Instrumentation and control of a target fixed-wing drone for launch and capture

João Carvalho* and Bruno J. Guerreiro*†
* NOVA School of Science and Technology, UNINOVA-CTS, LASI, NOVA University Lisbon, 2829-516 Caparica, Portugal.
† Institute For Systems and Robotics, LARSyS, 1049-001, Lisbon, Portugal.
E-mails: jmfde.carvalho@campus.fct.unl.pt and bj.guerreiro@fct.unl.pt

***Abstract*—This paper presents the aerodynamics model of a small scale fixed-wing drone, the Multiplex Easyglider 4, together with the inner and outer loop control strategies to enable effective path following. The XFLR software is used to obtain the vehicle aerodynamic response based on a basic 3-D model of the fuselage and surfaces, towards having a complete dynamical model, including the aerodynamic coefficients. With this model, nonlinear control techniques are explored to deal with the aerodynamic nonlinearities and then integrated with a path following algorithm. Two types of attitude controllers were developed: a linear controller based on PI and a nonlinear controller based on the backstepping technique. An external loop was then added to make the vehicle follow a specific path. Two different techniques were implemented: a path following algorithm that would make the vehicle follow a vector field around the intended trajectory and an adaptive algorithm capable of dealing with uncertainties in the environment, such as wind with unknown direction and intensity. Simulation results are presented for each strategy and the instrumentation of the vehicle to be used in field trials is also provided.**

***Index Terms*—UAV, PID, Backstepping, Path following, Vector field, Adaptive**

I. Introduction

Drones, also known as unmanned aerial vehicles (UAVs), are aircraft that are either controlled remotely or operate autonomously through the use of computers and sensors. They can be used for a variety of purposes, such as surveillance, mapping, search and rescue, delivery of goods, and even for recreational purposes. One of the main benefits of drones is their ability to operate in environments that may be hazardous or difficult for humans to access, such as disaster-stricken areas or high altitudes. Fixed-wing aircraft are extremely energy efficient for long endurance flights, while not being able to hover in the air. Launching and landing can also be a difficult task, needing a catapult launcher or a runway, depending on its size. The term unmanned aerial vehicle does not only include the aircraft but all the systems associated, including sensors, navigation systems, users' interface and communications hardware [1]. In autonomous flight, sensors like magnetometer, accelerometer, gyroscope , static pressure sensor, dynamic pressure sensor and GPS have to be included in the aircraft.

Most UAV autopilots currently use classical PID controllers and ad-hoc tuning methods, which have limitations in terms of performance and robustness due to the nonlinearity of the vehicle motion equations. However, there are several techniques for controlling flying objects with nonlinear motion models, such as sliding mode control, backstepping control, adaptive control, or model predictive control. For instance, [2] proposed an approach called Adaptive Backstepping, which can control the attitude of a UAV and deal with unmatched parametric uncertainties in the plant's nonlinearities.

There are also plenty approaches towards guiding UAVs along a particular path. The fundamental idea is to define, explicitly of implicitly, a vector field around the desired path that can provide the necessary references for the vehicle to follow. The path following algorithms typically rely on Lyapunov stability analysis to ensure that the vehicle converges towards the desired path in a stable manner. In [3] a method that enhances the path following control law is presented, incorporating an estimator that mitigates the impact of unknown wind components. This results in an adaptive vector field path following strategy. An alternative approach detailed in [4], defines a geometric path-following algorithm, where the attitude control problem is formulated in the natural space of rotation matrices, SO(3).

The objective of this paper is to implement an autopilot and a path following for a fixed-wing UAV, in particular using the commercial frame Multiplex EasyGlider 4, depicted in Figure 1a, including all modeling, instrumentation, and integrating of the required systems to enable the vehicle to be captured in a cooperative effort by another rotary-wing UAV. This problem arises within the scope of the CAPTURE project [5], in which we see a rotary-wing drone (the shuttle) is required to capture a fixed-wing drone (the target), so that it can land in a confined environment. This problem presented in this paper is the modeling of the target drone, its instrumentation, the control of its attitude through non-linear control techniques robust to external disturbances, as well as the elaboration of an autopilot that makes the drone perform manoeuvres with a view to being integrated into a cooperation environment with the shuttle drone.

The paper is organized as follows. Section II provides the details on the vehicle modeling, both regarding the aerodynamic effects and rigid body dynamics, whereas the inner-

This work was partially funded by the FCT projects CAPTURE (PTDC/EEI-AUT /1732/2020), CTS (UIDB/EEA/00066/2020), and LARSYS (UIDB/50009/2020).

loop for attitude control and the outer-loop path following are presented in Section III. Section IV presents the vehicle instrumentation, detailing the materials and methods that were used to assemble all the hardware necessary for the UAV to fly. Simulation results are provided in Section V whereas final remarks and future work are offered in VI.

II. Vehicle Model

In this section, we discuss the modeling of aerodynamic forces and moments in the vehicle as well as the methods for obtaining the aerodynamic coefficients.

