pp 2087–2113. © The Author(s), 2021. Published by Cambridge University Press on behalf of Royal Aeronautical Society.
doi:10.1017/aer.2021.53

Design and development of a backstepping controller autopilot for fixed-wing UAVs

D. Sartori
dan.sartori@sjtu.edu.cn
Shanghai Jiao Tong University
Institute of Sensing and Navigation
Shanghai
China

F. Quagliotti
Politecnico di Torino
Department of Mechanical and Aerospace Engineering
Torino
Italy

M.J. Rutherford
University of Denver
Department of Computer Science
Denver
USA

K.P. Valavanis
University of Denver
Department of Electrical & Computer Engineering
Denver
USA

ABSTRACT

Backstepping represents a promising control law for fixed-wing Unmanned Aerial Vehicles (UAVs). Its non-linearity and its adaptation capabilities guarantee adequate control performance 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

Received 22 September 2020; revised 2 April 2021; accepted 24 May 2021.

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, while slower outer loop navigation variables are Proportional–Integral–Derivative (PID) controlled. The controller is evaluated through numerical simulations for two very diverse fixed-wing aircraft performing complex manoeuvres. The controller behaviour 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.

Keywords: Backstepping; Controller; Control systems; UAV; Fixed-wing; Aircraft; 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 non-linear 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
$\hat{p}, \hat{q}, \hat{r}$	non-dimensional 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
$\mathrm{X}_B, \mathrm{Y}_B, \mathrm{Z}_B$	body axes
$\mathrm{X}_N, \mathrm{Y}_N, \mathrm{Z}_N$	NED axes
$\mathrm{X}_S, \mathrm{Y}_S, \mathrm{Z}_S$	stability axes
$\mathrm{X}_W, \mathrm{Y}_W, \mathrm{Z}_W$	wind axes
$x_1, x_2, \qquad x_2$	global change of variable states
$\boldsymbol{x}$	general state vector ∈ R^n
y	general change of variable state
y_α, y_β	change of variable states for f_α and f_β

Greek symbols

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 UAVs is characterised by highly non-linear behaviour. A severe coupling exists between the longitudinal and lateral–directional dynamics, and the sensitivity to external disturbances is considerable. 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.

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

In a recent publication(4), the authors illustrated 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, while the designer has adequate control over the system response and a clear understanding of the control action. The PID gains can be tuned using many non-heuristic methods, as explained in Refs (6) and(7), and currently such research is focusing on the auto-tuning process; see for example Ref. (8). One drawback of the standard PID approach is its inability to cope with the full flight envelope, thus gain scheduling is commonly adopted as the performance of a PID controller decreases when moving away from the design point. Another disadvantage is that traditional PIDs do not guarantee sufficient robustness to the extent of model parametric uncertainties expected to be found in small fixed-wing unmanned aircraft.

The solutions proposed in Ref. (4) include non-linear, adaptive or robust control laws that theoretically can guarantee satisfactory performance over a large flight envelope, also in presence of uncertainties; For instance, the authors of Ref. (9) propose a non-linear model predictive control for fixed-wing UAV path tracking, Ref. (10) investigates the feasibility of H_2 and H_∞ autopilots for longitudinal UAV control and Ref. (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 combine high-level intelligent tasks smoothly with low-level input/output routines are the main obstacles. The miniaturisation 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 available for fixed-wing aircraft remains relatively small. An example is Ref. (12), which further extends the work of Ref. (9) and solves the need to compromise between smooth convergence and computational performance originally highlighted by the authors. Similarly, in Ref. (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 non-linear backstepping is presented in this paper. The objective of the paper is twofold: the adaptation of an existing backstepping technique(2) with the aim of generating a comprehensive control configuration suitable for mini-UAV autopilots, and its real-time implementation on a microcontroller board. In fact, in contrast to many related studies, the implementation simplicity of the proposed approach allows the advantages of the backstepping controller to be exploited on a fixed-wing UAV. This was demonstrated in a related work by the same authors(39), 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 Linear–Quadratic Regulator (LQR) or feedback linearisation, backstepping is a non-linear control law that guarantees satisfactory performance over the whole flight envelope(1). Useful non-linearities are maintained, and additional non-linear damping terms can be introduced to increase robustness to model errors or improve transient performance(14). Furthermore, as backstepping belongs to the Lyapunov family, it has guaranteed convergence of the tracking error and asymptotic stability(15).

The remainder of this paper is organised 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 can be expressed in a suitable form for the controller design,

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

which is later introduced in Section 4. Section 5 describes the results of the software simulations, while Section 6 presents the implementation and the results of the hardware-in-the-loop (HIL) simulations. Finally, Section 7 concludes the paper and describes future work towards achieving comprehensive flight tests.

