Flight code convergence: fixedwing, rotorcraft, hybrid

van Wijngaarden, D.C.; Smeur, E.J.J.; Remes, B.D.W.

Publication date
2021

Document Version
Final published version

Published in
12th International Micro Air Vehicle Conference

Flight Code Convergence: Fixedwing, Rotorcraft, Hybrid

D.C. van Wijngaarden*, E.J.J.Smeur, and B.W.D. Remes
Delft University of Technology, Kluyverweg 1, Delft, The Netherlands

ABSTRACT

Rotorcraft, fixed wing and hybrid Unmanned Air Vehicles (UAV) each have applications in which they excel. Traditionally, dedicated autopilot control code is written to accommodate flight of each UAV type. This causes fragmentation of control code and may lead to performance differences or errors. In this paper, we propose to use the same INDI controller for rotorcraft, fixed wing and hybrid UAVs, with only parametric differences in control effective matrix definitions and roll, pitch and airspeed limits. The controller is based on earlier work, but relevant derivations are included in this paper. Successful test flights, performed with a Bebop2 quadrotor, a Disco fixed wing, and a Nederdrone tailsitter hybrid demonstrate the feasibility of this approach.

1 INTRODUCTION

The amount of applications for Unmanned Air Vehicles (UAVs) has drastically increased over the last couple of years [1]. To best serve their purpose, different applications require different types of drones: fixed wing, rotorcraft, or hybrid. Fixed wing aircraft have superior endurance, while rotorcraft have more flexible maneuvering and hovering capabilities. Hybrid UAVs take the middle ground in terms of endurance and flexible maneuvering.

In the UAVs that are flown, one can broadly make the distinction between commercial UAVs that have proprietary software that is dedicated to one specific UAV type, and open source autopilot systems, that provide flight code for a variety of different UAV types. Examples of the latter are PX4 [2], Ardupilot [3], and Paparazzi [4]. These open source autopilot systems have found a broad user base with universities, amateur drone pilots and startup companies, and each of these autopilot systems supports various fixed wing, rotorcraft and hybrid drone types.

However, the control and guidance code for these different types of UAV, is typically separated. For example, the aforementioned autopilot systems contain dedicated control code for fixed wing, VTOL, and rotorcraft drones. Even the control of hybrid, or transitioning, drones are often done by switching from one controller to another as the vehicle transitions to forward flight [5, 6]. There have been researchers presenting an more integrated, or unified control structure for hybrid UAVs, making use of Nonlinear Dynamic Inversion (NDI) [7] or Incremental Nonlinear Dynamic Inversion (INDI) [8], but this was always aimed at one specific hybrid vehicle.

Figure 1: UAV platforms used for the experiments described in this paper. From left to right: Nederdrone (hybrid), Parrot Disco (fixed wing) and a Parrot Bebop2 (rotorcraft).

This fragmentation of code can result in implementation differences and therefore performance differences between the different UAV types. Next to that, keeping all control code up to date requires extra work in code maintenance, which also increases the chance of errors.

In this paper, we demonstrate that it is possible to fly fixed wing, rotorcraft, and hybrid UAVs using the same uniform INDI control and guidance algorithm, with only parametric changes in the control effectiveness matrix definitions and flight envelope protection concerning pitch, roll and airspeed limits. This is achieved through a cascaded INDI controller, based on [9] and [10], that controls attitude and position through closed loop control of angular and linear accelerations. Different vehicle configurations are accommodated through parametric changes of (1) the control effectiveness and (2) flight envelope limits. The flight code used for these experiments is publicly available on Github [1].

In the following sections, a universal controller is outlined that is applied to a Bebop2 quadrotor (rotorcraft), a Disco fixed wing and a Nederdrone (hybrid UAVs). These platforms are described in Section 2. In Section 3 the inner (attitude) controller is explained. Section 4 describes the

*Email address(es): D.C.vanWijngaarden@tudelft.nl

[1] https://github.com/tudelft/paparazzi/tree/convergence

outer loop (position) controller, and highlights the parametric differences required for the different UAV types. Section 5 presents test flights of a rotorcraft, fixed wing and hybrid UAV, using the same control code. Conclusions are drawn in Section 6.

2 TEST PLATFORMS

Though the controller described in this paper is applicable to a broad range of rotorcraft, fixed wing and hybrid UAVs, we will consider three platforms in particular. These are the Parrot Bebop2 quadrotor, the Parrot Disco fixed wing, and the Nederdrone developed by the MAVLab in Delft [11], all depicted in figure 1. The left UAV in the figure is the Nederdrone, a hybrid UAV that can hover, can transition 90 degrees to fly forward horizontally and take-off and land vertically. The Nederdrone is a biplane tail-sitter with 12 motors and 8 control surfaces distributed over its wings.

The middle UAV is the Parrot Disco, a fixed wing UAV that can only fly forward and does not have hovering capabilities. The parrot Disco is a flying wing that has a single motor and a pair of elevons. These control surfaces can be used for pitch and roll control. The vehicle is passively directionally stable.