A. Rigid body kinematics and dynamics

The aircraft dynamic model is modeled as a 6 degrees-of-freedom (DoFs) rigid body, where the position of the aircraft relative to a local tangent plane frame, considered as an inertial frame I for simplicity, is defined as $\boldsymbol{p} = \begin{bmatrix} p_n & p_e & p_d \end{bmatrix}^T \in \mathbb{R}^3$. The p_d component is pointing downwards, and thus, we use $h = -p_d$ to refer to the altitude [1]. The linear velocity of the vehicle center of mass relative the the inertial frame, expressed in the body frame B is $\boldsymbol{v}_B = \begin{bmatrix} u & v & w \end{bmatrix}^T \in \mathbb{R}^3$, which is related with the variation of the aircraft position by the linear kinematics equation

$$\dot{\boldsymbol{p}} = \boldsymbol{R}\boldsymbol{v}_B \tag{1}$$

with $\boldsymbol{R} \in \mathcal{SO}(3)$ being the rotation matrix from the body frame to the inertial frame, and $\mathcal{SO}(3) = \{\boldsymbol{R} \in \mathbb{R}^{3\times 3} : \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}, \det(\boldsymbol{R}) = 1\}$ is the special orthogonal group in 3-dimensional space. Defining the Euler angles vector $\boldsymbol{\lambda} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \in \mathbb{R}^3$, and considering the Z-Y-X parametrization, the above rotation matrix can also be expressed as

$$\boldsymbol{R}(\boldsymbol{\lambda}) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \tag{2}$$

where ϕ is the roll angle, θ is the pitch angle, and ψ is the yaw angle, where $c_\alpha = \cos(\alpha)$ and $s_\alpha = \sin(\alpha)$ for simplicity.

The angular velocity of the body frame relative to the inertial frame, expressed in the body frame is defined as $\boldsymbol{\omega}_B = \begin{bmatrix} p & q & r \end{bmatrix}^T \in \mathbb{R}^3$, which can be related with the Euler angles using the angular kinematics equation

$$\dot{\boldsymbol{\lambda}} = \boldsymbol{Q}(\boldsymbol{\lambda})\boldsymbol{\omega}_B \tag{3}$$

with

$$\boldsymbol{Q}(\boldsymbol{\lambda}) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \tag{4}$$

Using a Newton-Euler approach, we can derive the linear and angular kinematics of the vehicle in the body frame, which will result in the equations

$$m\dot{\boldsymbol{v}}_B = -\boldsymbol{S}(\boldsymbol{\omega}_B)m\boldsymbol{v}_B + \boldsymbol{f}_A + \boldsymbol{f}_P + \boldsymbol{f}_G \tag{5}$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}}_B = -\boldsymbol{S}(\boldsymbol{\omega}_B)\boldsymbol{J}\boldsymbol{\omega}_B + \boldsymbol{\tau}_A + \boldsymbol{\tau}_P \tag{6}$$

where m denotes the vehicle mass, $\boldsymbol{J}$ the vehicle inertia tensor, $\boldsymbol{f}_*$ and $\boldsymbol{\tau}_*$ to be defined below are external forces and moments acting on the vehicle, respectively, and the operator $\boldsymbol{S}(.)$ is such that $\boldsymbol{S}(\boldsymbol{a})\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$.

B. Forces and Moments

As defined in Eqs. (5)-(6), we assume that the forces and moments are predominantly attributable to three sources: gravity $\boldsymbol{f}_G$, aerodynamics $\boldsymbol{f}_A$, and propulsion $\boldsymbol{f}_P$. The gravitational acceleration is assumed constant, g, and to act on the z axis of the inertial frame, $\boldsymbol{e}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$, being perceived at the body frame as $\boldsymbol{f}_G = mg\boldsymbol{R}^T\boldsymbol{e}_3$.

Aerodynamic forces and moments are determined by the aircraft's contact with the airflow, which is a function of the shape of its surface, and its attitude and also depends on the velocity relative to the surrounding air [6]. This relative velocity, $\boldsymbol{v}_a$, described in the body frame, is defined as the difference between the vehicle body velocity, $\boldsymbol{v}_B$, and that of the wind velocity as perceived at the body frame, resulting in $\boldsymbol{v}_A = \boldsymbol{v}_B - \boldsymbol{R}^T\boldsymbol{v}_W$, where $\boldsymbol{v}_W$ is the wind velocity in the inertial frame. Considering this relative velocity vector, $\boldsymbol{v}_A$ an auxiliary wind frame can be defined using the angle of attack, α, side-slip angle, β, and the course angle, χ, as $\boldsymbol{v}_A = \boldsymbol{R}_A(\alpha,\beta)V_a\boldsymbol{e}_1$, where $\boldsymbol{e}_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$, $\boldsymbol{R}_A(\alpha,\beta) = \boldsymbol{R}_z(\beta)\boldsymbol{R}_y(\alpha)$ and $V_a = \|\boldsymbol{v}_A\|$. The surfaces shape can be modified by control surfaces, which usually are aileron, rudder and elevator. Through the generated aerodynamic forces, the ailerons deflection δ_a controls the roll angle ϕ, the rudder deflection δ_r controls the yaw angle ψ and the elevator's deflection δ_e controls the pitch angle θ.

