Page 1 of 22. © Royal Aeronautical Society 2021
doi: 10.1017/aer.2016.1

Design and Development of a Backstepping Controller Autopilot for Fixed-wing UAVs

Daniele Sartori
dan.sartori@sjtu.edu.cn

Shanghai Jiao Tong University
Institute of Sensing and Navigation
Shanghai
China

Fulvia Quagliotti

Politecnico di Torino
Department of Mechanical and Aerospace Engineering
Torino
Italy

Matthew J. Rutherford

University of Denver
Department of Computer Science
Denver
USA

Kimon P. Valavanis

University of Denver
Department of Electrical & Computer Engineering
Denver
USA

Received DD MM YYYY; revised DD MM YYYY; accepted DD MM YYYY.

ABSTRACT

Backstepping represents a promising control law for fixed-wing Unmanned Aerial Vehicles (UAVs). Its nonlinearity and its adaptation capabilities guarantee adequate control performances over the whole flight envelope, even when the aircraft model is affected by parametric uncertainties. In the literature, several works apply backstepping controllers to various aspects of fixed-wing UAV flight. Unfortunately, many of them have not been implemented in a real-time controller and only few attempt simultaneous longitudinal and lateral-directional aircraft control. In this paper, an existing backstepping approach able to control longitudinal and lateral-directional motions is adapted for the definition of a control strategy suitable for small UAV autopilots. Rapidly changing inner-loop variables are controlled with non-adaptive backstepping, slower outer loop navigation variables are controlled with PID. The controller is evaluated through numerical simulations for two very diverse fixed-wing aircraft performing complex maneuvers. The controller behavior with model parametric uncertainties or in presence of noise is also tested. The performance results of a real-time implementation on a microcontroller are evaluated through hardware-in-the-loop simulation.

NOMENCLATURE

b	wingspan, m
C_L, C_Y	aerodynamic coefficients
$C_{L\alpha}$	lift aerodynamic derivative
$C_{l\beta}, C_{l\dot{\beta}}, C_{lp}$, C_{lr}	roll moment aerodynamic derivatives
$C_{m0}, C_{m\alpha}, C_{m\dot{\alpha}}, C_{mq}$	pitch moment aerodynamic derivatives
$C_{n\beta}, C_{n\dot{\beta}}, C_{np}$, C_{nr}	yaw moment aerodynamic derivatives
$C_{l\delta_a}$, $C_{l\delta_e}$, $C_{l\delta_r}$	roll moment control derivatives
$C_{m\delta_a}$, $C_{m\delta_e}$, $C_{m\delta_r}$	pitch moment control derivatives
$C_{n\delta_a}$, $C_{n\delta_e}$, $C_{n\delta_r}$	yaw moment control derivatives
c	wing mean aerodynamic chord, m
$\boldsymbol{F} = (F_x, F_y, F_z)^T$	body-axes force vector, N
$f(\omega_1, y)$	change of variable function
$f_\alpha(\alpha, y_\alpha)$, $f_\beta(\beta, y_\beta)$	functions of the aircraft states
$f_i(\mathbf{x}, \xi_\mathbf{j})$	general nonlinear function
g	gravity acceleration, m/s^2
g_2, g_3	gravity contributions, m/s^2
H	change of variable reference value
h, h^{ref}	measured and reference altitude, m
I	body-axes inertia matrix, kg·m^2
k_u, k_1, k_2	change of variable backstepping controller gains
$k_{\alpha,1}$, $k_{\alpha,2}$, $k_{\beta,1}$, $k_{\beta,2}$	backstepping controller gains
$Lift$	lift force, N
$\boldsymbol{M} = (M, L, N)^T$	body-axes moment vector, N·m
m	aircraft mass, kg
os	step response overshoot
$\hat{p}$, $\hat{q}$, $\hat{r}$	nondimensional angular rates
p_s, p_s^{ref}	measured and reference stability-axes roll rate, rad/s
R_{sb}	rotation matrix

T	engine thrust, N
t_r	step response rise time from 10% to 90% of reference, s
t_s	step response settling time at ±2% of reference, s
$\boldsymbol{u}_c = (u_1, u_2, u_3)^T$	control action vector, rad/s^2
u_s	change of variable control action
V, V^{ref}	measured and reference linear velocity, m/s
$\boldsymbol{V} = (u, v, w)^T$	body-axes linear velocity vector, m/s
X, Y, Z	body-axes aerodynamic forces, N
X_B, Y_B, Z_B	body axes
X_N, Y_N, Z_N	NED axes
X_S, Y_S, Z_S	stability axes
X_W, Y_W, Z_W	wind axes
x_1, x_2	global change of variable states
$\mathbf{x}$	general state vector $\in \mathbb{R}^n$
y	general change of variable state
y_α, y_β	change of variable states for f_α and f_β

Greek Symbol

α, α^{ref}	measured and reference angle of attack, rad
β	sideslip angle, rad
δ_{th}	throttle command
$\boldsymbol{\delta} = (\delta_a, \delta_e, \delta_r)^T$	surface deflections vector, rad
ξ_j	general scalar state
σ	sensor measure standard deviation
$\boldsymbol{\Phi} = (\phi, \theta, \psi)^T$	body axes to NED axes Euler angles vector, rad
$\Psi(x_1)$	function of the global change of variable state
ψ, ψ^{ref}	measured and reference heading angle, rad
$\Omega(x_1)$	global change of variable function
$\boldsymbol{\omega} = (p, q, r)^T$	body-axes angular velocity vector, rad/s
$\boldsymbol{\omega}_s = (p_s, q_s, r_s)^T$	stability-axes angular velocity vector, rad/s
ω_1, ω_2	change of variable controlled states

1.0 Introduction

The flight dynamics of fixed-wing Unmanned Aerial Vehicles (UAVs) is characterized by highly nonlinear behavior. A severe coupling exists between longitudinal and lateral-directional dynamics and the sensitivity to external disturbances is considerable. The interest in finding a suitable control law for these systems is growing in response to the recognition that these platforms will soon be performing missions in many civilian applications.

In a recent publication[(4)], the authors illustrate the state of the art of advanced control laws design for fixed-wing unmanned aircraft. While several interesting solutions exist, the use of PID gains is still a popular approach in practice, as demonstrated by widely available open source firmware autopilots, such as the Pixhawk family[(5)]. This method guarantees simple implementation and low computational effort, the designer has adequate control over the system response and a clear understanding of the control action. The tuning of the PID gains can be performed with many non-heuristic methods, as explained in[6] and[7], and currently the research focuses on the auto-tuning process, see for example[8]. One drawback of the standard PID approach is the inability to cope with the full flight envelope, therefore gain scheduling is commonly adopted as the performances of a PID controller decrease when moving away from the design point. Another disadvantage is that traditional PIDs do not guarantee enough robustness to the extent of model parametric uncertainties expected to be found in small fixed-wing unmanned aircraft.

The solutions proposed in[4] include nonlinear, adaptive or robust control laws able to theoretically guarantee satisfying performance over a large flight envelope, also in presence of uncertainties. For instance the authors of[9] propose a nonlinear model predictive control for fixed-wing UAV path tracking,[10] investigates the feasibility of H_2 and H_∞ autopilots for longitudinal UAV control and[11] employs an adaptive second order sliding mode for robust attitude and airspeed control. Nevertheless, several practical constraints might make these controllers impractical for small fixed-wing UAVs implementation. High computational requirements, complex algorithms and the necessity to smoothly combine high-level intelligent tasks with low-level input/output routines are the main obstacles. The miniaturization and reduction in cost of microcontrollers, together with their increase in performance, is now enabling researchers to deploy fixed-wing unmanned aircraft flown by self-developed control laws. Whereas several examples have been published for multi-rotor configurations, the number is relatively small for fixed-wing aircraft. An example is[12], which further extends the work of[9] and solves the need of compromising between smooth convergence and computational performance originally highlighted by the authors. Similarly, in[13] a control strategy combining model predictive and $\mathcal{L}_1$ control is designed, implemented and flight tested on a fixed-wing UAV.