Finally, the right UAV is a Parrot Bebop2, a regular quadrotor without any wing surface. The Bebop2 has 4 motors that can provide control inputs for pitch, roll and yaw. The Bebop2 does not have an airspeed sensor, as opposed to the other two vehicles, which use it while flying "forward".

For all vehicles, the body reference frame is defined with the Z axis in the opposite direction as the thrust, the Y axis pointing to the right and the X axis completing the right handed axis system (for the fixed wing orthogonal to the wing surface, for the quadrotor through the nose). The body reference frame is illustrated by figure 2.

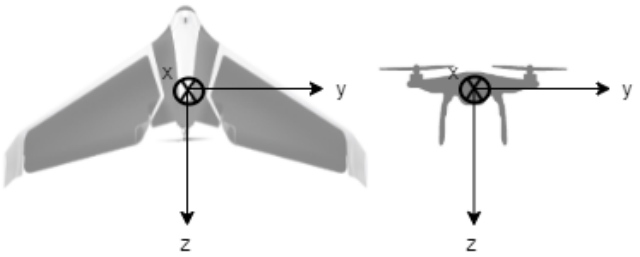

Figure 2: Body reference frame for a fixed wing and a multirotor.

3 INDI INNER LOOP CONTROL

The inner loop controller is implemented along the lines of [12], but without the online control effectiveness estimation. For details, we refer to that paper, but for completeness, the controller will be summarized in this section.

The model on which the controller is based is given by the equation for the translational dynamics:

$$\ddot{\boldsymbol{\xi}} = \boldsymbol{g} + m^{-1}\left(\boldsymbol{M}_{NB}\boldsymbol{f} + \boldsymbol{f}_{\text{ext}}\right), \tag{1}$$

where $\boldsymbol{\xi}$ is the position vector in the North East Down (NED) frame, $\boldsymbol{g}$ is the gravity vector in the NED frame and m is the mass of the drone. $\boldsymbol{M}_{NB}$ is the rotation matrix from body to NED frame, which is obtained from the attitude quaternion $\boldsymbol{q}$. $\boldsymbol{f}$ is the input force due to changes in attitude and thrust level and $\boldsymbol{f}_{\text{ext}}$ is an unmodeled external force.

The rotational dynamics are given by:

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \begin{pmatrix} 0 \\ \boldsymbol{\Omega} \end{pmatrix}, \tag{2}$$

$$\dot{\boldsymbol{\Omega}} = \boldsymbol{J}^{-1}(\boldsymbol{m} + \boldsymbol{m}_{\text{ext}} - \boldsymbol{\Omega} \times \boldsymbol{J}\boldsymbol{\Omega}), \tag{3}$$

where $\boldsymbol{\Omega}$ denotes the angular rates of the vehicle and $\otimes$ the Hamilton product. The inertia matrix of the vehicle is denoted by $\boldsymbol{J}$ and is assumed to be diagonal, $\boldsymbol{m}$ is the moment due to the inputs and $\boldsymbol{m}_{\text{ext}}$ is the moment due to unmodeled external moments.

Neglecting the cross-term in Eq. 3, we can approximate the change in angular acceleration of the vehicle due to a change in the input vector $\boldsymbol{u}$ as:

$$\dot{\boldsymbol{\Omega}} - \dot{\boldsymbol{\Omega}}_0 = \boldsymbol{G}_1(\boldsymbol{u} - \boldsymbol{u}_0) + \boldsymbol{G}_2(\dot{\boldsymbol{u}} - \dot{\boldsymbol{u}}_0), \tag{4}$$

where $\boldsymbol{G}_1$ is the control effectiveness matrix (which incorporates the inertia and may also be a function of velocity), and $\boldsymbol{G}_2$ is a control effectiveness matrix that can account for effectiveness in the derivative of $\boldsymbol{u}$, which can occur due to significant propeller inertia. The current angular acceleration $\dot{\boldsymbol{\Omega}}_0$ can be obtained by differentiation of the gyroscope signal. As differentiation will amplify the high frequency vibrations, a low-pass filter has to be applied to all signals that have the subscript 0, which will now receive the subscript f. As $\boldsymbol{u}$ and $\dot{\boldsymbol{u}}$ are inter-dependent, we approximate the derivative in discrete time using the unit delay operator L as $\dot{\boldsymbol{u}} \approx (\boldsymbol{u}(k) - \boldsymbol{u}(k-1))/T_s = (\boldsymbol{u} - L\boldsymbol{u})/T_s$, where T_s is the sample time. This leads to the prediction of $\dot{\boldsymbol{\Omega}}$ based on a new input command u_c:

$$\dot{\boldsymbol{\Omega}} - \dot{\boldsymbol{\Omega}}_f = (\boldsymbol{G}_1 + \boldsymbol{G}_2)(\boldsymbol{u}_c - \boldsymbol{u}_f) + \boldsymbol{G}_2 L(\boldsymbol{u}_c - \boldsymbol{u}_f). \tag{5}$$

This system can be inverted using the pseudo inverse, denoted by $(.)^+$, or by a more sophisticated control allocation scheme, such as Weighted Least Squares (WLS) control allocation [13]. Since this is now a control law to which the angular acceleration is an input, it is denoted as the virtual control $\boldsymbol{\nu}$.

$$\boldsymbol{u}_c = \boldsymbol{u}_f + (\boldsymbol{G}_1 + \boldsymbol{G}_2)^+(\boldsymbol{\nu} - \dot{\boldsymbol{\Omega}}_f + \boldsymbol{G}_2 z^{-1}(\boldsymbol{u}_c - \boldsymbol{u}_f)) \tag{6}$$

Due to the feedback of the angular acceleration, using the control effectiveness of the actuators, disturbances can be counteracted quickly [12, 9].