The aerodynamic force and moment vectors acting on a fixed-wing aircraft can be expressed as

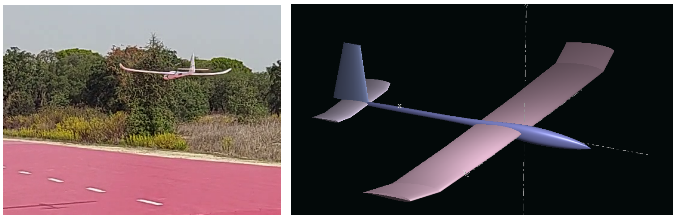
$$\boldsymbol{f}_A = \frac{1}{2}\rho V_a^2 S \boldsymbol{R}_y(\alpha)\begin{bmatrix} -C_D \\ C_Y \\ -C_L \end{bmatrix}, \quad \boldsymbol{\tau}_A = \frac{1}{2}\rho V_a^2 S \begin{bmatrix} bC_l \\ cC_m \\ bC_n \end{bmatrix} \tag{7}$$

where C_L, C_Y, and C_D are respectively the lift, drag, and lateral force nondimensional aerodynamic coefficients, c is the wing cord, b is the wing span, and C_l, C_m, and C_n are respectively the moments coefficients about the x, y, and z axis of the body frame [1]. These aerodynamic coefficients can be obtained by interpolation of data for a small range of α around 0 using XFOIL and semi-empirical methods for higher angles of attack, like wind tunnel [7], and linearized using the Taylor Series, resulting in the coefficients associated with the longitudinal dynamics

$$\begin{aligned} C_D &= C_{D_0} + C_D^\alpha\alpha + C_D^q\frac{c}{2V_a}q + C_D^{\delta_e}\delta_e \\ C_L &= C_{L_0} + C_L^\alpha\alpha + C_L^q\frac{c}{2V_a}q + C_L^{\delta_e}\delta_e \\ C_m &= C_{M_0} + C_M^\alpha\alpha + C_M^q\frac{c}{2V_a}q + C_M^{\delta_e}\delta_e. \end{aligned}$$

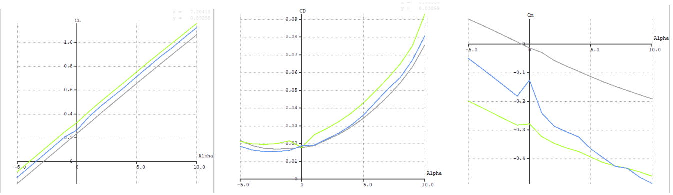
and those of the lateral dynamics

$$\begin{aligned} C_Y &= C_{Y_0} + C_Y^\beta\beta + \frac{C_Y^p b}{2V_a}p + \frac{C_Y^r b}{2V_a}r + C_Y^{\delta_r}\delta_r + C_Y^{\delta_a}\delta_a \\ C_l &= C_{l_0} + C_l^\beta\beta + \frac{C_l^p b}{2V_a}p + \frac{C_l^r b}{2V_a}r + C_l^{\delta_r}\delta_r + C_l^{\delta_a}\delta_a \\ C_n &= C_{n_0} + C_n^\beta\beta + \frac{C_n^p b}{2V_a}p + \frac{C_n^r b}{2V_a}r + C_n^{\delta_r}\delta_r + C_n^{\delta_a}\delta_a. \end{aligned}$$



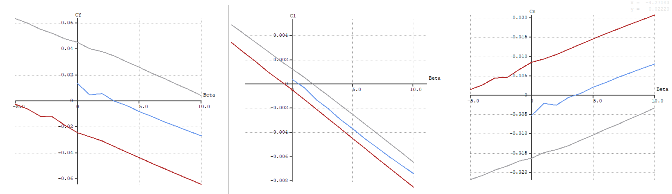
(a) EasyGlider 4 trial. (b) Lift force.

Fig. 1: Multiplex EasyGlider 4.



(a) Lift force. (b) Drag force. (c) Pitching moment.

Fig. 2: Longitudinal coefficients as functions of α. Grey line: negative elevator deflection; Blue line: neutral configuration; Green line: positive elevator deflection.



(a) Lateral force. (b) Roll moment. (c) Yaw moment.

Fig. 3: Lateral coefficients as functions of β and δ_r. Red line: negative rudder deflection; Blue line: neutral configuration; Grey line: positive rudder deflection.

These coefficients are determined using XFOIL and semi-empirical methods, including constant terms and those that express affine relations with the variables α, β, ω_B components, and on the surfaces angles δ_e, δ_a, and δ_r.

Finally, we have to consider the forces and moments produced by the rotor, which, for control purposes, can simply be defined as

$$\boldsymbol{f}_P = \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{\tau}_P = \begin{bmatrix} -\frac{DC_Q}{C_T} T \\ 0 \\ 0 \end{bmatrix} \tag{8}$$

where D is the propeller diameter, C_T and C_Q are non-dimensional aerodynamic coefficients. The thrust can also be described as a function of the rotor angular velocity, Ω_p, as $T = \frac{\rho D^4}{4\pi^2}\Omega_p C_T$, whereas C_T and C_Q are usually quadratic functions of the air speed and the rotor angular velocity [1], yet can be approximated as constants for certain operational conditions.