2.0 RELATED WORK

A variety of projects have illustrated 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, Ref. (16) presents an adaptive backstepping control law for angle-of-attack tracking, Ref. (18) uses adaptive backstepping for UAV velocity and flight path angle control and Ref. (19) combines the $\mathcal{L}_1$ adaptive methodology with backstepping for longitudinal control of a multi-axis thrust vectoring fighter aircraft. The path-following problem is addressed in Ref. (20), where the roll angle command is generated through backstepping with the parameter adaptation technique, and hardware-in-the-loop (HIL) simulations validate the results. Fault-dependent control allocation using $\mathcal{L}_1$ adaptive backstepping for longitudinal UAV control is analysed in Ref. (21).

The number of works describing combined longitudinal and lateral–directional aircraft control is limited. In Ref. (22), outer-loop variables such as incidence, sideslip angle and roll angle are controlled by neural network adaptive backstepping through body-axes angular rates. In Ref. (23), 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 Ref. (17) present different backstepping solutions, combined with sliding mode control, for 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, viz. the angle-of-attack, sideslip angle and stability-axes roll rate, are controlled via the backstepping approach described in Refs (2) and (3). This method is designed for general aircraft manoeuvring within the whole flight envelope. Its ability to deal with the high angles of attack and sharp turns typical of small highly manoeuvrable UAVs is demonstrated. Non-linear natural-stabilising aerodynamic loads are included and employed by the controller. This approach differs from feedback linearisation, where these forces are first modelled and then cancelled, allowing for a less accurate knowledge of the aircraft dynamics. More slowly changing outer-loop navigation variables, viz. 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 practical employment of backstepping control on microcontrollers for small UAVs. Adaptation and a more advanced outer-loop design lies beyond the scope of this paper.

A constant in most of the backstepping approaches summarised above is its combination with complex adaptation laws. The benefits of non-linear control supported by advanced adaptation are clear, but the problems of real-time implementation are considerable. To the best of the authors' knowledge, among the adaptive backstepping solutions described above, only the

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

work of Ref. (20) has been implemented on a microcontroller suitable for small fixed-wing UAVs. The algorithm described in Ref. (24), 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 Ref. (25), where basic roll and pitch angles hold is achieved through implementation on a Procerus Kestrel autopilot. The work of Ref. (17) has only been tested in open loop to acquire and validate the magnitude of the control signals. In Ref. (26), some of the same authors illustrated flight test results from a backstepping controlling altitude, roll angle and yaw angle, but with very limited implementation and testing details.

In the present paper, an innovative use of microprocessor technology based on cutting-edge transistor computers is employed to support the controller implementation[27]. 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 satisfactory flight performance. Furthermore, we demonstrate the applicability of our backstepping solution for two fixed-wing aircraft having very different configurations 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 aircraft and its orientation with respect to a reference system[29]. The force equation in generic Centre of Gravity (CG)-centred body axes (X_B, Y_B, Z_B) (Fig. 1) is expressed as

$$m\dot{\mathbf{V}} = \mathbf{F} - \boldsymbol{\omega} \times m\mathbf{V}, \qquad \ldots(1)$$

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 centre of mass: aerodynamics forces, engine thrust and gravity force. The moment equation has a similar structure:

$$I\dot{\boldsymbol{\omega}} = \mathbf{M} - \boldsymbol{\omega} \times I\boldsymbol{\omega}, \qquad \ldots(2)$$

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

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

The vector $\boldsymbol{\Phi} = (\phi, \theta, \psi)^T$ 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.

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

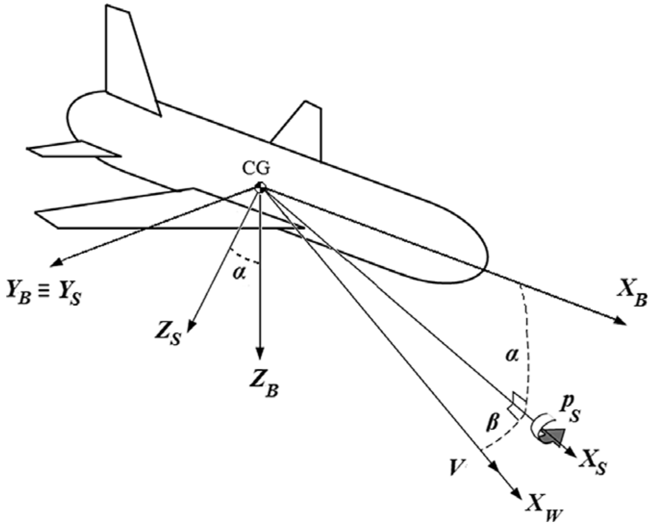


Figure 1. Controlled variables and reference axes at initial time.