Since the INDI controller takes care of most of the non-linearities in the system, the reference angular acceleration $\boldsymbol{\nu}$ can simply be obtained through a PD controller. The attitude is represented as a quaternion, and the vector part of the quaternion error with the desired attitude is used for feedback:

$$\boldsymbol{\nu} = K_D(K_P \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix}_e - \boldsymbol{\Omega}), \tag{7}$$

where $\begin{bmatrix} q_x & q_y & q_z \end{bmatrix}_e^T$ is the vector part of the attitude quaternion error.

The gains K_D and K_P can be tuned or derived analytically [9].

There is a difference between the platforms in terms of their inner loop control. The Disco fixed wing UAV cannot fully control its attitude, as it does not have a rudder, but it still uses the same complete attitude controller. To make sure the reference attitude does not drift away from the real attitude in terms of its heading, the heading of the reference is continuously reset to its current heading in the outer loop. This is only done for the Disco, and not for the other UAVs. This would not be necessary for a fixed wing with yaw control.

Besides this, the differences are purely parametric: each vehicle has its own control effectiveness matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$. For the Disco and the Nederdrone, these matrices are a function of airspeed, as the airflow over the control surfaces greatly affects their effectiveness. For the Bebop2, these control effectiveness matrices are static.

4 INDI OUTER LOOP CONTROL

Though it may be apparent that the inner loop of the three types of UAV can be the same, since they all use their actuators to generate moments through which the attitude can be controlled, this is not obvious for the outer loop. A fixed wing UAV uses its wing to generate lift, and has to manipulate the angle of attack and the bank angle in order to maneuver, and the thrust in order to accelerate in the direction of flight. A rotorcraft on the other hand, can manipulate the amount of lift directly with its propellers, and has to tilt this lift vector in the desired direction of acceleration in order to maneuver.

Yet, hybrid (tailsitter) UAVs have demonstrated that it is possible to combine these different ways of flying in one vehicle, using one unified controller. Using appropriate flight envelope protections, it is reasonable to assume this controller can also be applied to pure rotorcraft or fixed wing UAV.

This is achieved using the guidance algorithm as presented in [10]. For completeness, a slightly shortened derivation will be given here. We will make use of the attitude representation in Euler angles, with the ZXY rotation order ($\boldsymbol{\eta} = \begin{bmatrix} \psi & \phi & \theta \end{bmatrix}$) such that the Euler angle derivatives are well defined at -90 degrees pitch. Then, the linear acceleration is given by:

$$\ddot{\boldsymbol{\xi}} = \boldsymbol{g} + \frac{1}{m}\boldsymbol{L}_N(\boldsymbol{\eta}, V) + \frac{1}{m}\boldsymbol{D}_N(\boldsymbol{\eta}, V) + \frac{1}{m}\boldsymbol{T}_N(\boldsymbol{\eta}, T), \tag{8}$$

where $\boldsymbol{L}_N$, $\boldsymbol{D}_N$, $\boldsymbol{T}_N$ are the lift, drag and thrust in the NED frame (denoted with the subscript N). Using the transformation matrix between the body and NED reference frames:

$$\boldsymbol{M}_{NB} = \begin{bmatrix} \mathrm{c}\theta\mathrm{c}\psi - \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi & -\mathrm{c}\phi\mathrm{s}\psi & \mathrm{s}\theta\mathrm{c}\psi + \mathrm{s}\phi\mathrm{c}\theta\mathrm{s}\psi \\ \mathrm{c}\theta\mathrm{s}\psi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{c}\psi & \mathrm{c}\phi\mathrm{c}\psi & \mathrm{s}\theta\mathrm{s}\psi - \mathrm{s}\phi\mathrm{c}\theta\mathrm{c}\psi \\ -\mathrm{c}\phi\mathrm{s}\theta & \mathrm{s}\phi & \mathrm{c}\phi\mathrm{c}\theta \end{bmatrix}, \tag{9}$$

the thrust can be written as:

$$\boldsymbol{T}_N = \boldsymbol{M}_{NB}\begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} = \begin{bmatrix} (\mathrm{s}\theta\mathrm{c}\psi + \mathrm{s}\phi\mathrm{c}\theta\mathrm{s}\psi)T \\ (\mathrm{s}\theta\mathrm{s}\psi - \mathrm{s}\phi\mathrm{c}\theta\mathrm{c}\psi)T \\ \mathrm{c}\phi\mathrm{c}\theta T \end{bmatrix}, \tag{10}$$