C. Aerodynamic analysis

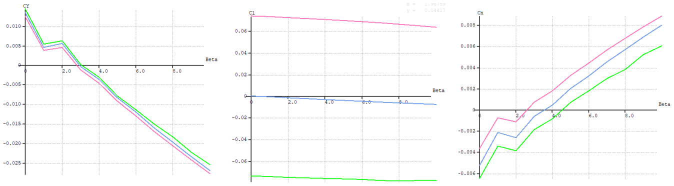
The next step is to obtain the aerodynamic coefficients as a function of the wind frame angles α and β and the deflections of the rudder δ_r, the elevator δ_e and aileron δ_a, using XFLR software. The XFLR uses the XFOIL engine, which can calculate the pressure distribution on a 2D airfoil given the coordinates specifying the shape, Reynolds and Mach numbers. By replicating the EasyGlider 4 model in the XFOIL software, in particular, the wings and tail surfaces, fuselage, as well as its mass distribution, as illustrated in Figure 1b, we can use this software to obtain the necessary coefficients.

Two types of analysis is performed: (i) obtain the coefficients associated with the longitudinal dynamics, varying the angle of attack α, and (ii) the coefficients associated with the lateral dynamics, varying the β angle, both at a fixed speed with all combinations of elevator, rudder and aileron deflections.

Looking at the Figure 2 we can see that in each graph we have 3 curves that represent the aerodynamic response for all the configurations of the elevator deflection. If we examine the CL-alpha plot, we can see the lift force increased as we deflect the elevator positively and the pitching moment Cm decreases, as expected. Next, the same procedure was done, but this time varying the beta angle in order to obtain the lateral dynamic response, for all rudder deflection combinations, which are again: neutral, positive deflection and negative deflection.

The lateral aerodynamic response is graphically represented in Figures 3 and 4, as lateral, roll, and yaw, respectively, for rudder and aileron deflections. As we can see, the roll moment coefficient is relatively small when compared to the yaw moment, however it is not zero, confirming that the control of surfaces is not exclusive to its corresponding movement. The graphs are expected to be linear, as they are computed at low angles, however, as weight was added to the UAV in a manner not programmed by the manufacturer, its aerodynamic response is quite abnormal.

To conclude, understanding the forces and moments acting on an aircraft is crucial for aircraft design and control. The mathematical models presented in this section provide a way to analyze these forces and moments, and their effect on the aircraft's stability and control.



(a) Lateral force. (b) Roll moment. (c) Yaw moment.

Fig. 4: Lateral coefficients as functions of β and δ_a. Green line: negative aileron deflection; Blue line: neutral configuration; Pink line: positive rudder deflection.

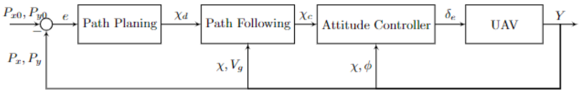


Fig. 5: Full control system design.

III. Control Design

This section aims at designing a path planning and a path following controller as well as the low-level controller, in order to control the attitude of the UAV. As shown in Figure 5, we can see that in the outer loop we have the controller responsible for guiding the vehicle along a trajectory (path planning and path following) and in the inner loop we have the low-level attitude controller.

A. Attitude Controller

The attitude controller uses the angular velocities p, q and r, attitude angles ϕ, θ and ψ, by sending angular references to the control surfaces of the UAV. This section presents a non-linear control approach using backstepping based on the method presented in [2].

The expressions of the elevator, rudder and aileron deflections were obtained by using the backstepping control method. This controller can be also divided into two parts: Longitudinal controller, which is responsible to control the pitch angle θ through the elevator deflection δ_e, and the Lateral controller which controls the roll angle ϕ and yaw angle ψ, with the aileron deflection δ_a and rudder deflection δ_r, respectively. With this method, the attitude angles are controlled by means of the angular rates, and the angular rates are controlled by control surfaces deflection.

The longitudinal controller is responsible to stabilize the pitch angle θ. Starting by defining the pitch error $e_\theta = \theta - \bar{\theta}$, with $\bar{\theta}$ being the pitch angle desired value and the pitch angular velocity error $e_q = q - \bar{q}$, with

$$\bar{q} = (-\mu_\theta e_\theta + r \sin\phi)/\cos\phi \tag{9}$$

its derivative can be written as

$$\dot{e}_\theta = -\mu_\theta e_\theta + e_q \tag{10}$$

with μ_θ a positive constant. Considering the Lyapunov function $V_{lon} = 1/2e_\theta^2 + 1/2e_q^2$, it can be seen that a stabilizing control law for δ_e can be defined as

$$\delta_e = \frac{J_y}{P_d S \bar{c} C_{m_{\delta_e}}}(\delta_1 - \frac{P_d S \bar{c}}{J_y}\delta_2 + \delta_3) \tag{11}$$

with

$$\delta_1 = -e_\theta cos(\phi) - \mu_q e_q \tag{12}$$

$$\delta_2 = C_{m_0} + C_{m_\alpha}\frac{\theta - \gamma - \beta\sin\phi}{\cos\phi} + \frac{\bar{c}}{2V}C_{m_q}q \tag{13}$$

$$\delta_3 = \frac{J_x - J_z}{J_y}pr + \frac{J_{xz}}{J_y}(p^2 - r^2) + \dot{\bar{q}} \tag{14}$$

being μ_q a positive constant. For these expressions, the inertia tensor is assumed to have a particular structure, with nonzero elements J_x, J_y, J_z, and J_{xz}. Some dependencies on other angular velocity components and surface deflections, which would be mainly due to the lateral force terms, are also assumed negligible.