The recursive nature of the backstepping controller requires that the equations governing the system may assume a general triangular structure called pure-feedback form(14):

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

In Equation (4), $\mathbf{x} \in \mathrm{R}^n$ is the state vector and $\xi_{1,...,k}$ are scalars denoting other states of the system. The functions f_i $(i = 1, ..., k)$ are non-linear and depend only on $\mathbf{x}$ and 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)–(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 (Fig. 1). The aim is to design a controller such 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 behaviour and the flight direction. A null sideslip angle is desired in cruise flight to achieve symmetric flight and reduce aerodynamic drag. The ability of an aircraft to cancel out sideslip angle perturbations is a sign of its lateral–directional static stability.

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

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

of the wing aerofoil and Y_S completes the right-handed reference frame (Fig. 1). X_S and X_B are separated by the angle-of-attack, and a single rotation of magnitude α about $\mathrm{Y}_S \equiv \mathrm{Y}_B$ 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

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

The dynamics considered for the control design are obtained from the force equation written in wind axes; the complete derivation is available in Ref. (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 centre-of-mass trajectory and Z_W lies in the plane of symmetry of the aircraft, directed from the upper to the lower wing aerofoil surface. Given this background, the following assumptions are proposed:

- **Assumption 1:** The deflection of the control surfaces only generates a variation in moments, while 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 the traditional configuration, where control surfaces are far from the aircraft centre of gravity[29]. 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 the 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 centre of gravity. Furthermore, the addition of a reference lever-arm distance in the moment mathematical formulation (see for instance Equation (19)) increases the moment contribution with respect to the force contribution. Once the trim condition is achieved, the control deflections for manoeuvre are minimal, reducing to a negligible value the variation of forces so produced. Assumption 2 is considered valid in steady flight or during smooth manoeuvres. In fact, the disregarded aerodynamic derivatives $C_{L\dot{\alpha}}$ and $C_{Y\dot{\beta}}$ originate from 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 manoeuvres; 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 to the angular velocities and α and β themselves are then

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

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

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

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

with $g = 9.81\text{m/s}^2$ gravity acceleration. Equation (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 Equation (5) the stability-axes angular rates are introduced in the $\dot{\alpha}$ and $\dot{\beta}$ dynamics. The result is

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

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 is described by the relationship $\dot{\boldsymbol{\omega}}_s = \mathbf{u}_c$. Combining this formulation with Equation (8) gives

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

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 to be independent of 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

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 with 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 manoeuvres, 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.

Equation (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 follows:

$$\dot{p}_s = u_1 \qquad \ldots (10)$$

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

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

three sub-controllers stabilising 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 it is not physically realistic and not necessary, as shown below. 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 manoeuvres where strong coupling exists between longitudinal and lateral–directional planes.

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

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

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

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

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

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

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

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	$q_s + f_\alpha(\alpha^{ref}, y_\alpha)$	$-r_s + f_\beta(0, y_\beta)$
$\Omega(x_1)$	$f_\alpha(\alpha, y_\alpha) - f_\alpha(\alpha^{ref}, y_\alpha)$	$f_\beta(\beta, y_\beta) - f_\beta(0, y_\beta)$

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

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

As shown in Ref. (2), a linear globally stabilising control law for the system of Equation (14) is

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

with $k_2 > 2k_1 > \max\{0, k_u\}$. Using the relationships in Table 1, the control laws for Equations (11) and (12) become

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

with

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

Finally, the proportional control adopted for p_s is

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

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

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 thus

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

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 stabilises three variables related to the aircraft attitude. A global autopilot configuration capable of controlling speed V, altitude h and heading ψ is required. In a real-life implementation, these variables could be easily measured with, respectively, a pitot tube, a barometric pressure sensor and a 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, characterising attitude, from the slower dynamics, characterising navigation. A prompt response of the backstepping controller is necessary when dealing with attitude variables, which are of prime importance for aircraft safety. For instance, immediate control of α in presence of vertical gusts could prevent stall or dangerous flight regimes. Consistent with assumption 3, the slower navigation variables can be successfully handled using traditional PIDs whose gains are tuned manually using an heuristic approach. The goal is to optimise 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 to avoid the request for motion that is incompatible with the aircraft dynamics during sudden manoeuvres. 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, while the conversion to stability axes p_s^{ref} is performed using Equation (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 but not in forces. The deflection vector $\boldsymbol{\delta} = (\delta_e, \delta_a, \delta_r)^T$ is obtained by substituting into Equation (18) the most general expressions for the moments,