Within this context, an autopilot configuration for longitudinal and lateral-directional fixed-wing aircraft control based on nonlinear backstepping is presented in this paper. The objective of the paper is double: the adaptation of an existing backstepping technique[2] with the aim of generating a comprehensive control configuration suitable for mini-UAV autopilots; its real-time implementation on a microcontroller board. In fact, differently from many related studies, the implementation simplicity of the proposed approach allows to actually exploit the backstepping controller advantages on a fixed-wing UAV. This has been demonstrated in a related work from the same authors[45], while the current paper focuses on the theoretical framework and on the pre-flight software and hardware simulation and validation. It is shown that, unlike traditional linear control techniques such as LQ or feedback linearization, backstepping is a nonlinear control law that guarantees satisfying performance over the whole flight envelope[1]. Useful nonlinearities are maintained and additional nonlinear damping terms can be introduced to increase robustness to model errors or to improve transient performance[18]. Furthermore, as backstepping belongs to the Lyapunov family, it has guaranteed convergence of the tracking error and asymptotic stability[19].

The paper is organized as follows. Section 2 describes the existing applications of backstepping controllers to fixed-wing UAVs. Section 3 briefly presents the aircraft equations of motion and how they are arranged in a suitable form for the controller design, which is later introduced in Section 4 . Section 5 describes the results of the software simulations, while Section 6 presents the implementation and the hardware-in-the-loop (HIL) simulations results. Finally, Section 7 concludes the paper and describes the future work towards achieving comprehensive flight tests.

2.0 Related Work

A variety of projects illustrate the application of the backstepping technique for fixed-wing aircraft flight control. Unlike with rotorcraft, difficulties exist in arranging the equations of motion into the required cascade form. For this reason, in the literature several examples treat longitudinal and lateral-directional control independently. For instance,[20] presents an adaptive backstepping control law for angle of attack tracking,[22] uses adaptive backstepping for UAV velocity and flight path angle control and[23] combines $\mathcal{L}_1$ adaptive methodology with backstepping for longitudinal control of a multi-axis thrust vectoring fighter aircraft. The path-following problem is addressed in[25], where the roll angle command is generated through backstepping with the parameter adaptation technique, hardware-in-the-loop (HIL) simulations validate the results. Fault-dependent control allocation using $\mathcal{L}_1$ adaptive backstepping for longitudinal UAV control is analyzed in[26].

The number of works describing combined longitudinal and lateral-directional aircraft control is limited. In[27], outer loop variables such as incidence, sideslip angle and roll angle are controlled by neural network adaptive backstepping through body-axes angular rates. In[28], constrained adaptive backstepping with neural adaptation laws is employed for tracking angle of attack, stability-axes roll rate and total velocity while the sideslip angle is maintained at zero. The authors of[21] present different backstepping solutions, combined with sliding mode control, for the decoupled control of altitude, roll angle and heading angle.

In contrast to the majority of existing work, a comprehensive autopilot configuration for combined longitudinal and lateral-directional control of fixed-wing UAVs based on the backstepping technique is presented in this paper. Inner loop variables, angle of attack, sideslip angle and stability-axes roll rate, are controlled via the backstepping approach described in[2] and[3]. This method is designed for general aircraft maneuvering within the whole flight envelope. Its ability to deal with high angles of attack and sharp turns typical of small highly-maneuverable UAVs is demonstrated. Nonlinear natural-stabilizing aerodynamic loads are included and employed by the controller. This approach differs from feedback linearization where these forces are first modeled and then canceled, allowing for a less accurate knowledge of the aircraft dynamics. Slower changing outer loop navigation variables, velocity, altitude and heading angle, are controlled by PID gains. This choice allows the designer to maintain a clearer understanding of the control action, limits the required computational power and eases the implementation procedure. In fact, the main purpose of this work is to provide a framework for the actual employment of backstepping control on microcontrollers for small UAVs. Adaptation and a more advanced outer loop design is beyond the scope of this paper.

A constant in most of the backstepping approaches summarized above is its combination with complex adaptation laws. The benefits of nonlinear control supported by advanced adaptation are clear, but the problems of real-time implementation are considerable. To our best knowledge, among the adaptive backstepping solutions previously described, only the work of[25] has been implemented on a microcontroller suitable for small fixed-wing UAVs. The algorithm described in[29], based on adaptive backstepping for directional control in the presence of crosswind, was declared to be under implementation, this effort being aided by the limited number of controlled variables and the simplicity of the adaptation approach. The only flying application of a simple adaptive backstepping controller on a fixed-wing UAV is presented in[30], where basic roll and pitch angles hold is achieved through implementation on a Procerus Kestrel autopilot. The work of[21] has been only tested in open loop to acquire and validate the magnitude of the control signals. Some of the same authors illustrate in[31] the flight test results from a backstepping controlling altitude, roll angle and yaw angle, but very limited implementation and testing details are provided.

In the present paper, an innovative use of microprocessor technology based on cutting-edge transistor computers is employed to support the controller implementation(32). The combination of this tool with the proposed control layout facilitates the passage from theoretical simulation to practical application. In fact, HIL simulations validate the control scheme and real-time operation is achieved with satisfying flight performances. Furthermore, we show the applicability of our backstepping solution for two fixed-wing aircraft having very different configuration and specific properties, also in presence of noise and parametric uncertainties.

3.0 Fixed-wing Aircraft Model

Fixed-wing aircraft dynamics are defined by a six-degree of freedom model. Three sets of differential equations describe the forces and moments acting on the airplane and its orientation with respect to a reference system(34). The force equation in generic CG-centered body axes (X_B, Y_B, Z_B), see Fig. 1, is expressed as:

$$m\dot{\mathbf{V}} = \mathbf{F} - \boldsymbol{\omega} \times m\mathbf{V}, \tag{1}$$

with m aircraft mass, $\mathbf{V} = (u, v, w)^T$ linear velocity vector and $\boldsymbol{\omega} = (p, q, r)^T$ angular velocity vector, all expressed in body axes. The vector $\mathbf{F} = (F_x, F_y, F_z)^T$ represents the sum along X_B, Y_B and Z_B of all forces acting on the aircraft center of mass: aerodynamics forces, engine thrust and gravity force. The moment equation has a similar structure:

$$I\dot{\boldsymbol{\omega}} = \mathbf{M} - \boldsymbol{\omega} \times I\boldsymbol{\omega}, \tag{2}$$

where I is the body-axes inertia matrix. The vector $\mathbf{M} = (M, L, N)^T$ contains the sum of the moments about X_B, Y_B and Z_B generated by aerodynamic forces and engine thrust. The attitude equation is:

$$\dot{\mathbf{\Phi}} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \boldsymbol{\omega}. \tag{3}$$

The vector $\mathbf{\Phi} = (\phi, \theta, \psi)^T$ contains the Euler angles between the body axes and the (X_N, Y_N, Z_N) axes of the North-East-Down (NED) reference system. These angles are called, respectively, roll, pitch and yaw.

The recursive nature of the backstepping controller requires that the equations governing the system may assume a general triangular structure called pure-feedback form(18):

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\xi_1 \\ \dot{\xi}_1 = f_1(\mathbf{x}, \xi_1, \xi_2) \\ \dot{\xi}_2 = f_2(\mathbf{x}, \xi_1, \xi_2, \xi_3) \\ \vdots \\ \dot{\xi}_{k-1} = f_{k-1}(\mathbf{x}, \xi_1, ..., \xi_k) \\ \dot{\xi}_k = f_k(\mathbf{x}, \xi_1, ..., \xi_k, u_b). \end{cases} \tag{4}$$

In Eq. (4) $\mathbf{x} \in \mathbb{R}^n$ is the state vector and $\xi_{1,...,k}$ are scalars denoting other states of the system. The functions f_i ($i = 1, ..., k$) are nonlinear and depend only on $\mathbf{x}$ and on the states ξ_j ($j =$

$1, ..., i + 1$), i.e., they depend at most on the state variable of the upper order subsystem. The scalar u_b is the external controller of the global system; each subsystem represented by the state ξ_l ($l = 1, ..., k - 1$) is controlled by the virtual control input ξ_{l+1}.