For the lateral control, we start by defining the attitude errors $e_\phi = \phi - \bar{\phi}$ and $e_\psi = \psi - \bar{\psi}$, with $\bar{\phi}$ and $\bar{\psi}$ being the desired roll and yaw angles, respectively, and also the angular velocity errors $e_p = p - \bar{p}$ and $e_r = r - \bar{r}$, with

$$\bar{p} = -\mu_\phi e_\phi - q\tan(\theta)\sin(\phi) - r\tan(\theta)\cos(\phi) \tag{15}$$

$$\bar{r} = -\mu_\psi \frac{\cos(\theta)}{\cos(\phi)}e_\psi - \frac{\sin(\theta)}{\cos(\phi)}q \;. \tag{16}$$

As such, the angular error derivatives can be written as

$$\dot{e}_\phi = -\mu_\phi e_\phi + e_p \tag{17}$$

$$\dot{e}_\psi = -\mu_\psi e_\psi + e_r \tag{18}$$

with μ_ϕ and μ_ψ positive constants. Considering the Lyapunov function $V_{lat} = 1/2e_\phi^2 + 1/2e_p^2 + 1/2e_\psi^2 + 1/2e_r^2$, it can be seen that a stabilizing control law for the rudder and aileron deflections is given by

$$\delta_r = \frac{\tilde{A}_1 C_{n_{\delta_a}} - \tilde{A}_2 C_{l_{\delta_a}}}{C_{l_{\delta_r}} C_{n_{\delta_a}} - C_{l_{\delta_a}} C_{n_{\delta_r}}} \tag{19}$$

$$\delta_a = \frac{-\tilde{A}_1 C_{n_{\delta_r}} + \tilde{A}_2 C_{l_{\delta_r}}}{C_{l_{\delta_r}} C_{n_{\delta_a}} - C_{l_{\delta_a}} C_{n_{\delta_r}}} \tag{20}$$

with

$$\begin{aligned}\tilde{A}_1 &= \frac{J_x}{P_d S b}(-e_\phi - \mu_p e_p + \frac{J_z - J_y}{J_x}qr - \frac{J_{xz}}{J_x}pq + \dot{\bar{p}}) \\ &\quad - [C_{l_\beta}\beta + \frac{b}{2V}(C_{l_p}p + C_{l_r}r)] \\ \tilde{A}_2 &= \frac{J_z}{P_d S b}(-e_\psi - \mu_r e_r + \frac{J_y - J_x}{J_z}pq + \frac{J_{xz}}{J_z}qr + \dot{\bar{r}}) \\ &\quad - [C_{n_\beta}\beta + \frac{b}{2V}(C_{n_p}p + C_{n_r}r)]\end{aligned}$$

Having a stabilizing controller for the vehicle attitude, we can proceed to design motion direction references, to enable higher levels of autonomy, for instance defining a course angle command, χ, which in the absence of wind can be considered the same as the yaw angle, ψ, as detailed below. This attitude controller can also be modified to take an angular velocity to zero instead of the angular position to a desired value, as is usually necessary to dampen the dutch roll mode and achieve effective coordinated turn behaviour.

B. Path Following

Unlike the trajectory tracking problem, path following makes the UAV converge on a certain trajectory without any time specification [4]. The typical controller is based on defining vector fields that point towards the path, and once there align the vehicle motion with the path direction, for a fixed airspeed V_a, such as the method detailed in [3]. Usually these methods are defined in the horizontal plane and later, with simple changes, the vehicle can follow more general 3-D paths.

An alternative control technique combines the path following law based on the vector field method with an adaptive term that mitigates the effect of the unknown wind component [8]. The authors consider the wind as having a constant known component and an unknown time-varying in component. The designer is unaware of all the parameters relating to the time-varying element of the wind. The designer only knows the wind's constant component.

The desired course angle for following a straight line with this strategy can be defined as

$$\chi^d = \chi^\infty \frac{2}{\pi} \arctan(k e_p) + \arctan(a) \tag{21}$$

where e_p is a position error relative to a straight line, a is the direction of the line, and χ^∞ is the path approach course angle, and k is a constant parameters that can be used to tune the smoothness of the transition between approach and path following. Using a Lyapunov stability approach the ensure the position error, e_p, and the course angle error, $\tilde{\chi} = \chi - \chi_d$, go to zero, resulting in the control law for the commanded course angle given by

$$\chi_c = \chi + \frac{2\chi^\infty}{\alpha_f \pi} \frac{k}{1 + (k e_p)^2} \hat{V}_g (\sin\chi - \arccos\chi) - \frac{\kappa}{\alpha_f} \mathrm{sat}(\frac{\tilde{\chi}}{\epsilon}) \tag{22}$$

where the estimator for the ground velocity is defined as

$$\dot{\hat{V}}_g = -\Gamma \frac{2\rho\tilde{\chi}\chi^\infty}{\pi} \frac{k}{1 + (k e_p)^2} (\sin\chi - \arccos\chi) - \sigma\Gamma\hat{V}_g \tag{23}$$

where Γ is a positive estimator gain, σ is the σ-modification value, which allows us to project an adaptive control without the need of knowing the boundaries of the disturbances, ρ and κ are positive scale factors, and α_f is a constant that captures the first-order dynamics behaviour of the vehicle.

