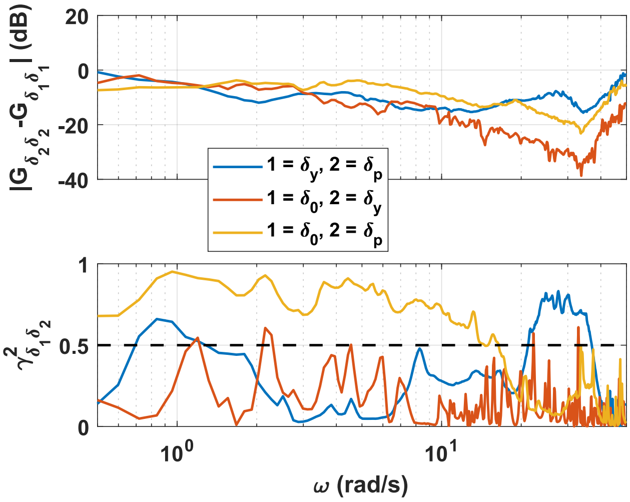


Figure 6: Cross correlation for the critical cases

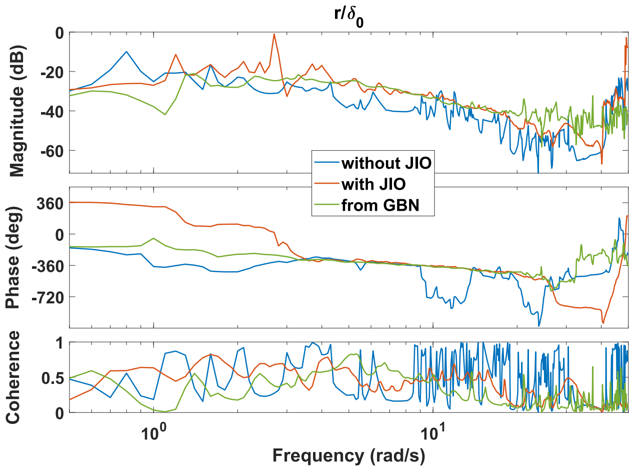


Figure 7: Comparison of frequency responses generated with and without JIO

generated frequency responses. As only frequency responses from GBN maneuvers were available in forward flight, the responses from the GBN maneuvers were also used in hover for comparability. As the frequency responses showed only small variation with speed, the same model structure could be used for hover and forward flight.

As the on-axis responses for pitch and roll rate (q/δ_x and p/δ_y) showed a phase drop of 40 dB/decade, second-order dynamics were required and thus flapping had to be accounted for. The on-axis response in yaw rate had a phase drop of 20 dB/decade so that the standard 6-Degrees of Freedom (DoF) formulation was sufficient. Accounting for inflow and coning was not necessary as the frequency response of vertical acceleration due to collective (a_z/δ_0) showed no rise in amplitude at higher frequencies.

Flapping was modeled using the explicit formulation from chapter 15 of [5] where the regressive flapping is modeled by two coupled equations for the flapping angles

$$
\begin{aligned}
\tau_f \dot{\beta}_{1s} &= -\beta_{1s} + \tau_f p + Lf_{\beta_{1c}}\beta_{1c} \\
&\quad + Lf_{\delta_x}\delta_x + Lf_{\delta_y}\delta_y \\
\tau_f \dot{\beta}_{1c} &= -\beta_{1c} + \tau_f q + Mf_{\beta_{1s}}\beta_{1s} \\
&\quad + Mf_{\delta_x}\delta_x + Mf_{\delta_y}\delta_y
\end{aligned} \tag{7}
$$

where β_{1c} and β_{1s} denote the longitudinal and lateral flapping of the tip-path plane.