Equations (1)-(2)-(3), as such, cannot assume the structure of (4). The cascade form is not respected since $\mathbf{F}$ and $\mathbf{M}$ are function of the states $\mathbf{V}$ and $\boldsymbol{\omega}$, of the aerodynamic angles and of control actions. Nevertheless, under the assumptions described below, it is possible to convert the equations of motion into a suitable form for a limited number of aircraft states: angle of attack α, sideslip angle β and stability-axes roll rate p_s, see Fig. 1. The aim is to design a controller so that $\alpha = \alpha^{ref}$, $p_s = p_s^{ref}$ and $\beta = 0$. Control over angle of attack and roll rate is essential to determine, respectively, the longitudinal behavior and the flight direction. A null sideslip angle is desired in cruise flight to achieve symmetric flight and to reduce aerodynamic drag. The ability of an aircraft to cancel out sideslip angle perturbations is a sign of its lateral-directional static stability.

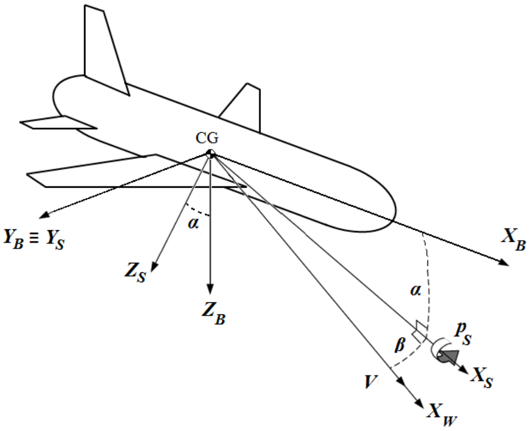


Figure 1. Controlled variables and reference axes at initial time

Stability axes are a particular type of body axes where X_S lies along the projection of the initial $\mathbf{V}$ on the aircraft plane of symmetry, Z_S is positive from the upper to the lower side of the wing airfoil, Y_S completes the right handed reference frame, as shown in Fig. 1. X_S and X_B are separated by the angle of attack, a single rotation of magnitude α about $Y_S \equiv Y_B$ is sufficient to align body axes with stability axes. Such rotation allows the definition of the angular velocities in stability axes $\boldsymbol{\omega}_s = (p_s, q_s, r_s)^T$ as:

$$\boldsymbol{\omega}_s = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \boldsymbol{\omega} = R_{sb}\boldsymbol{\omega}. \tag{5}$$

The dynamics considered for the control design are obtained from the force equation written in wind axes, the complete derivation is available in[(2)]. Wind axes are defined as follows: X_W is aligned with the airspeed direction, Y_W is orthogonal to X_W oriented from left to right with respect to the center of mass trajectory, Z_W lies in the plane of symmetry of the aircraft, directed from the upper to the lower wing airfoil surface. Given this background, the following assumptions are proposed:

- **Assumption 1**: The deflection of the control surfaces only generates a variation in moments, the variation in forces is small enough to be neglected.
- **Assumption 2**: Lift and side force coefficients, C_L and C_Y, only depend on the aerodynamic angles and not on the aerodynamic angle rates of change: $C_L = C_L(\alpha)$, $C_Y = C_Y(\beta)$.

The first assumption is reasonable for aircraft with traditional configuration where control surfaces are far from the aircraft center of gravity(34). The deflection of a control surface generates forces and, as a consequence, moments. The comparison between the control derivative for a force and the one for the resulting moment shows that, in general, the latter has same order of magnitude or is larger. In fact, its definition includes, among other terms, the product between the force derivative and the distance of the control surface from the center of gravity. Furthermore, the addition of a reference lever-arm distance in the moment mathematical formulation, see for instance Eq. (19), increases the moment contribution with respect to the force contribution. Once the trim condition is achieved, the control deflections for maneuver are minimal, reducing to a negligible value the variation of forces so produced. Assumption 2 is considered valid in steady flight or during smooth maneuvers. In fact, the disregarded aerodynamic derivatives $C_{L\dot{\alpha}}$ and $C_{Y\dot{\beta}}$ are originated by the delay in the pressure distribution of the unsteady flow to adjust to sudden attitude variations. Assumption 2 is on the conservative side as it targets progressive maneuvers, the ability of the backstepping architecture to control aggressive flight will be demonstrated.

Given these assumptions, the differential equations governing the variation in time of the controlled variables α, β and p_s are now obtained. By defining $V = |\mathbf{V}| = \sqrt{u^2 + v^2 + w^2}$, the aerodynamic angles are commonly expressed as $\alpha = \arctan(w/u)$ and $\beta = \arcsin(v/V)$. The equations relating the derivatives of the aerodynamic angles with the angular velocities and α and β themselves are:

$$\begin{aligned} \dot{\alpha} &= q - (p\cos\alpha + r\sin\alpha)\tan\beta + \frac{Z\cos\alpha - (X+T)\sin\alpha + mg_2}{mV\cos\beta} \\ \dot{\beta} &= p\sin\alpha - r\cos\alpha + \frac{Y - T\cos\alpha\sin\beta + mg_3}{mV} \end{aligned} \tag{6}$$

with T engine thrust and X, Y, Z aerodynamic forces in body axes. The gravity acceleration components g_2 and g_3 are:

$$\begin{aligned} g_2 &= g(\cos\alpha\cos\theta\cos\phi + \sin\alpha\sin\theta) \\ g_3 &= g(\cos\beta\cos\theta\sin\phi + \sin\beta\cos\alpha\sin\theta - \sin\alpha\sin\beta\cos\theta\cos\phi) \end{aligned} \tag{7}$$

with $g = 9.81$ m/s^2 gravity acceleration. Eq. (6) can be written in a more compact and meaningful form. Since:

$$Lift = X\sin\alpha - Z\cos\alpha$$

the lift force $Lift$ is included in the $\dot{\alpha}$ equation and thanks to Eq. (5) the stability-axes angular rates are introduced in the $\dot{\alpha}$ and $\dot{\beta}$ dynamics. The result is:

$$\begin{aligned} \dot{\alpha} &= q_s - p_s\tan\beta + \frac{-Lift - T\sin\alpha + mg_2}{mV\cos\beta} \\ \dot{\beta} &= -r_s + \frac{Y - T\cos\alpha\sin\beta + mg_3}{mV} \end{aligned} \tag{8}$$

The backstepping controller is designed to directly control the stability-axes angular velocities $\boldsymbol{\omega}_s$ through the control vector $\mathbf{u}_c = (u_1, u_2, u_3)^T$. Therefore, the dynamics of the stability-axes angular velocities are described by the relationship $\dot{\boldsymbol{\omega}}_s = \mathbf{u}_c$. Combining this formulation with Eq. (8) gives:

$$\begin{cases} \dot{p}_s = u_1 \\ \dot{\alpha} = q_s - p_s \tan\beta + \dfrac{-Lift - T\sin\alpha + mg_2}{mV\cos\beta} \\ \dot{q}_s = u_2 \\ \dot{\beta} = -r_s + \dfrac{Y - T\cos\alpha\sin\beta + mg_3}{mV} \\ \dot{r}_s = u_3 \end{cases} \tag{9}$$

Note that $Lift$ depends on the angle of attack through the coefficient $C_L = C_L(\alpha)$ and that Y depends on the sideslip angle through the coefficient $C_Y = C_Y(\beta)$. The thrust T is considered independent from the aerodynamic angles.

4.0 Control Design

This section illustrates the main theoretical steps followed to design the backstepping controller and the architecture adopted for the full aircraft control.

4.1 Backstepping Controller Design

In order to simplify the controller design, an additional set of assumptions is proposed:

- **Assumption 3**: The time derivatives of speed V, altitude h and heading ψ can be neglected as they have a slower rate of change compared to the controlled variables α, β, and p_s.
- **Assumption 4**: Actuators have rapid enough dynamics, thus they can be ignored in the design process.

Assumptions 3 is mainly valid for cruise flight and progressive maneuvers, when a controlled change in the aircraft equilibrium has a primary effect on the faster attitude dynamics and a secondary one on the navigation variables. Finally, Assumption 4 is very common and generally reasonable provided that Assumptions 2 and 3 are respected.