The method for orbit path following is similarly defined, considering the desired course as

$$\chi_d = \gamma + \frac{\pi}{2} + \arctan(k\tilde{d}). \tag{24}$$

where $\tilde{d}$ is the radial distance to the orbit, γ is the angular position of the vehicle and k is a positive constant. Using Lyapunov stability to ensure the angular distance and the course error are driven to zero results in the control law

$$\chi_c = \chi + \frac{\hat{V}_g}{\alpha_f d} \sin(\chi - \gamma) + \frac{\beta\hat{V}_g}{\alpha_f} \cos(\chi - \gamma) - \frac{\kappa}{\alpha_f} \mathrm{sat}(\frac{\tilde{\chi}}{\epsilon}) \tag{25}$$

with estimator

$$\dot{\hat{V}}_g = -\Gamma \frac{\rho\tilde{\chi}}{d} \sin(\chi - \gamma) + \beta\cos(\chi - \gamma)) - \sigma\Gamma\hat{V}_g. \tag{26}$$

We have now defined the control signal χ_c for both straight and orbit follow algorithms, which will be sent to the autopilot. We will then test this algorithm with the attitude controller and with the model developed, in a closed loop.

IV. Instrumentation

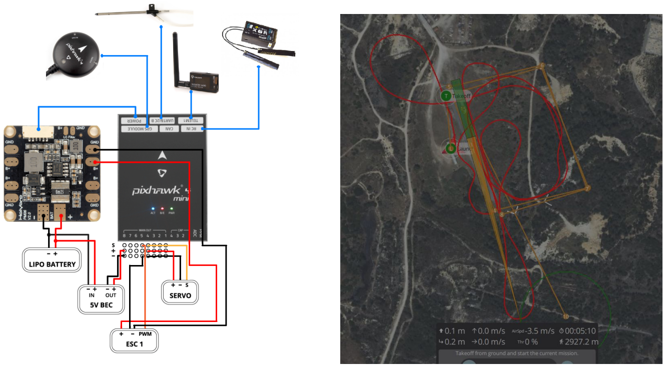
This section describes the configuration and assembly of the hardware necessary for the UAV to perform an autonomous flight, including wiring, electronic components, flight modes configuration and sensor calibration, using the *Multiplex EasyGlider 4* platform, illustrated in Figure **??**.

A. Hardware

The PX4 autopilot firmware stack is considered in combination with the open-hardware standard Pixhawk, which can provide a high-end autopilot for small unmanned vehicles at a low cost. The EasyGlider 4 has a wingspan of 1.8 m, an overall length of 1,08 m and a wing area of 40 dm^2. With the additional hardware configuration, the mass increases from 1100g to 1470g, and its center of mass is sligtly shifted, latter compensated by readjusting the battery position. We use the *Pixhawk 4 mini*, in combination with the electronic speed controller (ESC) *Roxxy BL Control 720 S-BEC*, with an integrated 5V BEC, and four 5V servos to move the control surfaces, which will be harnessed and connected to the Pixhawk, as detailed in Figure 6a.

For manual flight mode, we only need the radio and telemetry system connected to the Pixhawk, the ESC, the servos, a radio transmitter and receiver. The telemetry module, connected to the **TELEM1** pin, will be exclusively used to monitor the vehicle and arm it to start the flight, in QGroundControl software. As we can see in Figure 6a, we need to connect the Pixhawk to the power management board (PMB), which functions as both a power module and a power distribution board, delivering power to the ESC from a 3s LiPo battery.

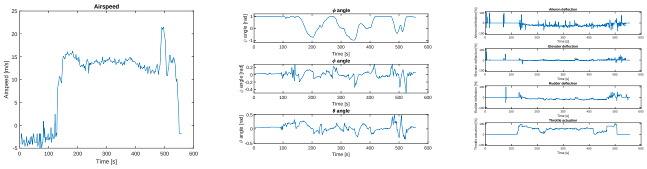
To carry out an autonomous flight, Pixhawk will have to receive information about the path it will have to follow, as well as the settings it will have to follow, through the QGround Control software. The UAV can also operate in Off-Board control mode, which allows us to control the drone using an external controller, running on a computer. In both cases, Pixhawk will communicate with the ground computer through the telemetry module using the MAVLink protocol. Vehicle stabilization and navigation control will be done using peripheral modules, as shown in figure 6a. The *Pixhawk 4 mini* is integrated with a gyroscope/accelerometer, a magnetometer and a barometer, therefore it is extremely important that the controller is aligned with the body frame axis. Then it is necessary to connect the GPS module, integrated with a compass, to the **GPS MODULE** input, to provide information about the spatial location of the UAV. For fixed-wing drones, airspeed sensors are critical for autonomous flight, since the autopilot has no other way of detecting stall condition. The pitot tube-based airspeed sensor communicates with the Pixhawk via the I2C protocol, and is therefore connected to the Pixhawk's **UART & I2C** input. In order to provide a power supply to the servos, it is necessary to connect a 5V BEC, which is integrated into the ESC, to the **MAIN OUT** pins. This module operates in 2 modes: 5V output and 6V output. In this application, 5V is sufficient.



(a) High level schematic.

(b) Flight test results map.

Fig. 6: Vehicle avionics and flight test map.