and the lift as:

$$\boldsymbol{L}_N = \boldsymbol{M}_{NB}^{\theta=-\frac{\pi}{2}}\boldsymbol{L}_B(\theta, V) = \begin{bmatrix} \mathrm{s}\phi\mathrm{s}\psi L(\theta, V) \\ -\mathrm{s}\phi\mathrm{c}\psi L(\theta, V) \\ \mathrm{c}\phi L(\theta, V) \end{bmatrix} \tag{11}$$

assuming that the flight path angle is small, such that the lift vector is only rotated from the vertical by the bank angle. Note that T and $L(\theta, V)$ will typically be negative, since the body Z axis points down.

Using a first order Taylor expansion, by taking partial derivatives with respect to the controlled input variables ($\boldsymbol{v} = \begin{bmatrix} \phi & \theta & T \end{bmatrix}^T$), we can arrive at the incremental model:

$$\ddot{\boldsymbol{\xi}} = \ddot{\boldsymbol{\xi}}_0 + \frac{1}{m}\left(\boldsymbol{G}_T(\boldsymbol{\eta}, T) + \boldsymbol{G}_L(\boldsymbol{\eta}, V)\right)(\boldsymbol{v} - \boldsymbol{v}_0), \tag{12}$$

where we have assumed that the drag changes slowly with respect to the other variables such that its influence on the change in acceleration can be neglected, and where the control effectiveness matrices of thrust and lift are given by:

$$\boldsymbol{G}_T(\boldsymbol{\eta}, T) = \begin{bmatrix} \left(\frac{\partial}{\partial\phi}\frac{1}{m}\boldsymbol{T}_N(\phi, \theta_0, \psi_0, T_0)|_{\phi=\phi_0}\right)^T \\ \left(\frac{\partial}{\partial\theta}\frac{1}{m}\boldsymbol{T}_N(\phi_0, \theta, \psi_0, T_0)|_{\theta=\theta_0}\right)^T \\ \left(\frac{\partial}{\partial T}\frac{1}{m}\boldsymbol{T}_N(\phi_0, \theta_0, \psi_0, T)|_{T=T_0}\right)^T \end{bmatrix}^T \tag{13}$$

and

$$\boldsymbol{G}_L(\boldsymbol{\eta}, V) = \begin{bmatrix} \left(\frac{\partial}{\partial\phi}\frac{1}{m}\boldsymbol{L}_N(\phi, \theta_0, \psi_0, V_0)|_{\phi=\phi_0}\right)^T \\ \left(\frac{\partial}{\partial\theta}\frac{1}{m}\boldsymbol{L}_N(\phi_0, \theta, \psi_0, V_0)|_{\theta=\theta_0}\right)^T \\ (\boldsymbol{0})^T \end{bmatrix}^T. \tag{14}$$

Elaborating these control effectiveness functions gives:

$$\boldsymbol{G}_T(\boldsymbol{\eta}, T) = \begin{bmatrix} \mathrm{c}\phi\mathrm{c}\theta\mathrm{s}\psi T & (\mathrm{c}\theta\mathrm{c}\psi - \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi)T & \mathrm{s}\theta\mathrm{c}\psi + \mathrm{s}\phi\mathrm{c}\theta\mathrm{s}\psi \\ -\mathrm{c}\phi\mathrm{c}\theta\mathrm{c}\psi T & (\mathrm{c}\theta\mathrm{s}\psi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{c}\psi)T & \mathrm{s}\theta\mathrm{s}\psi - \mathrm{s}\phi\mathrm{c}\theta\mathrm{c}\psi \\ -\mathrm{s}\phi\mathrm{c}\theta T & -\mathrm{c}\phi\mathrm{s}\theta T & \mathrm{c}\phi\mathrm{c}\theta \end{bmatrix} \tag{15}$$

and

$$\boldsymbol{G}_L(\boldsymbol{\eta}, V) = \begin{bmatrix} \mathrm{c}\phi\mathrm{s}\psi L(\theta, V) & \mathrm{s}\phi\mathrm{s}\psi\frac{\partial}{\partial\theta}L(\theta, V) & 0 \\ -\mathrm{c}\phi\mathrm{c}\psi L(\theta, V) & -\mathrm{s}\phi\mathrm{c}\psi\frac{\partial}{\partial\theta}L(\theta, V) & 0 \\ -\mathrm{s}\phi L(\theta, V) & \mathrm{c}\phi\frac{\partial}{\partial\theta}L(\theta, V) & 0 \end{bmatrix}. \tag{16}$$

$\ddot{\boldsymbol{\xi}}_0$ is measured by the accelerometer, but this sensor also picks up a vibrations in the airframe, which has to be removed with a low-pass filter. Like before, to keep all signals synchronized with the same phase delay, all terms with subscript zero will be filtered and receive subscript f instead. The equation can then be inverted to obtain:

$$\boldsymbol{v} = \boldsymbol{v}_f + m\left(\boldsymbol{G}_T(\boldsymbol{\eta}, T) + \boldsymbol{G}_L(\boldsymbol{\eta}, V)\right)^{-1}\left(\ddot{\boldsymbol{\xi}}_{\text{ref}} - \ddot{\boldsymbol{\xi}}_f\right) \quad (17)$$

where $\ddot{\boldsymbol{\xi}}_{\text{ref}}$ is the reference acceleration to track.