The flapping angles are coupled to the rigid body via

$$
\begin{aligned}
\dot{u} &= X_u u + X_v v + X_w w - w_0 q + (X_r + v_0) r \\
&\quad - g\cos\theta_0 \theta + X_{\beta_{1c}}\beta_{1c} + X_{\delta_p}\delta_p + X_{\delta_0}\delta_0 \\
\dot{v} &= Y_u u + Y_v v + Y_w w + w_0 p + (Y_r - u_0) r \\
&\quad + g\cos\theta_0 \phi + Y_{\beta_{1s}}\beta_{1s} + Y_{\delta_p}\delta_p + Y_{\delta_0}\delta_0 \\
\dot{p} &= L_u u + L_v v + L_w w + L_r r + L_{\beta_{1s}}\beta_{1s} \\
&\quad + L_{\delta_p}\delta_p + L_{\delta_0}\delta_0 \\
\dot{q} &= M_u u + M_v v + M_w w + M_r r + M_{\beta_{1c}}\beta_{1c} \\
&\quad + M_{\delta_p}\delta_p + M_{\delta_0}\delta_0
\end{aligned} \tag{8}
$$

where the force stiffness terms are constrained to $X_{\beta_{1c}} = -Y_{\beta_{1s}} = g$.

The equations for vertical velocity and yaw rate are those of a 6-DOF model

$$
\begin{aligned}
\dot{w} &= Z_u u + Z_v v + Z_w w + Z_p p + (Z_q + u_0) q + Z_r r \\
&\quad - g\sin\theta_0 \theta + Z_{\delta_x}\delta_x + Z_{\delta_y}\delta_y + Z_{\delta_p}\delta_p + Z_{\delta_0}\delta_0 \\
\dot{r} &= N_u u + N_v v + N_w w + N_p p + N_q q + N_r r \\
&\quad + N_{\delta_x}\delta_x + N_{\delta_y}\delta_y + N_{\delta_p}\delta_p + N_{\delta_0}\delta_0
\end{aligned} \tag{9}
$$

First, the on-axis responses (p/δ_y, q/δ_x, r/δ_p and a_z/δ_y) were identified using this model structure. The resulting match is shown in Figure 8 for the hover case.

Further evaluation showed that the coherence of the cross-axis frequency responses was not sufficient for determining all important coupling derivatives. Thus, the identification approach was modified such that the on-axis parameters that had been identified in the frequency domain were fixed and the remaining parameters then identified in the time domain using the 2-3-1-1 multistep input maneuvers.

Analyzing the achieved match in the frequency domain showed that the phase drop in roll rate due to collective was too small. This deficit is usually an indication for unmodeled engine dynamics. Consequently, the model structure was enhanced in the yaw axis by a first-order Padé approximation of a time delay in collective to approximate engine dynamics as suggested in chapter 15 of [5]. Figure 9 shows the achieved match of the enhanced model in the time domain for the hover case. It can be seen that all coupling responses are modeled sufficiently.