Eq. (9) is not suitable for the application of a total backstepping controller because the cascade form is not respected, in particular due to the presence of β in the α dynamics, and vice versa. However by separating its dynamics as:

$$\dot{p}_s = u_1 \tag{10}$$

$$\begin{cases} \dot{\alpha} = q_s - p_s \tan\beta + \dfrac{-Lift - T\sin\alpha + mg_2}{mV\cos\beta} \\ \dot{q}_s = u_2 \end{cases} \tag{11}$$

$$\begin{cases} \dot{\beta} = -r_s + \dfrac{Y - T\cos\alpha\sin\beta + mg_3}{mV} \\ \dot{r}_s = u_3 \end{cases} \tag{12}$$

three sub-controllers stabilizing the desired states α, β and p_s can be defined. Cross-coupling exists due to the presence of p_s and β in the α dynamics and, at the same time, to the presence of α in the β dynamics. During the individual sub-controls design, β and p_s are imposed constant in the α controller and α constant in the β controller. On the contrary, during the simultaneous control of the three variables, this assumption is disregarded because not physically realistic and not necessary, as it will be later shown. Because of this coupling, the computation of a control action considers, at each moment, the value of the state controlled by another control action. For instance, the control law defining u_2 is evaluated with the instantaneous value of β controlled by u_3. This solution is beneficial when dealing with maneuvers where strong coupling exists between longitudinal and lateral-directional planes.

A simple proportional controller is chosen for p_s, Eq. (10), while the cascade form of Eq. (11) and Eq. (12) allows the application of a backstepping controller for α and β. Note that Eq. (11) and (12) have similar structure:

$$\begin{cases} \dot{\omega}_1 = f(\omega_1, y) + \omega_2 \\ \dot{\omega}_2 = u_s \end{cases} \tag{13}$$

A single backstepping controller designed for Eq. (13) is therefore also suitable for Eq. (11) and (12). As it is preferable to have the origin as the desired equilibrium point, a change of variables is therefore defined:

$$\begin{aligned} x_1 &= \omega_1 - H \\ x_2 &= \omega_2 + f(H, y) \\ \Omega(x_1) &= f(x_1 + H, y) - f(H, y) \end{aligned}$$

where H is the reference value for the controlled variable. The resulting dynamics are:

$$\begin{cases} \dot{x}_1 = \Omega(x_1) + x_2 \\ \dot{x}_2 = u_s \end{cases} \tag{14}$$

The external control input u_s controls x_2 that, in cascade, acts as virtual control to stabilize x_1. Table 1 summarizes the relationships between the variables used in the new and in the original systems. Functions $f_\alpha(\alpha, y_\alpha)$ and $f_\beta(\beta, y_\beta)$ are:

$$\begin{aligned} f_\alpha(\alpha, y_\alpha) &= -p_s \tan\beta + \frac{-Lift - T\sin\alpha + mg_2}{mV\cos\beta} \\ f_\beta(\beta, y_\beta) &= \frac{Y - T\cos\alpha\sin\beta + mg_3}{mV} \end{aligned}$$

As shown in[(2)], a linear globally stabilizing control law for the system of Eq. (14) is:

$$u_s = -k_2(x_2 + k_1 x_1)$$

with $k_2 > 2k_1 > \max\{0, k_u\}$. Using the relationships of Table 1, the control laws for Eq. (11) and (12) are:

$$\begin{aligned} u_2 &= -k_{\alpha,2}\left(q_s + k_{\alpha,1}\left(\alpha - \alpha^{ref}\right) + f_\alpha(\alpha^{ref}, y_\alpha)\right) \\ u_3 &= k_{\beta,2}\left(-r_s + k_{\beta,1}\beta + f_\beta(0, y_\beta)\right) \end{aligned} \tag{15}$$

Table 1
Change of variable relationships

General system	Longitudinal	Lateral-directional
ω_1	α	β
ω_2	q_s	$-r_s$
u_s	u_2	$-u_3$
y	$p_s, \beta, V, h, \theta, \phi$	$\alpha, V, h, \theta, \phi$
$f(\omega_1, y)$	$f_\alpha(\alpha, y_\alpha)$	$f_\beta(\beta, y_\beta)$
H	α^{ref}	0
x_1	$\alpha - \alpha^{ref}$	β
x_2	$q_s + f_\alpha(\alpha^{ref}, y_\alpha)$	$-r_s + f_\beta(0, y_\beta)$
$\Omega(x_1)$	$f_\alpha(\alpha, y_\alpha) - f_\alpha(\alpha^{ref}, y_\alpha)$	$f_\beta(\beta, y_\beta) - f_\beta(0, y_\beta)$

with:

$$\begin{aligned} &k_{\alpha,2} > 2k_{\alpha,1}, \quad k_{\alpha,1} > \max\{0, k_\alpha\}, \quad k_\alpha = \max_{\alpha, y_\alpha} \frac{\partial f_\alpha(\alpha, y_\alpha)}{\partial \alpha} \\ &k_{\beta,2} > 2k_{\beta,1}, \quad k_{\beta,1} > \max\{0, k_\beta\}, \quad k_\beta = \max_{\beta, y_\beta} \frac{f_\beta(\beta, y_\beta) - f_\beta(0, y_\beta)}{\beta} \end{aligned} \tag{16}$$

Finally, the proportional control adopted for p_s is:

$$u_1 = k_{p_s}\left(p_s^{ref} - p_s\right), \quad k_{ps} > 0. \tag{17}$$

The relation between the control inputs and the stability-axes angular accelerations is defined by $\mathbf{u}_c = (u_1, u_2, u_3)^T = \dot{\boldsymbol{\omega}}_s$. The angular accelerations are the result of the variation in moments originated primarily by the deflection of the aircraft control surfaces. The vector of deflections $\boldsymbol{\delta}$ is obtained from the moment equation:

$$\mathbf{M}(\boldsymbol{\delta}) = I\left(R_{sb}^T \mathbf{u_c} + \dot{R}_{sb}^T \boldsymbol{\omega}_s\right) + \boldsymbol{\omega} \times I\boldsymbol{\omega} \tag{18}$$

To calculate $\boldsymbol{\delta}$, a control strategy matching the controlled variables with the aircraft control surfaces must be defined.

4.2 Control Strategy

The controller described above stabilizes three variables related to the aircraft attitude. A global autopilot configuration capable of controlling speed V, altitude h and heading ψ is required. In real-life implementation, these variables could be easily measured with, respectively, a pitot tube, a barometric pressure sensor and magnetometer. The control strategy is defined as follows: the backstepping controller acts on α, β and p_s in the inner loop, while three PID controllers act on V, h and ψ in the outer loop. This approach separates the fast dynamics, characterizing attitude, from the slower dynamics, characterizing navigation. The prompt response of the backstepping controller is necessary when dealing with attitude variables which are of prime importance for the aircraft safety. For instance, an immediate control of α in presence of vertical gusts could prevent the stall or dangerous flight regimes. Consistently with Assumption 3, slower navigation variables can be successfully handled using traditional PIDs, whose gains are tuned manually with an heuristic approach. The goal is to optimize the response in terms of overshoot, rise time, settling time and ringing.