The functions $L(\theta, V)$ and $\frac{\partial}{\partial\theta}L(\theta, V)$ still have to be defined for the vehicles with a wing, for a pure rotorcraft they are zero. Recognizing that the lift only needs to be known to compute the effectiveness of rolling, we assume level flight and a simple relationship with the pitch angle:

$$L(\theta, V) \approx L(\theta) = -9.81\sin(-\theta)m \quad (18)$$

where θ is bounded between $-\pi/2$ and 0.

Similarly, we assume that in forward flight the thrust just compensates the drag, and its effect on accelerations other than in the thrust axis is small, such that for T in Eq. 12 we can write:

$$T(\theta) = -9.81\cos(\theta)m \quad (19)$$

where again θ is bounded between $-\pi/2$ and 0.

Finally, the rate of change of the reference ψ is computed in order to reduce sideslip β, using feed forward and feed back:

$$\dot{\psi}_{\text{ref}} = \frac{g\tan(\phi_t)}{V_l} + K_\beta\beta \quad (20)$$

where $\dot{\psi}_{\text{ref}}$ is the rate of change of the heading reference, g is the gravitational constant and V_l is a limited airspeed, with 10 m/s as a lower limit, to avoid unachievable rotations. This term is not relevant for the quadrotor, as sideslip is not detrimental to its lift generation. Still, it can be kept in as it will also not degrade the flight performance of the quadrotor.

Equation 17 provides a control law for the linear accelerations. The reference can again be obtained from a simple linear PD controller, if the goal is to hover at a waypoint. In order to track a certain trajectory, appropriate reference accelerations can be computed. In this paper, additionally the line tracking velocity vector field [10] was used together with a proportional ground velocity controller.

Though the assumptions in this section are quite crude, and the control effectiveness is probably inaccurate, the controller is still able to track linear accelerations and execute a simple flight plan. One can imagine that with a more accurate control effectiveness, the performance may improve. Important to observe is that the INDI controller provides an abstraction layer: a simple linear controller that outputs an acceleration reference can be used on top of it, that does not need to know about the flight mechanism of the vehicle. The changing control effectiveness matrices will account for the different methods of manipulating the acceleration of the vehicle.

4.1 Flight envelope limits

Another aspect in which the three types of UAVs are clearly different is their flight envelope. The rotorcraft should not pitch or roll too much, as that will render the vertical component of the thrust too small to support the weight of the drone. The fixed wing should not fly slower than its minimum airspeed, as that will stall the wing, and the reduced dynamic pressure will render the control surfaces ineffective.

To make sure the different UAVs don't exit their respective flight envelopes, different limits are imposed on the controlled flight states of the outer loop INDI controller. These are considered parametric differences.

Limit	Fixed wing	Hybrid	Rotorcraft
Pitch [deg]	[-115,-75]	[-120,25]	[-35,35]
Roll [deg]	[-45,45]	[-30,30]	[-35,35]
Airspeed [m/s]	>10	-	-

Table 1: Flight envelope limits in the outer loop INDI controller.

5 FLIGHT TEST RESULTS

Flight tests have been performed for three types of UAVs discussed by this paper. All flights have been performed with the same type of INDI control method. Each UAV has flown a route over line segments connecting 4 waypoints. For the rotorcraft and hybrid UAV, some static waypoints have been added to the mission to demonstrate hover capabilities of those platforms. Furthermore, those static waypoints force the hybrid UAV to transition from forward to hover flight and vice versa.

5.1 Horizontal guidance

Figure 3 depicts the two dimensional top view of the path flown by the Parrot Disco running the INDI outer loop guidance code. The flight path for this flight is directed in clockwise direction. The target airspeed for this flight was set to 12 m/s to prevent the UAV from stalling. The Parrot Disco starts its turn to the next line segment 50 metres before reaching a waypoint as can be seen in the diagram. This parameter was programmed such that the UAV can join its next line segment. It can be seen that the fixed wing UAV platform manages to follow its target lines during its route.

A top view of the path flown by the Parrot Bebop2 quadrotor is depicted in figure 4. It can be seen that target lines and target positions have been defined for this flight. The target lines are being followed during a route whereas target positions are approached, after which the vehicle hovers at position. The direction of flight over the target lines is in clockwise direction. It can be seen that when the UAV approaches a line segment from a target position that it first flies perpendicular to the line segment after which it joins and follows the line segment.