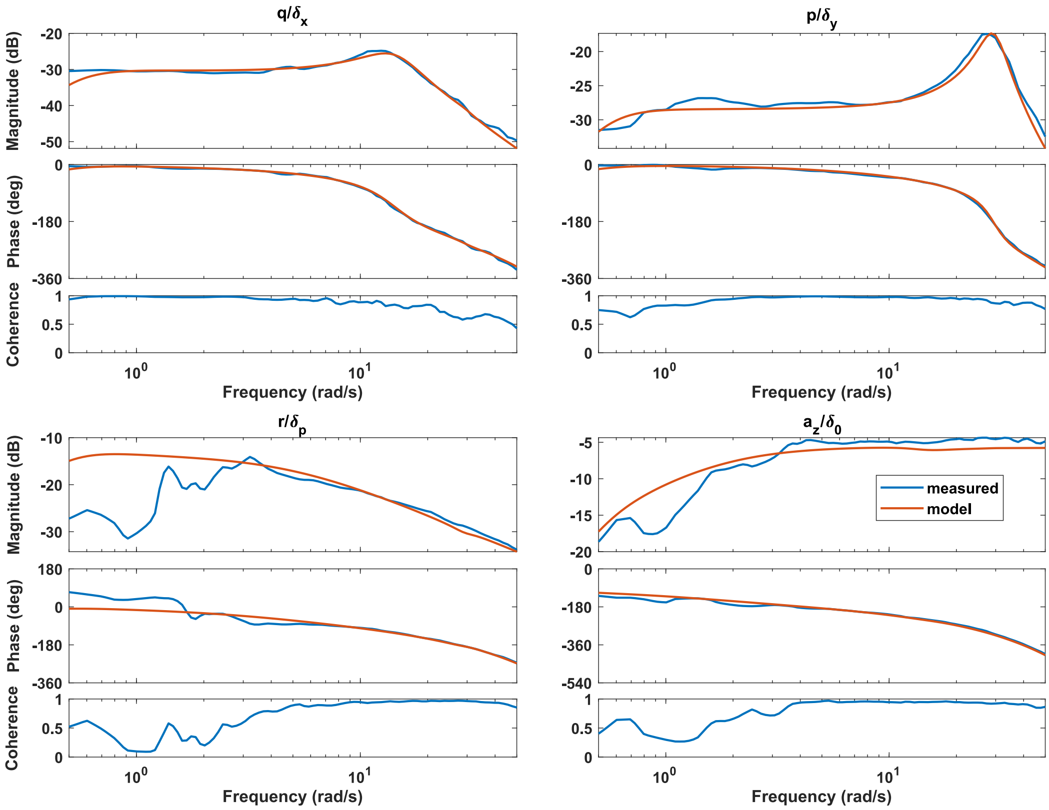


Figure 8: Achieved match for the on-axis frequency responses (hover)

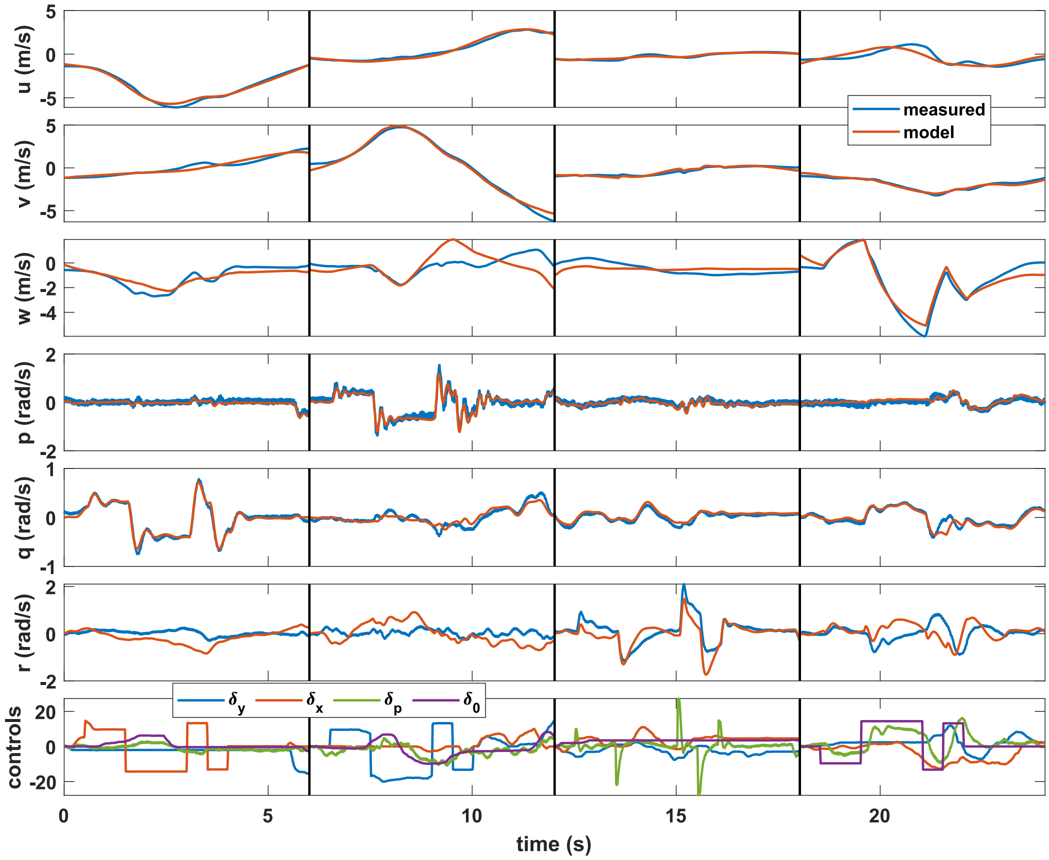


Figure 9: Achieved match in the time domain (hover)

4. CONTROLLER DESIGN

To improve the performance of the existing controller, a redesign of the inner control loop was performed based on the identified system models from section 3. The system identification resulted in a total of three linear system models each describing the dynamics of the MIDIARTIS at a different airspeed: at hover, 10 m/s and 20 m/s. The goal was to design a controller capable of achieving a higher performance while maintaining good robustness to sensor noise and model errors. For simplicity, no gain scheduling or similar techniques should be employed.