The PID controlling the speed feeds the backstepping controller with the desired angle of attack, while the PID controlling the heading defines the desired roll rate. These values are limited in magnitude in order to avoid the request of a motion incompatible with the aircraft dynamics during sudden maneuvers. Saturation is imposed at the stall angle of attack for α^{ref} and at a typical roll rate for p^{ref}. Note that the desired roll rate p^{ref} is expressed in body axes, the conversion to stability axes p_s^{ref} is performed with Eq. (5). The control surfaces employed are the elevator δ_e, the aileron δ_a and the rudder δ_r. According to Assumption 1, these only generate a variation in moments and not in forces. The deflection vector $\boldsymbol{\delta} = (\delta_e, \delta_a, \delta_r)^T$ is obtained substituting in Eq. (18) the most general expressions of the moments:

$$\begin{aligned}
L(\delta_a, \delta_e, \delta_r) &= \frac{1}{2}\rho V^2 S\, b\left(C_{l\beta}\beta + C_{l\dot{\beta}}\dot{\beta} + C_{lp}\hat{p} + C_{lr}\hat{r} + C_{l\delta_a}\delta_a + C_{l\delta_e}\delta_e + C_{l\delta_r}\delta_r\right)\\
M(\delta_a, \delta_e, \delta_r) &= \frac{1}{2}\rho V^2 S\, c\left(C_{m0} + C_{m\alpha}\alpha + C_{m\dot{\alpha}}\dot{\alpha} + C_{mq}\hat{q} + C_{m\delta_a}\delta_a + C_{m\delta_e}\delta_e + C_{m\delta_r}\delta_r\right) \qquad (19)\\
N(\delta_a, \delta_e, \delta_r) &= \frac{1}{2}\rho V^2 S\, b\left(C_{n\beta}\beta + C_{n\dot{\beta}}\dot{\beta} + C_{np}\hat{p} + C_{nr}\hat{r} + C_{n\delta_a}\delta_a + C_{n\delta_e}\delta_e + C_{n\delta_r}\delta_r\right)
\end{aligned}$$

and solving the resulting linear system with three equations and three unknowns. The nondimensional angular rates $\hat{p}$, $\hat{q}$ and $\hat{r}$ are typically defined as:

$$\hat{p} = \frac{pb}{2V}, \quad \hat{q} = \frac{qc}{2V}, \quad \hat{r} = \frac{rb}{2V}$$

and ρ is the air density, b is the aircraft wingspan, c the mean aerodynamic chord and S the wing area. The aerodynamic derivatives are $C_{m0}, C_{m\alpha}, C_{m\dot{\alpha}}, C_{mq}, C_{l\beta}, C_{l\dot{\beta}}, C_{lp}$, C_{lr}, $C_{n\beta}, C_{n\dot{\beta}}, C_{np}$ and C_{nr}, while the control derivatives are $C_{m\delta_a}$, $C_{m\delta_e}$, $C_{m\delta_r}$, $C_{l\delta_a}$, $C_{l\delta_e}$, $C_{l\delta_r}$, $C_{n\delta_a}$, $C_{n\delta_e}$ and $C_{n\delta_r}$. Note that commonly the contribution of $C_{m\delta_a}$, $C_{m\delta_r}$, $C_{l\delta_e}$ and $C_{n\delta_e}$ is very small or zero. In this case, the calculation of the commands is more simple: δ_e is found from the $M(\delta_e)$ equation, while δ_a and δ_r are found solving the linear system with $L(\delta_a, \delta_r)$ and $N(\delta_a, \delta_r)$. The engine thrust vector is considered aligned with the aircraft X_B axis and so it does not generate moments.

The third PID controls the altitude by defining the required throttle value δ_{th} independently of the backstepping controller which acts through angular rates. The outer loop strategy, where control surfaces, in practice the elevator, control airspeed and the throttle controls altitude is a standard autopilot mode. As explained in[35], this approach guarantees better tracking of the airspeed which is a key flight safety parameter. Table 2 summarizes the controlled variables, their commands and the control method.

Table 2
Relationship between variables and commands

Outer loop	Inner loop	Command	Control method
V	α	δ_a, $\boldsymbol{\delta_e}$, δ_r	Backstepping + PID
h	–	δ_{th}	PID
ψ	p, β	$\boldsymbol{\delta_a}$, δ_e, $\boldsymbol{\delta_r}$	Backstepping + PID

The proposed control scheme is shown in Fig. 2. The computed control inputs act on the aircraft and its measured controlled states, total speed, altitude and heading angle, are the

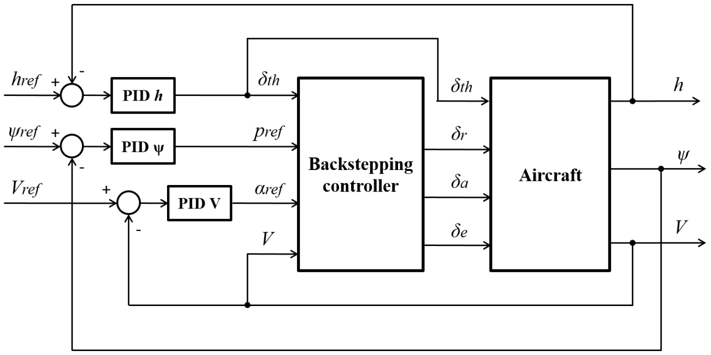


Figure 2. Backstepping control strategy for fixed-wing aircraft

feedback variables. Their differences with the corresponding reference values, V^{ref}, h^{ref} and ψ^{ref}, define the error inputs for the PIDs. The throttle command and the measured speed are given as input to the backstepping controller as required by the control law definition and for the estimation of the inner loop states. Note, in fact, that the variables α, β and p_s, used for the definition of the inner loop error, are estimated with good degree of accuracy inside the backstepping controller by integrating Eq. (10)-(12), as later shown in Fig. 5.1. A support to the accurate estimation of α and β can be provided by the feedback of ϕ and θ, easily measurable with an IMU. These values appear in Eq. (7) for the calculation of g_2 and g_3. The reason for this unconventional solution lies in the intention of implementing and testing the backstepping controller on a real aircraft. The possibility to effectively estimate these variables much simplifies the structure of the autopilot system and significantly reduces the development time and cost. The need for a measure of α and β would be undermined by the lack of affordable, reliable and compact aerodynamic angles sensors suitable for small UAVs.

5.0 Numerical Simulations

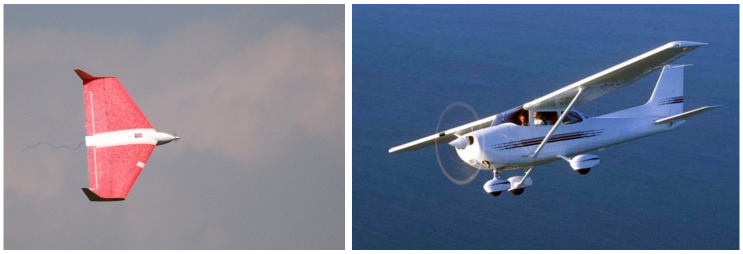
Two sets of numerical simulations are illustrated in this section. The first one is performed in Simulink and demonstrates the capability of the proposed solution to accommodate aircraft different in size and configuration, also in presence of noise and parametric uncertainties. The second set tests the C code implementation of the controller according to a software-in-the-loop (SIL) approach. It is performed on a single aircraft through FlightGear simulator(33) and demonstrates the ability of the controller to work in real-time at slow sampling rates.

5.1 Simulink Simulations

Simulink simulations are performed integrating the equations through a second order Heun method with 0.01 seconds time step, continuous time blocks are employed. The block scheme follows the structure of Fig. 2. The nonlinear equations of motion of(34) are adopted in the aircraft block. Here, actuators transfer functions as from(36) and a simplified linear motor model are included. Standard continuous time Simulink PID blocks are employed. Note that these blocks contain a low pass filter in the derivative action, $D\frac{C_{PID}}{1+C_{PID}/s}$ with D derivative gain and s complex variable. The default coefficient value of $C_{PID} = 100$ is maintained.

The controller is applied to two self-developed nonlinear models representing the MH850

UAV and the Cessna 172P aircraft, see Fig. 3. The MH850 is characterized by tailless configuration, electric propulsion and non-movable vertical fins at wingtips[37]. The wingspan is 85 cm, the mass 1 kg and the cruise speed 15 m/s. Aerodynamic control is achieved with elevons, they control longitudinal motion when symmetrically deflected and lateral-directional motion when antisymmetrically deflected. A numerically-derived database including of all aerodynamic derivatives is available to build the nonlinear aircraft model[39] -[38]. The Cessna 172P is a single combustion engine aircraft with standard configuration including high-wing and fixed tricycle landing gear. Take off weight is around 880 kg and wingspan 11 m. The aircraft is powered by a Lycoming O-320-D2J engine able to produce 160 hp and to guarantee a cruise speed of 60 m/s. The control surfaces are aileron, elevator and rudder. Its choice is motivated by two reasons: it is a popular aircraft with much technical data available; a detailed aircraft model is available in FlightGear. The two aircraft differ considerably not only in terms of absolute weight, dimension and power. Relative characteristics of the C172P, such as power-to-weight ratio and wing loading, are poorer than those of the large majority of small UAVs, see Table 3. Testing the controller on a lower-performance platform allows to prove its universality and to identify its limits.