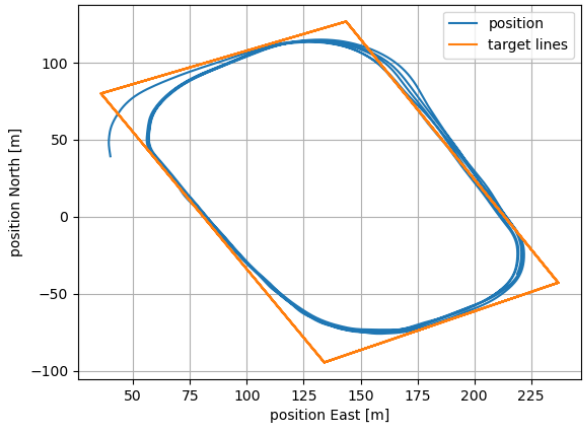

Figure 3: Horizontal position and target lines for the Parrot Disco.

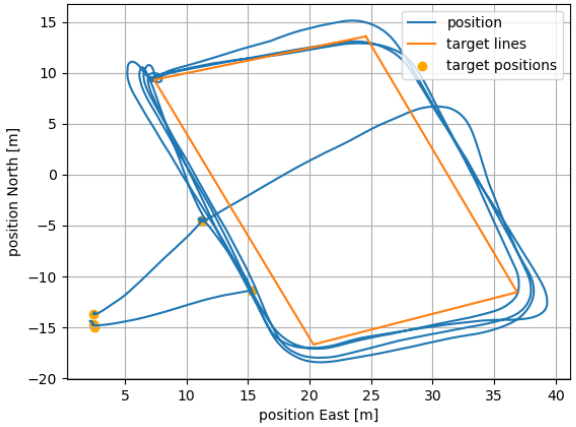

Figure 4: Horizontal position, target lines and target positions for the Parrot Bebop2.

A two dimensional view of the horizontal path flown by the Nederdrone, the hybrid UAV, is given by figure 5. It can can be seen that a mission consisting of target lines and target positions has been flown by this UAV. The flight path connected by the target lines is defined in counterclockwise direction. It can be seen that this platform does not track its lines with the same accuracy as the Disco in forward flight due to the smaller flight envelope in terms of roll angle and a higher airspeed target of 15 m/s in forward flight. However the line tracking accuracy in forward flight is lower for this UAV, the platform has the ability to track target positions with better precision in hover flight as can be seen in the diagram.

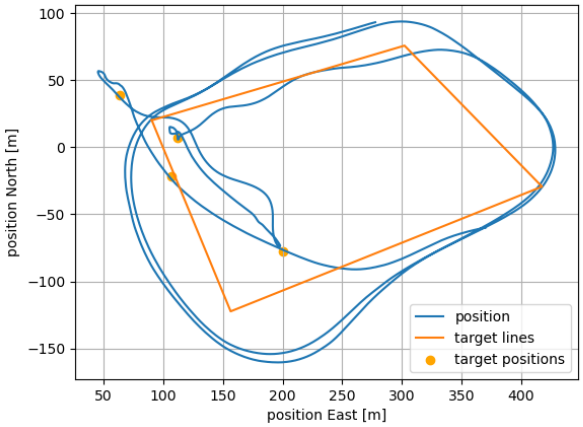

Figure 5: Horizontal position and target lines for the Nederdrone. Includes parts in forward flight, hovering, and complex twisting tailwind transition maneuvers.

5.2 *Vertical guidance*

The test results of the vertical guidance for each type of UAV can be evaluated through an altitude versus time plot as given by figure 6. This plot reflects the altitude and target altitude for each UAV over time.

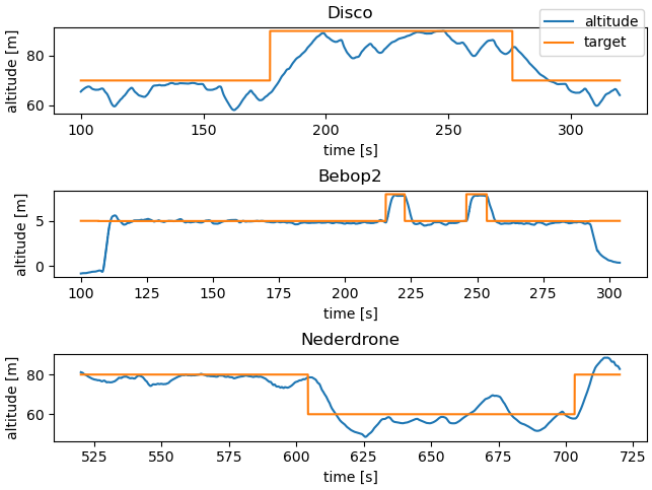

Figure 6: Altitude and target altitude versus time for the Parrot Disco, Parrot Bebop 2 and the Nederdrone.

First of all, it can be seen that the Parrot Disco has a climb and descend in its mission in order to evaluate vertical guidance commanded by the INDI outer loop control. An target altitude is set from 70 metres to 90 metres and back to 70 metres during the route. It is notified that there are some drops in altitude below the target. This occurs during turns which are not compensated for sideslip due to the lack of yaw control for this drone. The drop of altitude is approximately 10

metres below its target.