The improved inner-loop control structure consists of four controllers, each dedicated to one control axis. The four controls δ_x, δ_y, δ_p, and δ_0 are mapped to the variables θ, ϕ, ψ, and $\dot{h}$. The control structure for pitch and roll control is displayed in Figure 11a and consists of four key parts:

1. A command filter for generating reference commands,
2. A PID-feedback structure to handle control errors,
3. A feedforward transfer function to generate a nominal input and increase bandwidth/responsiveness,
4. A notch filter for cancelling the rotor flapping poles.

The command filter is a second-order filter with the (continuous) transfer function

$$\mathrm{CM}(s) = \frac{\omega_n^2}{s^2 + \omega_n \cdot \zeta \cdot s + \omega_n^2} \tag{10}$$

generating angle and angular velocity reference commands for the feedforward and feedback portion of the controller. The filter was implemented as a discrete-time filter. The damping of the second-order system was chosen to be ζ = 1. The natural frequency ω_n was set by the user during the numerical tuning phase, as a compromise between agility and available control authority. The natural frequency ω_n of the command filter is not tunable in the sense that the employed optimizer is not free to tune this variable, but has to be specified by the user before optimization. This process will be further described later in the paper. The chosen parameters are summarized in table 2.

Turn coordination reference values (ω_{TC}) originating from the outer loop, are added to the angular velocity reference values generated by the command filter before being fed to the feedforward filter and the PID feedback structure. The PID controller uses angular velocity proportional feedback, angle error proportional, and angle error integral feedback. For example, in the lateral axis, the PID controller takes the form

$$d_{y,fb} = k_P \cdot e_\phi + k_I \cdot \frac{T_s}{z-1} \cdot e_\phi + k_D \cdot e_p \tag{11}$$

with k_P, k_I, and k_D being the PID gains and T_s the sampling time of the controller. Furthermore, an anti-windup structure in the form of back-calculation is implemented to prevent integrator windup in case of actuator saturation [11, p. 83].

The feedforward filter is realized as a discrete transfer function with three tuning parameters a, b and c of the form

$$FF(z) = \frac{c \cdot z - a \cdot c}{z - b} \tag{12}$$

which is able to emulate a discrete low- or highpass filter.

Last but not least, notch or pole cancellation filters were employed for the roll and pitch axis in order to approximately cancel the weakly damped body-roll/rotor-flap and body-pitch/rotor-flap poles visible in the pole-zero map in Figure 10. These filters can be represented as

$$C(s) = \frac{(s+n)(s-n)}{(s-p)^2} \tag{13}$$

in continuous form with zeros n and poles p. This compensation is absolutely critical especially in roll, as otherwise the body-roll/rotor-flap pole would have a severe impact on the response of the controller leading to an undesirable weakly damped oscillation in the attitude response. Both notch filters, for roll and pitch, were manually designed to approximately cancel the coupled flapping poles. As can be seen in Figure 10, these poles hardly change with speed, so a single notch filter for each axis was sufficient. In order to achieve a causal filter, the poles p of the notch filters were chosen to be at –30. Notch filters are often used in such situation, e.g. in [12] where a notch filter was used to dampen resonance dynamics between rotor, stabilizer and fuselage.

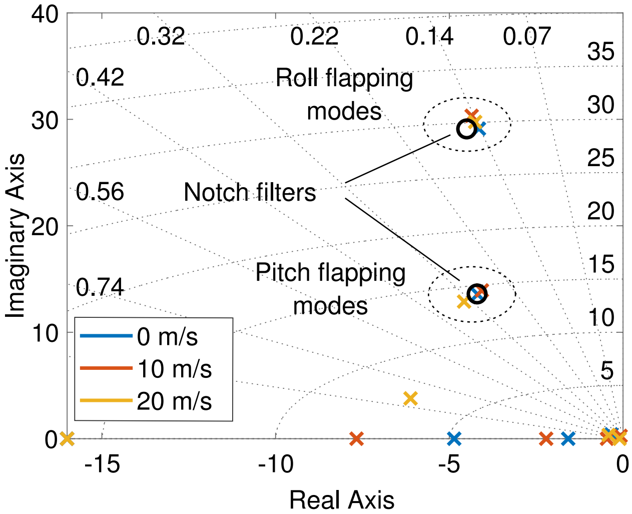


Figure 10: Pole-zero map of MIDIARTIS and position of notch filter

The control structure for the yaw controller is the same as for the roll and pitch controller, however a notch filter is not needed for this axis, as the yaw motion is not coupled with rotor flapping.