(a) MH850 UAV

(b) Cessna 172P

Figure 3. Aircraft employed for the numerical simulations

Table 3
Aircraft specific properties

Aircraft	Wing loading [kg/m^2]	Power-to-weight ratio [W/kg]
MH850	4	170
C172P	54	134

It is interesting to explain how the calculation of the commands for the MH850 rudderless configuration is performed. As already pointed out, the δ_e command is found from the $M(\delta_e)$ equation as $C_{m\delta_a} = 0$. Both L and N moments are function of the remaining δ_a command, this generates an overdetermined system of two equations with one unknown. It is chosen to disregard the $N(\delta_a)$ equation and to obtain δ_a from $L(\delta_a)$. This is motivated by the strong predominance of the rolling moment over the yawing moment in case of aileron deflection, for the MH850 being $C_{l\delta_a} \approx 10 \cdot C_{n\delta_a}$.

Initially the MH850 response to contemporary step inputs is tested. Reference values are arbitrarily defined as $V^{ref} = 17$ m/s, $h^{ref} = 120$ m and $\psi^{ref} = 30°$, typical figures expected in standard operations. Longitudinal and lateral-directional commands are applied at the same time. The outer loop responses are represented in Fig. 5.1 and demonstrate the capability of the controller to effectively achieve good tracking and short settling time. Although no rudder is used, the response on ψ is still satisfying with just aileron control. Fig. 5.1 and 5.1 respectively show the inner loop responses and the commands. Each of the inner loop plots includes the reference value, the state estimated within the backstepping controller and the aircraft state. The accurate velocity tracking is achieved thanks to an excellent control of the angle of attack in the inner loop. In this case, α is bounded to $\pm 12°$ in order to avoid near-stall conditions. Sideslip angle β shows some oscillations originating during the step transition. The limited directional damping provided by the vertical fins at the wingtips might be responsible for this. In any case, the magnitude of the oscillations is minimal, with a peak smaller than 0.4 degrees, barely noticeable in flight. The elevons deflection always remains within the 20° maximum value, throttle saturation is measured only for few seconds after the step input start.

On the same plots, the comparison with a well-tuned PID controller is proposed. The outer loop PID gains remain unchanged, while the backstepping controller is replaced by two inner loop PIDs. The first one determines δ_e according to the pitch angle error, the reference θ being the output of the outer PID on V. The second one defines δ_a based on the roll angle error, the reference ϕ is obtained from the outer PID on ψ. A similar configuration, commonly employed in commercial autopilots, was illustrated in[(40)]. In the outer loop, the PID performance is almost comparable to backstepping, the V response is slightly more oscillatory while the ψ response is slower and has larger overshoot. Similarly, no significant difference is observable in the inner loop. Instead, the commands δ_e and δ_a from the PID control show a higher oscillatory behavior, altitude and throttle remain basically unchanged.

Figure 4. Simulink responses for MH850

A validation of the controller robustness to aircraft parametric uncertainties is performed with two test cases. In both of them significant variations in aircraft mass, inertia and static margin are introduced. These parameters are altered in the nonlinear aircraft model while the controller is unchanged. Case 1 contemplates a heavier aircraft, with higher inertia and with a reduced static margin, so that the derivatives $C_{m\alpha}$ and $C_{m\delta_e}$ are weaker. In Case 2, the aircraft is lighter, has lower inertia and its center of gravity is moved forward, so that the magnitude of the derivatives is higher. The variation in m, I, $C_{m\alpha}$ and $C_{m\delta_e}$ is $\pm 30\%$ from the nominal values. Fig. 5 shows the obtained results in comparison with the nominal case. In the outer loop response of Fig. 5.1, V and ψ, which are indirectly controlled via backstepping, remain almost unchanged. Slightly higher oscillations in V are observable for Case 1 due to a lower pitch damping. The altitude response, controlled with throttle solely through PID, suffers stronger variations from the nominal case. As expected, the aircraft with higher mass and inertia has a slower response to step input, higher overshoot and settling time. In the inner loop, see Fig. 5.1, lateral oscillations are increased in amplitude in Case 1. It is interesting to observe how the α trim values change in the two cases. The commands plot of Fig. 5.1 confirms that full throttle command is required longer for the heavier aircraft of Case 1. Meaningful step response parameters such as overshoot os, rise time t_r and settling time t_s are listed in Table 4. As explained in Section 4.2, the goal of the heuristic approach used to tune the controllers, both PID and backstepping, is to minimize their values.

Figure 5. Simulink responses for MH850 in presence of uncertainties

The PID controller previously introduced is tested for the same perturbed aircraft configurations, results are presented in Fig. 6. It is evident that the nominal PID controller is not able

Table 4
Step response parameters for nominal backstepping, PID and backstepping in presence of uncertainties; t_r and t_s are in seconds, os has the unit of the considered variable.

	Nominal			**PID**			**Case 1**			**Case 2**		
	t_r	os	t_s	t_r	os	t_s	t_r	os	t_s	t_r	os	t_s
V	0.49	0.54	6.67	0.66	0.61	9.91	0.54	0.59	9.67	0.45	0.51	6.65
h	3.16	5.7	25.9	3.1	5.5	25.7	4.46	7.5	29.1	2.3	4.2	23.9
ψ	4.88	0.81	10	7.78	2.47	25.6	4.98	0.47	7.25	5.04	0.64	14.8

to withstand the uncertainties introduced in both cases. The aircraft loses directional control and it accelerates while quickly losing altitude, crashing to the ground in less than 10 seconds. Despite the backstepping and PID controllers being almost equivalent in the nominal case, it is clear that a traditional PID configuration is not able to deal with substantial changes in the aircraft parameters. On the contrary, the backstepping controller has proved to be robust as it guarantees satisfying performance in all cases.

Figure 6. Simulink responses for MH850 with PID controller in presence of uncertainties

In order to demonstrate the ability of the backstepping controller to withstand sensor noise, the same simulation is performed including this disturbance. White Gaussian noise is introduced on velocity, altitude and heading angle measurements. Noise characteristics are based on real data from available sensors: a pitot tube with standard deviation $\sigma_V = 0.3$ m/s, a barometric pressure sensor with $\sigma_h = 0.5$ m and an magnetometer with $\sigma_\psi = 1$ deg. A Kalman filter is applied to each noisy feedback variable to mitigate the effect of the disturbance. The simulation results appear in Fig. 7. A comparison with Fig. 4 shows that the aircraft response is equivalent, in particular for the outer loop variables. Here the velocity is the state most influenced by noise but it still shows a satisfying response. In fact, when steady state is achieved, the standard deviation is just 0.074 m/s. The inner loop is more affected by noise because of the derivative operations in the PID controller. This causes α^{ref} and p_s^{ref} to sustain high frequency oscillations which, on the contrary, are not present in the values of α and p_s estimated within the backstepping controller. It is necessary to point out that the PID gains for the noisy example have been slightly adjusted compared to the noise-free case. The major change is the reduction of the derivative gains for the V and h loops. Note that without any adjustment the controller would still guarantee satisfying reference tracking, even if with a more disturbed response. The calculated commands of Fig. 5.1, in particular the elevator, are also affected by noise, but they still remain compatible with the actuators dynamic response.

Figure 7. Simulink responses for MH850 in presence of noise

The C172P responses to ramp inputs are shown in Fig. 5.1. Excellent tracking performance is achieved, in particular for the speed, as previously observed for the MH850. The different nature of the reference signal is motivated by the different responses expected from the two airplanes: aggressive for the UAV and progressive for the Cessna. A more aggressive request to the C172P, for instance a higher climbing rate, would still result in zero altitude steady-state error, but with a larger deviation in the climbing phase. This is not due to a problem with the controller, but to the lack of power of the C172P. In the C172P case, the presence of the rudder command guarantees excellent heading angle tracking. The less demanding references generate a smoother behavior of the inner loop variables, as shown in Fig. 5.1. Note that the steady 0.4° error in the α tracking is motivated by the effect that the propeller-generated induced velocity has on the elevator. The different flow velocity on the elevator changes its moment contribution to the aircraft equilibrium and so it changes the trim angle of attack. This phenomenon is included in the simulated aircraft model but not in the simplified backstepping controller aircraft scheme. Saturation is observed for the elevator and the throttle, Fig. 5.1.