(a) Airspeed.

(b) Attitude.

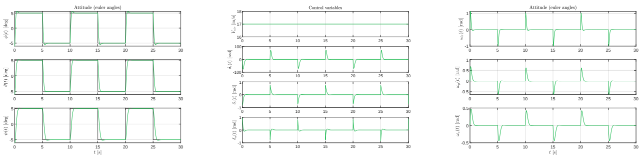
(c) Control effort.

Fig. 7: Flight test in autonomous mission mode: state and control temporal evolution

B. Tests

In order to validate the instrumentation effort, a basic remotely piloted flight test was performed. The goal was to validate the general assembly , namely the connection of the servos, the propeller and the telemetry and radio communication mechanisms, through the acquisition of data and control, respectively. Once it has been confirmed that the vehicle is well assembled through manual flight, an autonomous flight was performed. Mission mode instructs the vehicle to carry out a specified autonomous mission (flight plan) that was uploaded to the flight controller. A Ground Control Station is used to create and upload the mission. For security reasons, the PX4 was configured to be able to operate in two different modes: in manual mode and in mission mode. The mission is monitored and commanded through the QGroundControl, which communicates with the PX4 using the telemetry module, using the MAVLINK protocol. In Figures 6b and 7 we can see the mission that was planned and the UAV tracking the desired path. The UAV does the launch and the take off in manual mode, then switched to mission mode. To make the landing, it is necessary to first move to the position marked with the number 10, in order to position itself for the landing area marked as orange, where the UAV loses altitude, and green, where the UAV lands.

V. Simulation Results



(a) Attitude

(b) Actuation

(c) Angular velocities

Fig. 8: Attitude controller results for different gains.

This section aims to validate and show the performance of the methods developed above using Matlab/Simulink. First, we will test the attitude controller, giving it angular references, and then the path following algorithm, with spatial coordinates as references.

A. Attitude controller

The backstepping-based attitude controller was tested using the model and attitude controller presented in Sections II and III. The conditions used in this simulation assume $\boldsymbol{v}_B = \begin{bmatrix}20 & 2 & 0\end{bmatrix}^T$ m/s, $\boldsymbol{\omega}_B = \mathbf{0}$ rad/s, $\boldsymbol{\lambda} = \begin{bmatrix}0.01 & 0.01 & 0.01\end{bmatrix}^T$ rad, $\alpha = 0.02$ rad, and $\beta = 0.1$ rad.

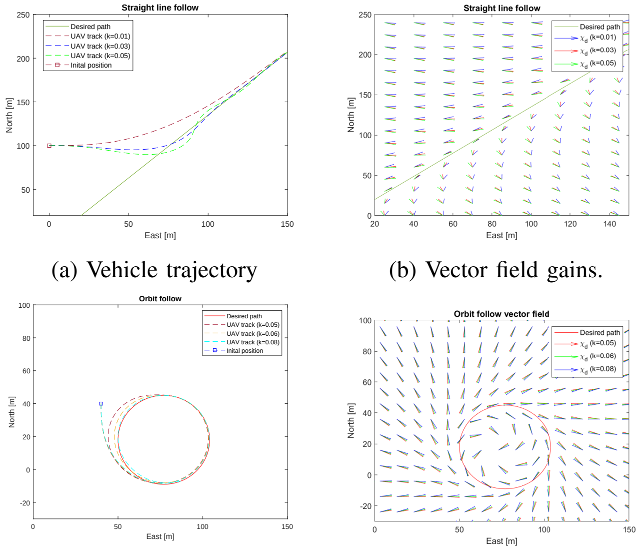
The choice for the parameters μ_p, μ_q and μ_r was made based on experience. First, they were all set to 1 and changed so that the control effort was as smooth as possible while ensuring that the response was fast enough so that the UAV was set to the desired attitude in a reasonable time. The same logic was applied to tune the values of μ_ϕ, μ_θ and μ_ψ. In Figure 8 we can observe the temporal evolution of the variables associated with the roll, pitch, and yaw channels, where the higher the values of μ_ϕ, μ_θ and μ_ψ the better the control performance is. However, the control surfaces exert a higher effort, which some applications may be undesirable.

B. Path following

This section presents the results for the real-time implementation in Matlab/Simulink of the path-following control algorithm with the vector field path following and the adaptive path following vector field based, for straight line following, orbit following and then show the capacity of the algorithm in performing more complex paths, comparing both path following algorithms.

In the simulation of the straight line following vector field the initial conditions are consider $\boldsymbol{p} = \begin{bmatrix}0 & 100 & 30\end{bmatrix}^T$ m, $V = 9$ m/s, $\alpha = 0.03$ rad, $\beta = 0$ rad, and $\chi = \pi/4$ rad, whereas the control parameters are $\alpha_f = 15$, $\chi^\infty = \pi/2$, $\epsilon = 0.5$, and $\kappa = \pi/2$. In Figure 9 we can see the experimental results of the straight line path following algorithm. We can see that the vector field, represented the χ_d approaches the direction of the line as the position error, represented in Figure 9b, goes to zero as expected. By changing the value of k, we can see in Figure 9a the different performances of the UAV following the straight line. It goes without saying that the smaller the value of k, the smoother the UAV approaches the path. We can justify that by observing figure **??**, and see that, the larger the value of k, the faster the direction of the vector field approaches the direction of the line.