The heave control structure is displayed in Figure 11b and largely follows the same form as that of the angle

Table 1: Optimization goals for the controller tuning

Hard goals	
Phase margin	6 dB
Gain margin	45°
Sensitivity	HF: 5 dB LF: –21.2 dB
Max. gain of feedforward	29.5 dB

Soft goals	
Nominal step tracking	See table 2

controllers. A second-order command filter is employed to generate a reference rate-of-climb velocity command which is used in a feedback structure in the form of a PI controller as well as a feedforward filter of the same form as (12).

The resulting inner loop control structure contains a total of 23 parameters for the feedback and feedforward controllers of the four axes. These parameters were tuned using methods of robust control by utilizing the MATLAB function *systune* [13] to satisfy a number of design requirements (or goals) that are summarized in table 1. The hard goals for gain and phase margin as well as sensitivity shall ensure adequate robustness against modelling errors and sensor noise. Additionally, the feedforward transfer functions in (12) were constrained to contain less control authority than approximately 30% of control input for a step input of 1 rad.

Furthermore, a soft goal for step tracking was introduced which governed that the output of each target control variable (θ, ϕ, ψ and $\dot{h}$) follows the commanded signal as specified by a second-order system. Please note that the targeted response for each axis of the completed controller (set through the soft goal) was specified to be the *same* as that of the respective command filters. Therefore the same reference dynamic was used for the soft goal for step tracking as that for the command filter.

This approach allowed to iteratively tune the desired response time of the controller to balance available control authority and response quickness by simply increasing or decreasing the eigenfrequency of the second-order system governing *both*, the command filters and their respective soft tuning goal for step tracking. Thus, a compromise can be found by increasing/decreasing response time and therefore decreasing/increasing control authority until an optimum is reached. As the goal of this controller design was to enable maximum agility, the eigenfrequency was chosen as high as possible until all available control authority was used for a step input of around 30°. For the heave axis, an equally agile response was fine-tuned. The parameters of the various second-order systems used in the command filters (and therefore the desired step responses used as soft tuning goal) are shown in table 2.

Table 2: Parameters for command filters and nominal step tracking soft tuning goals

Axis	ω [Hz]	ζ [-]	t_{90} [s]
Roll	1.3	1	0.49
Pitch	1.3	1	0.49
Yaw	1.5	1	0.42
Heave	0.8	1	0.79

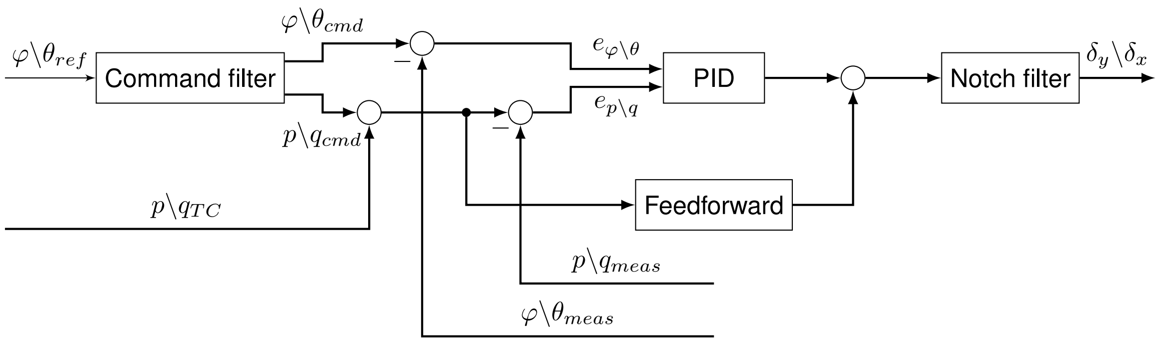
This architecture borrows on the Explicit Model Following (EMF) control concept e.g. of [14] which typically involves the inversion of the nominal system model for feedforward control. In this paper, the architecture is a simplified version of this concept by using axis-wise command models in conjunction with a first-order feedforward control filter. Due to the optimization-based approach for designing the various gains of this architecture, a single set of parameters is sufficient to stabilize the complete flight envelope while still delivering a high degree of performance.