Figure 8. Simulink responses for C172P

5.2 Software-in-the-loop Simulations

As first step to achieve real-time implementation on a microcontroller board, the control law is implemented in C code and applied to the C172P model existing within FlightGear simulator, the aircraft having same features described in Section 5.1. The adopted architecture is represented in Fig. 9.

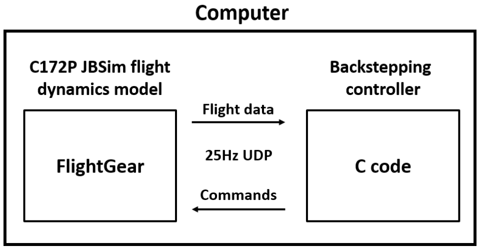


Figure 9. SIL layout

FlightGear is a freeware open-source flight simulator developed by volunteers around the world and it offers to academic developers an experienced tool to test their aircraft models and control laws, see for instance[41,43]. FlightGear version 2.6.0 is used and the JSBSim flight dynamics library is employed. JSBSim is an open source flight dynamics model defining the six-degree of freedom equations which characterize the aircraft motion. Data transfer between the C application and FlightGear is performed via UDP. FlightGear provides the value of the feedback variables V, h and ψ, the backstepping controller returns the commands δ_e, δ_a, δ_r and δ_{th}. A 25 Hz frequency is chosen in order to guarantee a consistent data rate compatible with real sensors, integration of the equations is performed with a second order Heun method running at 100 Hz.

Fig. 10 shows the results of a complex maneuver. The aircraft is requested first to climb and turn while accelerating, then to maintain the speed while climbing and turning more aggressively, finally to decelerate while performing another turn and rapidly losing altitude. All variables are tracked with good accuracy in every phase of the maneuver. Speed control performs the best, the quick response is guaranteed by the choice of using the elevator instead of the throttle. Similarly, the heading angle shows good results despite some mild overshoot. The altitude response is penalized by some overshoot/undershoot and some mild oscillations in the settling phase. The difference in slope between reference and actual values is caused by the slower engine response and the low power-to-weight ratio. In this paper, priority is given to speed tracking which is crucial to avoid stall. Finally, it is interesting to observe how the changes in altitude affect the speed. The commands are plotted in Fig. 11. The surface deflections always remain well within the saturation limits, 20° for elevator and aileron, 16° for rudder, while a rate limiter is imposed. The motor instead goes full throttle during the climbing phases.

Figure 10. FlightGear SIL simulated maneuver for C172P

Figure 11. FlightGear SIL simulated maneuver commands for C172P

6.0 Hardware-in-the-loop Simulations

The controller is implemented on a control board and tested in real time with FlightGear simulator. The chosen microcontroller is the XMOS XK-1A development board, a technology by XMOS Ltd(44). This board is characterized by a multi-core processor able to perform eight real-time tasks in parallel. Its parallel computing ability is essential for unmanned applications where high level tasks, for instance the control logic, have to be combined with low level assignments, such as I/O(32). An advantage in using the XMOS technology is the ease of programming the board. The coding language is called XC, the strong similarity with C allows a fast and simple implementation. The low cost of the board, its limited weight (19 g) and dimensions (50 x 50 mm) make it suitable for small UAV applications.

Figure 12. HIL setup

The hardware-in-the-loop (HIL) setup and scheme are represented in Fig. 12 and Fig. 13 respectively. The simulator does not communicate directly with the controller, a bridge application is placed between FlightGear and the board. Its role is to capture flight data arriving from FlightGear through UDP and send them over serial to the board; at the same time it acquires the serial commands from the board and sends them to FlightGear via UDP. The UDP data rate is maintained at 25 Hz. For serial communication the baudrate of 153600 Bd is chosen to maximize data transmission speed and avoid the overlapping of send and receive tasks. The controller computation time is slightly less than 0.004 seconds on the average. This result is obtained thanks to the second order Heun integration method and to the simplicity of the operation performed by the control law.

HIL simulations are performed with the same reference variables tested for the SIL simulations, backstepping and PID gains are unchanged. Fig. 14 validates the real-time implementation, the tracking achieved with the microcontroller is accurate and virtually identical to what is obtained in the SIL case. The commands for this simulation are represented in Fig. 15, an excellent matching with the commands from the SIL is evident.

The controller has been integrated on an Ultra Stick 25e RC aircraft model, Fig. 16, and preliminary validation flight tests have been conducted(45). The upgrade of the controller is under consideration. The initial control strategy was deliberately kept simple in order to demonstrate the feasibility of the real-time implementation and to perform flight tests. Adaptive backstepping and substitution of the PIDs with more advanced laws are the changes investigated.

Figure 13. HIL layout

Figure 14. FlightGear HIL simulated maneuver for C172P

7.0 Conclusions

In this paper an autopilot configuration combining nonlinear control with traditional PID technique is presented. The backstepping controller is employed to stabilize fast inner loop variables characterizing the aircraft attitude and aerodynamic angles, while PID gains control slower changing navigation variables. Backstepping method is chosen for its ability to deal with the nonlinearities that characterize small fixed-wing UAV dynamics. This method requires a fairly rich knowledge of the aircraft characteristics, but in return it ensures good performance over a large flight envelope.

The adopted backstepping approach guarantees simultaneous control of the longitudinal and lateral-directional planes. Through numerical simulations it is demonstrated that the proposed solution satisfactorily controls aircraft different in size and configuration, also in presence of parametric uncertainties and noise. Despite some conservative assumptions in the design process targeting smooth and progressive maneuvers, it is demonstrated that, aircraft allowing, aggressive flight is achievable. Complex maneuvers characterized by severe coupling are performed with limited tracking error. The simple solution adopted differs from the standard adaptive backstepping approaches popular in literature, but it guarantees simple implementation and low computational power without loss of efficacy or robustness. In fact, a real-time implementation on an autopilot board is validated and its performance on complex maneuvers is satisfying. The control strategy here described is believed to be implementable on any microcontroller board for small UAV application.

REFERENCES

1. HÄRKEGRD, O. Backstepping Designs for Aircraft Control - What is there to gain?, Linköping Universitet technical report LiTH-ISY-R-2339, 2001.
2. HÄRKEGRD, O. Flight Control Design Using Backstepping, Linköping Universitet, 2001.
3. HÄRKEGRD, O. and TORKEL GLAD, S. Flight Control Design Using Backstepping, *IFAC Proceedings Volumes*, July 2001, **34**, (6), pp 283 - 288.
4. MICHAILIDIS, M.G., VALAVANIS, K.P. and RUTHERFORD, M.J. Nonlinear Control of Fixed-Wing UAVs with Time-Varying and Unstructured Uncertainties, *Springer Tracts in Autonomous Systems*, Springer, 2020.
5. PIXHAWK The open standards for drone hardware, *www.pixhawk.org*, September 2020.
6. COMINOS, P. and MUNRO, N. PID controllers: recent tuning methods and design to specification, *IEEE Proceedings - Control Theory and Applications*, 2002, **149**, (1), pp 46-53.
7. ASTRÖM, K.J. and HÄGGLUND, T., PID Controllers: Theory, Design, and Tuning, 1995, Instrument Society of America.
8. POKSAWAT, P., WANG, L. and MOHAMED, A., Gain Scheduled Attitude Control of Fixed-

Figure 15. FlightGear HIL simulated maneuver commands for C172P

Figure 16. Controller integration and flight tests of the Ultra Stick 25e RC aircraft

Wing UAV With Automatic Controller Tuning, *IEEE Transactions on Control Systems Technology*, 2018, **26**, (4), pp 1192-1203.

9. Kang, Y. and Hedrick, J.K., Linear Tracking for a Fixed-Wing UAV Using Nonlinear Model Predictive Control, *IEEE Transactions on Control Systems Technology*, 2009, **17**, (5), pp 1202-1210.

10. Santoso, F., Liu, M. and Egan, G.K. H2 and H-infinity robust autopilot synthesis for longitudinal flight of a special unmanned aerial vehicle: a comparative study, *IET Control Theory & Applications*, 2008, **2**, (7), pp 1751-8644.