Secondly, the mission of the Parrot Bebop2 has two climb and descend step inputs between 5 and 7 metres altitude while following a line segment. It can be seen that the tracking of altitude for this platform is more accurate than for a fixed wing due to the direct control of lift by the propellers instead of a lift surface.

Finally, the Nederdrone has a descend and climb step in its mission between 80 and 60 metres altitude. Those climb and descend phases are carried out in forward flight. There is a drop in altitude below its target at 625 seconds. This is at the moment the drone transitions from forward flight to hover flight for which the production of lift is gradually exchanged from the wings to the propellers. The opposite is notified at 675 seconds then the UAV transitions from hover to forward flight.

5.3 Flight states

Other flight states that have been logged are the roll angle ϕ and the pitch angle θ using an Euler ZXY rotation order. The airspeed has been logged for the Parrot Disco and the Nederdrone which is used for forward flight.

Diagrams reflecting the pitch and roll angles for all three UAVs are given by figures 7 and 8 respectively. The flight envelope limits per platform as given by table 1 are visualized by dotted lines in those diagrams.

It can be seen that the pitch angle θ stays between the set limits for all platforms. It is notified that the Parrot Disco stays just below its maximum pitch angle limit to prevent the UAV from stalling. The transition of the Nederdrone from forward to hover flight that is initiated at 625 seconds can be seen in the pitch angle plot by an increase in pitch angle. At 710 seconds, the Nederdrone transitions back from hover to forward flight for which the pitch angle is decreased again.

The roll angle plots gives proof that the roll angle ϕ stays within the preset limits for the Parrot Disco and the Parrot Bebop2. The Nederdrone slightly exceeds its bank limit when making turns.

Finally, the airspeed over time for the Parrot Disco and Nederdrone are plotted in figure 9. The target air speeds for forward flight are set to 12 m/s and 15 m/s for the Parrot Disco and Nederdrone respectively.

It can be seen that there are some fluctuations in airspeed for the Disco around its target. This occurs during turns in which sideslip is not compensated due to lack of yaw control for this platform. Therefore the following two effects play a roll that cause fluctuations in airspeed: the airspeed sensor is not aligned with the direction of flight and the drag induced by sideslip affects the airspeed.

It can be seen for the Nederdrone that the target airspeed of 15 m/s is being tracked in forward flight. At the transition around 625 seconds it is notified that the airspeed drops to 0. The hover target position is moved to another place around 675 seconds for which a non zero airspeed is visible on the plot as pitch is being reduced to fly towards the moved location of the target position.

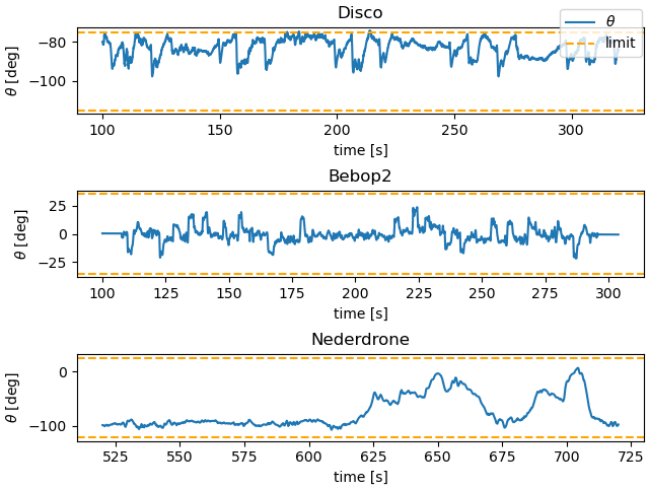

Figure 7: θ (Euler ZXY) versus time for the Parrot Disco, Parrot Bebop2 and the Nederdrone. Dotted lines visualizes the flightenvelope limits.

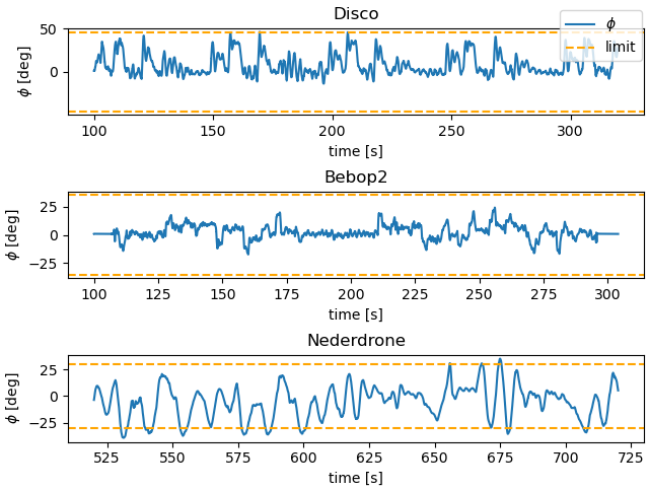

Figure 8: ϕ (Euler ZXY) versus time for the Parrot Disco, Parrot Bebop 2 and the Nederdrone. Dotted lines visualizes the flightenvelope limits.