The entire flight control system was implemented as a discrete control system in MATLAB/Simulink with a sampling rate of 100 Hz. In order to test the completed control system, a quasi-nonlinear stitched simulation framework [15] based on the identified linear models from section 3 was implemented. After being validated in this simulation, the controller was autocoded as a C/C++ module compatible with the Pixhawk PX4 autopilot system and uploaded to the flight controller hardware before being flight tested. The resulting controller validation is discussed in the following section.

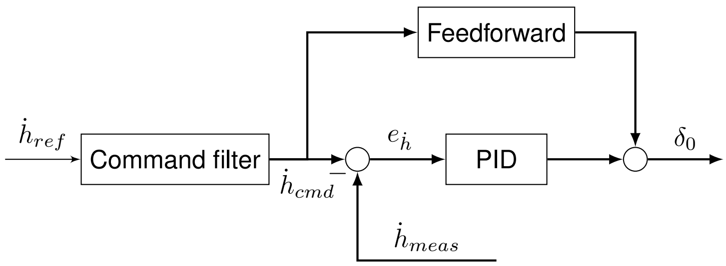
5. CONTROLLER VALIDATION

Flight tests were conducted to evaluate and validate the performance of the designed inner-loop flight controller and its impact on the outer-loop modes. The tests were conducted during a calm day with wind speed around 3 m/s to 4 m/s. The initial condition was a stabilized hover, achieved by manually commanding $\phi_{\mathrm{ref}} = \theta_{\mathrm{ref}} = 2^{\circ}$ which roughly correspond to the hover trim attitude. Subsequently, two doublet inputs in roll and pitch with an amplitude of $\pm 5°$ and $\pm 15°$, and a step duration of 1 s, were commanded. During the doublets, all axes of the controller were active. The resulting angles, rates and control inputs are shown in Figure 12a and 12b for the pitch and roll axis. Note that both the angle and rate measurements are subject to an oscillation stemming from the main rotor's 1/rev coupling into the fuselage with $\approx$21 Hz, which persists in the data. While a low-pass filter is implemented in the sensor feedback of the Pixhawk autopilot, it was not parameterized to filter out the 1/rev rotor oscillation.

As can be seen in these plots, the MIDIARTIS follows these step commands quickly and accurately. The high-gain feedback also guarantees low deviation from