11. Castañeda, H., Salas-Peña, O.S. and de León-Morales, J. Extended observer based on adaptive second order sliding mode control for a fixed wing UAV, *ISA Transactions*, 2017, **66**, pp 226 - 232.

12. Stastny, T.J., Dash, A. and Siegwart, R. Nonlinear MPC for Fixed-wing UAV Trajectory Tracking: Implementation and Flight Experiments, *AIAA Guidance, Navigation, and Control Conference*, 2017, pp 1-14.

13. Oettershagen, P., Melzer, A., Leutenegger, S., Alexis, K. and Siegwart, R. Explicit model predictive control and $\mathcal{L}_1$-navigation strategies for fixed-wing UAV path tracking, 22nd *Mediterranean Conference on Control and Automation*, 2014, pp 1159-1165.

14. Jung, D.Y, Levy, E.J., Zhou, D., Fink, R., Moshe, J., Earl, A. and Tsiotras, P. Design and Development of a Low-Cost Test-Bed for Undergraduate Education in UAVs, *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp 2739-2744.

15. Miller, J.A., Minear, P.D., Niessner, A.F., DeLullo, A.M., Geiger, B.R., Long, L.N. and Horn, J.F. Intelligent Unmanned Air Vehicle Flight Systems, *Journal of Aerospace Computing, Information, and Communication*, 2007, **4**, (5), pp 816-835.

16. Johnson E., Turbe, M., Wu, A., Kannan, S. and Neidhoefer, J. Flight Results of Autonomous Fixed-Wing UAV Transitions to and from Stationary Hover, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, pp 1-24.

17. Gu, Y., Seanor, B., Campa, G., Napolitano, M.R., Rowe, L., Gururajan, S., and Wan, S. Design and Flight Testing Evaluation of Formation Control Laws, *IEEE Transactions on Control Systems Technology*, 2006, **14**, (6), pp 1105-1112.

18. Krstić, M., Kanellakopoulos, I. and Kokotović, P. Nonlinear and Adaptive Control Design, 1995, Wiley.

19. Kim, K-S. and Kim, Y. Robust backstepping control for slew maneuver using nonlinear tracking function, *IEEE Transactions on Control Systems Technology*, 2003, **11**, (6), pp 822-829.

20. Ju, H. and Tsai, C. Longitudinal axis flight control law design by adaptive backstepping, *IEEE Transactions on Aerospace and Electronic Systems*, 2007, **43**, (1), pp 311-329.

21. Espinoza, T., Dzul, A., Lozano, R. and Parada, P. Backstepping Sliding mode controllers applied to a fixed-wing UAV, *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, pp 95-104.

22. Gavilan, F., Acosta, J.. and Vazquez, R. Control of the longitudinal flight dynamics of an UAV using adaptive backstepping, *18th IFAC World Congress*, 2011, **44**, (1), pp

1892-1897.

23. Liu, K., Zhu, J. and Yu, B. Longitudinal controller design for a fighter aircraft using $\mathcal{L}_1$ adaptive backstepping, *9th World Congress on Intelligent Control and Automation*, 2011, pp 341-346.

24. Ren, W. and Atkins, E. Nonlinear Trajectory Tracking for Fixed Wing UAVs via Backstepping and Parameter Adaptation, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005, pp 1-11.

25. Jung, D. and Tsiotras, P. Bank-to-Turn Control for a Small UAV using Backstepping and Parameter Adaptation, *17th IFAC World Congress*, 2008, **41**, (2), pp 4406-4411.

26. Nørgaard, M.E., Hansen, S., Breivik, M. and Blanke, M. Performance Comparison of Controllers with Fault-Dependent Control Allocation for UAV, *Journal of Intelligent & Robotic Systems*, 2017, **87**, pp 187-207.

27. Lee, T. and Kim, Y. Nonlinear Adaptive Flight Control Using Backstepping and Neural Networks Controller, *Journal of Guidance, Control, and Dynamics*, 2001, **24**, (4), pp 675-682.

28. Sonneveldt, L., Chu, Q.P. and Mulder, J.A. Nonlinear Flight Control Design Using Constrained Adaptive Backstepping, *Journal of Guidance, Control, and Dynamics*, 2007, **30**, (2), pp 322-336.

29. Brezoescu, A., Espinoza, T., Castillo, P. and Lozano, R., Adaptive Trajectory Following for a Fixed-Wing UAV in Presence of Crosswind, *Journal of Intelligent & Robotic Systems*, 2013, **69**, (1-4), pp 257-271.

30. Matthew, J.S., Knoebel, N.B., Osborne, S.R., Beard, R.W. and Eldredge, A. Adaptive backstepping control for miniature air vehicles, 2006 American Control Conference, pp 1-6

31. Espinoza-Fraire, T., Dzul, A., Cortés-Martínez, F. and Giernacki, W. Real-time Implementation and Flight Tests using Linear and Nonlinear Controllers for a Fixed-wing Miniature Aerial Vehicle (MAV), *International Journal of Control, Automation and Systems*, 2018, **16**, pp 392396.

32. Martins, G, Moses, A., Rutherford, M.J. and Valavanis, K.P. Enabling intelligent unmanned vehicles through XMOS Technology, *The Journal of Defense Modeling and Simulation*, 2012, **9**, (1), pp 71-82.

33. FlightGear FlightGear flight simulator, *www.flightgear.org*, June 2020.

34. Etkin, B. and Reid, L.D. Dynamics of Flight: Stability and Control, 1996, Wiley.

35. Guglieri, G. Effect of autopilot modes on flight performances of electric mini-UAVs, *The Aeronautical Journal*, 2013, **117**, (1187), pp 5769.

36. Guglieri, G. and Sartori, D. Experimental Characterization of Actuators for Micro Air Vehicles, *International Journal of Micro Air Vehicles*, 2011, **3**, (2), pp 49-59.

37. Marguerettaz, P., Sartori, D., Guglieri, G. and Quagliotti, F.B. Design and Development of a Man Portable Unmanned Aerial System for Alpine Surveillance Missions, *UAS International 2010*, 2010, pp 1-8.

38. Capello, E., Guglieri, G., Marguerettaz, P. and Quagliotti, F.B. Preliminary Assessment of Flying and Handling Qualities for Mini-UAVs, *Journal of Intelligent & Robotic Systems*, 2012, **65**, (1-4), pp 43-61.

39. CAPELLO, E., GUGLIERI, G., QUAGLIOTTI, F.B. and SARTORI, D. Design and Validation of an $\mathcal{L}_1$ Adaptive Controller for Mini-UAV Autopilot, *Journal of Intelligent & Robotic Systems*, 2013, **69**, (1-4), pp 109-118.

40. CAPELLO, E., SARTORI, D.GUGLIERI, G. and QUAGLIOTTI, F.B. Robust assessment for the design of multi-loop proportional integrative derivative autopilot, *IET Control Theory & Applications*, 2012, **6**, (11), pp 1610-1619.

41. PARK, H., KIM, M-H., CHANG, C-H., KIM, K., KIM, J-G. and KIM, D-H. Design and Experimental Validation of UAV Control System Software Based on the TMO Structuring Scheme, *5th IFIP WG 10.2 International Workshop on Software Technologies for Embedded and Ubiquitous Systems*, 2007, **4761**, pp 192-201.

42. SORTON, E. and HAMMAKER, S. Simulated Flight Testing of an Autonomous Unmanned Aerial Vehicle Using FlightGear, *Infotech at Aerospace 2005 Conference and Exhibit*, 2005, pp 1-13.

43. JUNG, D. and TSIOTRAS, P. Modeling and Hardware-in-the-Loop Simulation for a Small Unmanned Aerial Vehicle, *Infotech at Aerospace 2007 Conference and Exhibit*, 2007, pp 1-13.

44. XMOS XMOS technology, *www.xmos.com*, May 2020.

45. SARTORI, D., QUAGLIOTTI, F.B., RUTHERFORD, M.J. and VALAVANIS, K.P., Implementation and Testing of a Backstepping Controller Autopilot for Fixed-wing UAVs, *Journal of Intelligent & Robotic Systems*, 2014, **76**, (3), pp 505-525.