In the simulation of the orbit following vector field the initial conditions are $\boldsymbol{p} = \begin{bmatrix}40 & 40 & 30\end{bmatrix}^T$ m, $V = 9$ m/s,



(a) Vehicle trajectory

(b) Vector field gains.

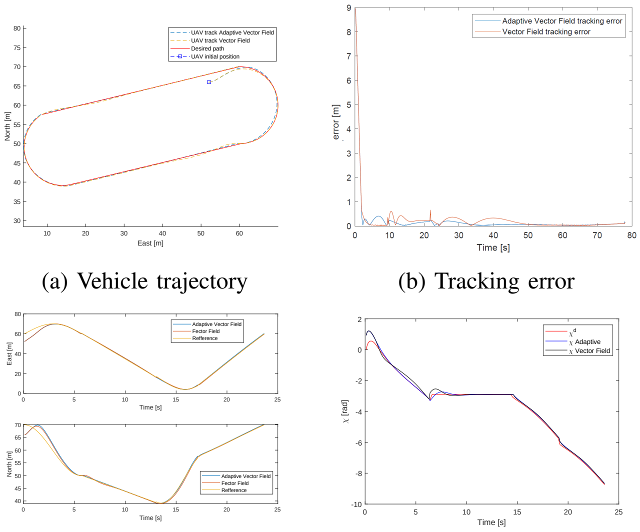
(c) Vehicle trajectory

(d) Vector field gains.

Fig. 9: Straight and orbit following results

$\alpha = 0.03$ rad, $\beta = 0$ rad, and $\chi = -\pi/2$ rad, and the control parameters of the straight line case were used. In Figure **??** we can also see the experimental results of the orbit follow algorithm. As in the previous algorithm, by varying the value of k, we can see in Figure 9c that the higher the value of k, the more attracted the UAV is to the desired path. In Figure 9d we can see that the smaller the value of k, the less the field vectors point toward the center of the trajectory, causing the UAV to scatter from its path.

We also test the path following algorithms, adaptive an non-adaptive, in more elaborate maneuvers, as in the case of Dubins Path with Right-Straight-Right (RSR) configuration, ending with a straight line to close the path. In order to test the two path following algorithms and compare their performance, we will introduce a wind component with constant amplitude and direction to the system, according to $V_W = 1.4$ m/s $\phi_w = 1.75$ rad, as well as another wind component with variable amplitude and direction, with $A = 3\cos 0.1t$ m/s and $\phi_A = 2.7\cos 0.1t$ rad. The control parameters are $k = 0.05$, $\alpha_f = 13$, $\epsilon = 0.5$, $\Gamma = 74$, and $\kappa = \pi/2$. In Figure 10 we can see the comparison between the two algorithms. We can see that the spatial response of the adaptive algorithm is more satisfactory, presenting fewer oscillations, and approaching the reference faster. It is noteworthy that the non-adaptive algorithm, when it passes from the first left curve to the straight line, presents an undesirable oscillation, being evident in Figure 10b that its tracking error is greater than that of the adaptive after starting to follow the line (approximately at t=9s). We can also see the temporal evolution of position and course angle, where the transitions between path components is more visible in terms of the desired course angle.



(a) Vehicle trajectory

(b) Tracking error

(c) Position

(d) Course angle

Fig. 10: Dubins path following results.

VI. Concluding Remarks

In conclusion, this work provides a comprehensive study on the modeling, control design, and path following algorithms for UAVs. The proposed dynamic model, attitude controller, and path-following controller can be used in the design and control of UAVs for various applications. The main contributions of this work are the development of an attitude controller using both linear and non-linear methods, the implementation of a path-planning and path-following controller and the UAV instrumentation. The proposed controllers and path following algorithms have been successfully simulated in Matlab, but further validation is needed through real-world flight testing. The next step in this work is to implement the controllers and path following algorithm on an autonomous UAV using off-board control instead of the PX4 algorithm.

References

[1] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice-Princeton University Press.pdf*. 2012.

[2] M. Lungu and R. Lungu, “Adaptive backstepping flight control for a mini-uav,” *International Journal of Adaptive Control and Signal Processing*, vol. 27, no. 8, pp. 635–650, 2013.

[3] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, “Vector field path following for miniature air vehicles,” *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 519–529, 2007.

[4] V. Cichella, E. Xargay, V. Dobrokhodov, I. Kaminer, A. M. Pascoal, and N. Hovakimyan, “Geometric 3D path-following control for a fixed-wing UAV on SO(3),” *AIAA Guidance, Navigation, and Control Conference 2011*, no. August, pp. 1–15, 2011.

[5] B. Guerreiro, “Project capture,” 2023.

[6] K. Goebel and B. Saha, *Prognostics applied to electric propulsion uav*. 2015.

[7] W. Khan and M. Nahon, “Real-time modeling of agile fixed-wing UAV aerodynamics,” *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, pp. 1188–1195, 2015.

[8] B. Zhou, H. Satyavada, and S. Baldi, “Adaptive path following for Unmanned Aerial Vehicles in time-varying unknown wind environments,” *Proceedings of the American Control Conference*, pp. 1127–1132, 2017.

[9] Pixhawk, “Px4 autopilot user guide (main).”