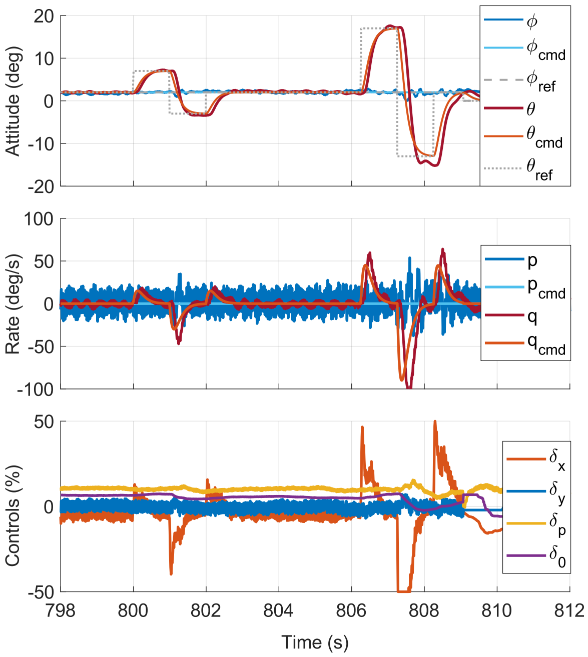


(a) Pitch and Roll controller

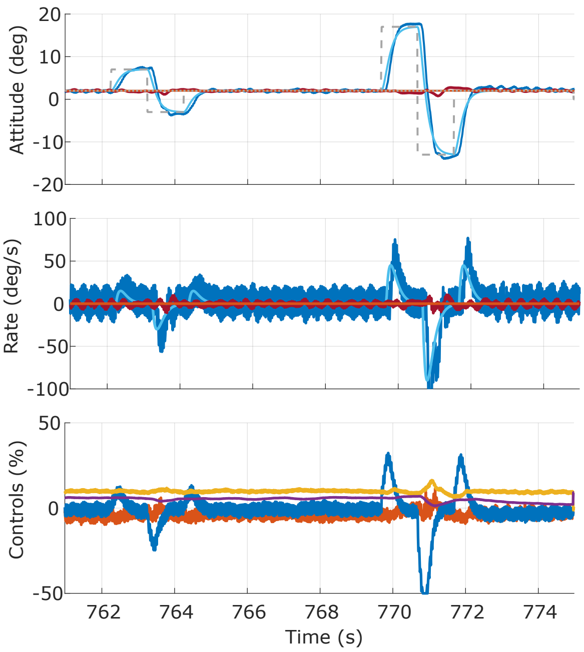


(b) Heave controller

Figure 11: Improved inner loop control structure



(a) Pitch command doublette



(b) Roll command doublette

Figure 12: Command doublette responses of the new inner-loop controller in hover condition

the setpoints with angle deviations well below $e_\theta\backslash e_\phi < \pm 0.5°$. The achieved rise time in the roll axis over all six steps is around $t_{90} \approx 0.47$ s and in the pitch axis $t_{90} \approx 0.52$ s, which is close to the designed values as outlined in table 2. Also visible is that in both axis, the input is completely saturated for a setpoint step of 30°, however the pitch axis is saturated slightly longer compared to the roll axis. Both of these results show that the MIDIARTIS is more agile in the roll axis than in the pitch axis, which is expected due to its higher inertia in that axis.

However, especially in the roll axis shown in Figure 12b some oscillation with a frequency of ≈ 4.5 Hz are still visible, indicating that the roll/flapping oscillation could not be entirely eliminated. This may stem from a improperly identified flapping eigenvalue, or more likely, from an improper pole-zero cancellation due to the approximate placement of the notch filter. While this slight roll oscillation would very likely be a problem for a manned rotorcraft, it is acceptable for this controller design.

The identified linear system models include cross coupling as indicated in eqs. (7) to (9), which are not accounted for in this control system architecture or controller design. This shortcoming shall be addressed in the future. Another noticeable effect is the actuator activity which shows the aforementioned 1/rev oscillation. This is certainly an undesirable effect which can lead to increased actuator wear and shall be eliminated in the future as well.

The improved design of the inner-loop flight controller also has significant impact on the performance of the outer-loop flight controllers. It could be shown that the improved response time of the inner loop allows retuning of the outer loop to significantly improve performance of both the position hold and trajectory-following loop. In this specific application, a hover hold accuracy of approximately 5 cm in x- and y-direction could be achieved during a flight tests, illustrating the improvement made with this controller.

6. CONCLUSIONS

The presented approach for improving a legacy flight controller of the MIDIARTIS UAV by employing system identification, proved to be highly successful.

1. Leveraging the legacy flight controller for system identification flight tests in conjunction with GBN system excitation inputs bears several advantages: Flight tests can be performed much faster, when compared to more traditional single-axis open-loop sweep inputs, as the GBN excitation simultaneously excites all axes which enables the execution of the complete system identification inputs in one long run.

2. Furthermore, by using the legacy controller to stabilize the aircraft during system identification, the influence of a human pilot is removed, resulting in a more accurate and repeatable behaviour of the vehicle, lowering workload, and further increasing efficiency of the flight test. It should be noticed, however, that the validation signals in form of modified 2-3-1-1 inputs were still flown manually. Because of the short duration of these signals, this was a small factor in overall flight test time, still demanding a high level of attention. In conclusion, it can be said that the GBN excitation proved to be easy to implement and highly efficient in flight testing.

3. After data collection and system identification, the resulting linear systems could be successfully employed for designing an improved inner-loop flight controller. The design mainly hinged on three elements: First, the notch filter for suppression of the flapping eigenmodes which enables a high-gain feedback without exciting the flapping dynamics. Second, a command filter generating references for feedforward and feedback controllers, and last, the numerical tuning method based on MATLAB's *systune* functionality.