6 CONCLUSIONS

This paper described a cascaded INDI inner and outer loop controller, that is applicable to rotorcraft, fixed wing and hybrid UAVs with only parametric differences. From the successful test flights, we conclude that it is indeed possible to use the same controller for these different UAVs. Parametric flight envelope limits prove to be a simple and effective way of preventing the controller from exiting the flight envelope.

The integrated sideslip controller is appropriate for hybrids and fixed wings, but is not required for multirotors,

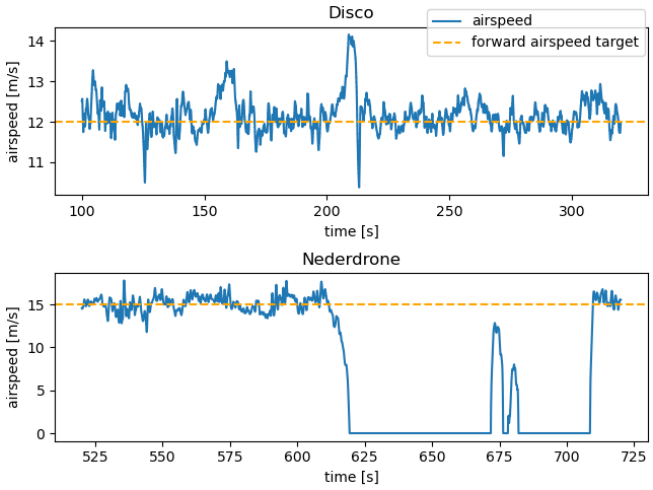

Figure 9: Airspeed versus time for the Parrot Disco and Nederdrone.

which may be constrained in their heading based on the application. This could be accommodated in the future by making this functionality modular. In the future we will show the implementation of this same controller in even more platforms like quad-planes, helicopters and tailed fixed wings. Showing the robustness of this controllers and leading to a single code stack for all types of UAV platforms.

REFERENCES

[1] Hazim Shakhatreh, Ahmad H. Sawalmeh, Ala Al-Fuqaha, Zuochao Dou, Eyad Almaita, Issa Khalil, Noor Shamsiah Othman, Abdallah Khreishah, and Mohsen Guizani. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access*, 7:48572–48634, 2019.

[2] 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. IEEE, 2015.

[3] Ardupilot. http://www.ardupilot.org/. Accessed: 2021-07-10.

[4] Gautier Hattenberger, Murat Bronz, and Michel Gorraz. Using the Paparazzi UAV System for Scientific Research. In *International Micro Air Vehicle Conference and Competition (IMAV)*, pages 247–252, 2014.

[5] C De Wagter, D Dokter, G de Croon, and B Remes. Multi-Lifting-Device UAV Autonomous Flight at Any Transition Percentage. In *EuroGNC 2013*, pages 1190–1204, 2013.

[6] Boyang Li, Jingxuan Sun, Weifeng Zhou, Chih-Yung Wen, Kin Huat Low, and Chih-Keng Chen. Transition Optimization for a VTOL Tail-sitter UAV. *IEEE/ASME Transactions on Mechatronics*, 2020.

[7] Philipp Hartmann, Carsten Meyer, and Dieter Moormann. Unified velocity control and flight state transition of unmanned tilt-wing aircraft. *Journal of Guidance, Control, and Dynamics*, 40(6):1348–1359, 2017.

[8] Stefan A. Raab, Jiannan Zhang, Pranav Bhardwaj, and Florian Holzapfel. Proposal of a Unified Control Strategy for Vertical Take-off and Landing Transition Aircraft Configurations. In *2018 Applied Aerodynamics Conference*, Reston, Virginia, jun 2018. American Institute of Aeronautics and Astronautics.

[9] Ewoud J. J. Smeur, G.C.H.E. de Croon, and Q. Chu. Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection. *Control Engineering Practice*, 73:79–90, apr 2018.

[10] Ewoud J. J. Smeur, Murat Bronz, and Guido C. H. E. de Croon. Incremental Control and Guidance of Hybrid Aircraft Applied to a Tailsitter Unmanned Air Vehicle. *Journal of Guidance, Control, and Dynamics*, 43(2):274–287, feb 2020.

[11] C. De Wagter, Bart Remes, Ewoud Smeur, Freek van Tienen, Rick Ruijsink, Kevin van Hecke, and Erik van der Horst. The NederDrone: A hybrid lift, hybrid energy hydrogen UAV. *International Journal of Hydrogen Energy*, pages 1–16, mar 2021.

[12] Ewoud J. J. Smeur, Qiping P Chu, and Guido C H E de Croon. Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Aerial Vehicles. *Journal of Guidance, Control, and Dynamics*, 39(3):450–461, 2016.

[13] Ewoud J.J. Smeur, D ; Höppener, and C. de Wagter. Prioritized Control Allocation for Quadrotors Subject to Saturation. In *IMAV 2017*, pages 37–43, 2017.