4. The resulting inner loop was designed to achieve maximum possible agility given the available control authority by iteratively setting the eigenfrequency of the soft goal for step tracking and simultaneously for the command filters. This approach worked very well for this setup, resulting in fast response times to command inputs and good disturbance rejection and therefore enabling very good results for outer loop modes such as position hold or trajectory following and future expansions of these functionalities.

Author contact:
Philippe Petit - Philippe.Petit@dlr.de
Susanne Seher-Weiß - Susanne.Seher-Weiss@dlr.de
Johannes Wartmann - Johannes.Wartmann@dlr.de
Alexander Donkels - Alexander.Donkels@dlr.de

7. ACKNOWLEDGEMENTS

The authors would like to thank the Federal Office of Bundeswehr Equipment, Information Technology and In-Service Support (BAAINBw L1.1) as well as the the Federal Ministry of Economic Affairs and Climate Action of Germany through the Federal Aviation Research Program (LuFo) VI-2 for funding parts of this project.

Supported by:

Federal Ministry
for Economic Affairs
and Climate Action

on the basis of a decision
by the German Bundestag

8. REFERENCES

[1] Brian L. Stevens, Frank L. Lewis, and Eric N. Johnson. *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. John Wiley & Sons, 2015.

[2] Bernard Mettler, Mark B. Tischler, and Takeo Kanade. System Identification of Small-Size Unmanned Helicopter Dynamics. In *American He-*

licopter Society 55th Forum. Montreal, Quebec, Canada, 1999.

[3] Lorenz Meier, Dominik Honegger, and Marc Pollefeys. Px4: A Node-based Multithreaded Open Source Robotics Framework for Deeply Embedded Platforms. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6235–6240, 2015.

[4] R. K. Heffley, Lehmam J. M. Jewel, and R. A Von Winkle. A Compilation and Analysis of Helicopter Handling Qualities Data; Volume I: Data Compilation. *NASA CR 3144*, 1979.

[5] Mark B. Tischler and Robert K. Remple. *Aircraft and Rotorcraft System Identification: Engineering Methods with Flight-Test Examples*. American Institute of Aeronautics and Astronautics, Inc., second edition, 2012.

[6] Herbert J. A. F. Tulleken. Generalized Binary Noise Test-signal Concept for Improved Identification-experiment Design. *Automatica*, 26(1):37–49, January 1990.

[7] Dronecode Foundation. Qgroundcontrol, https://qgroundcontrol.com/, Accessed on 2025-04-06.

[8] Ravindra V. Jategaonkar. *Flight Vehicle System Identification—A Time Domain Methodology*. Progress in Astronautics and Aeronautics. Reston, VA, 2nd ed. edition, 2015.

[9] Johannes Wartmann, Susanne Seher-Weiß, Philippe Petit, Alexej Dikarew, Andreas Voigt, and Kristian K. Fettig. System Identification of a Compound Intermeshing Rotor UAV. In *Vertical Flight Society's 79th Annual Forum & Technology Display*, West Palm Beach, FL, USA, May 2023.

[10] Tom Berger, Mark B. Tischler, Marit E. Knapp, and Mark J. S. Lopez. Identification of Multi-Input Systems in the Presence of Highly Correlated Inputs. *Journal of Guidance, Control, and Dynamics*, 41(10):2247–2257, October 2018.

[11] Karl J. Åström and Tore Hägglund. *PID Controllers: Theory, Design, and Tuning*. ISA - The Instrumentation, Systems and Automation Society, 2nd edition, 1995.

[12] Bernard Mettler, Takeo Kanade, Mark B. Tischler, and William Messner. Attitude control optimization for a small-scale unmanned helicopter. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 4059, 2000.

[13] Pierre Apkarian. Tuning controllers against multiple design requirements. In *2012 16th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 1–6. IEEE, 2012.

[14] Mario Hamers, Robin Lantzsch, and Jens Wolfram. First Control System Evaluation of the Research Helicopter FHS. In *33rd European Rotorcraft Forum*, September 2007.

[15] Eric L. Tobias and Mark B. Tischler. A Model Stitching Architecture for Continuous Full Flight-envelope Simulation of Fixed-wing Aircraft and Rotorcraft from Discrete-point Linear Models. *US Army AMRDEC Special Report RDMR-AF-16-01*, 2016.