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

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

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

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

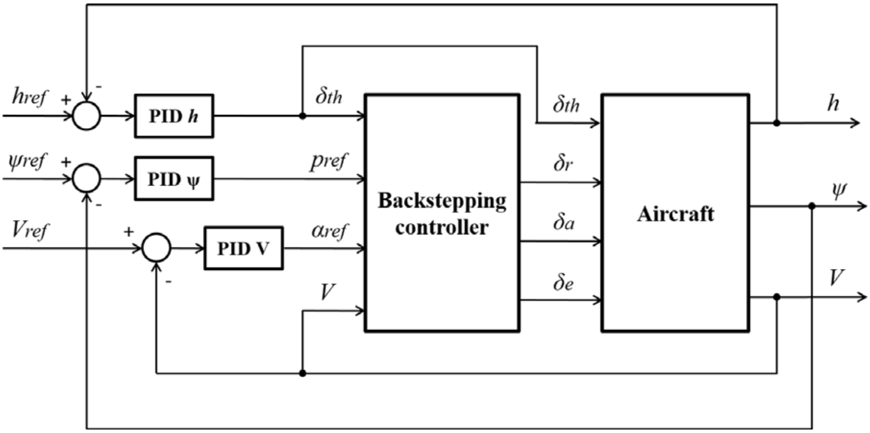


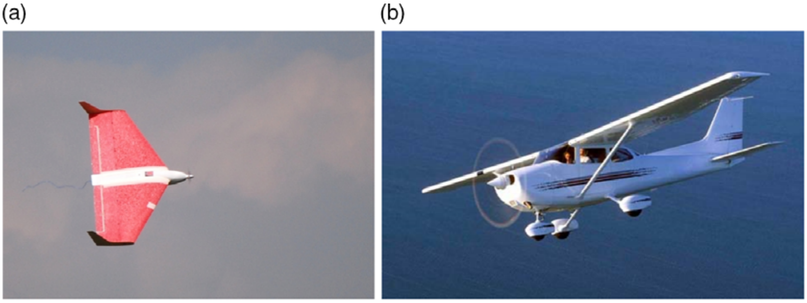
Figure 2. Backstepping control strategy for fixed-wing aircraft.

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 simpler: δ_e is found from the $M(\delta_e)$ equation, while δ_a and δ_r are found by solving the linear system with $L(\delta_a, \delta_r)$ and $N(\delta_a, \delta_r)$. The engine thrust vector is considered to be aligned with the aircraft X_B axis and so 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 the airspeed and the throttle controls the altitude, is a standard autopilot mode. As explained in Ref. (30), this approach guarantees better tracking of the airspeed, which is a key flight safety parameter. Table 2 summarises the controlled variables, their commands and the control method.

The proposed control scheme is shown in Fig. 2. The computed control inputs act on the aircraft, while its measured controlled states, viz. total speed, altitude and heading angle, are the feedback variables. Their differences from 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 inputs to the backstepping controller as required by the control law definition and for the estimation of the inner-loop states. Note that, in fact, the variables α, β and p_s, used for the definition of the inner-loop error, are estimated with a good degree of accuracy inside the backstepping controller by integrating Equations (10)–(12), as shown below in

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press



MH850 UAV

Cessna 172P

Figure 3. Aircraft employed for the numerical simulations.

Fig. 4(b). A support for the accurate estimation of α and β can be provided by the feedback of ϕ and θ, easily measurable with an Inertial Measurement Unit (IMU). These values appear in Equation (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 greatly 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 the FlightGear simulator(28) and demonstrates the ability of the controller to work in real time at slow sampling rates.

5.1 Simulink simulations

Simulink simulations are performed by integrating the equations through a second-order Heun method with 0.01s time step; continuous time blocks are employed. The block scheme follows the structure of Fig. 2. The non-linear equations of motion of Ref. (29) are adopted in the aircraft block. Here, actuator transfer functions as from Ref. (32) 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 non-linear models representing the MH850 UAV and the Cessna 172P aircraft (Fig. 3). The MH850 is characterised by a tailless configuration, electric propulsion and non-movable vertical fins at the wingtips(32). The wingspan is 85cm, the mass 1kg and the cruise speed 15m/s. Aerodynamic control is achieved with elevons, which control the 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 non-linear aircraft

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

Table 3
Aircraft specific properties

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

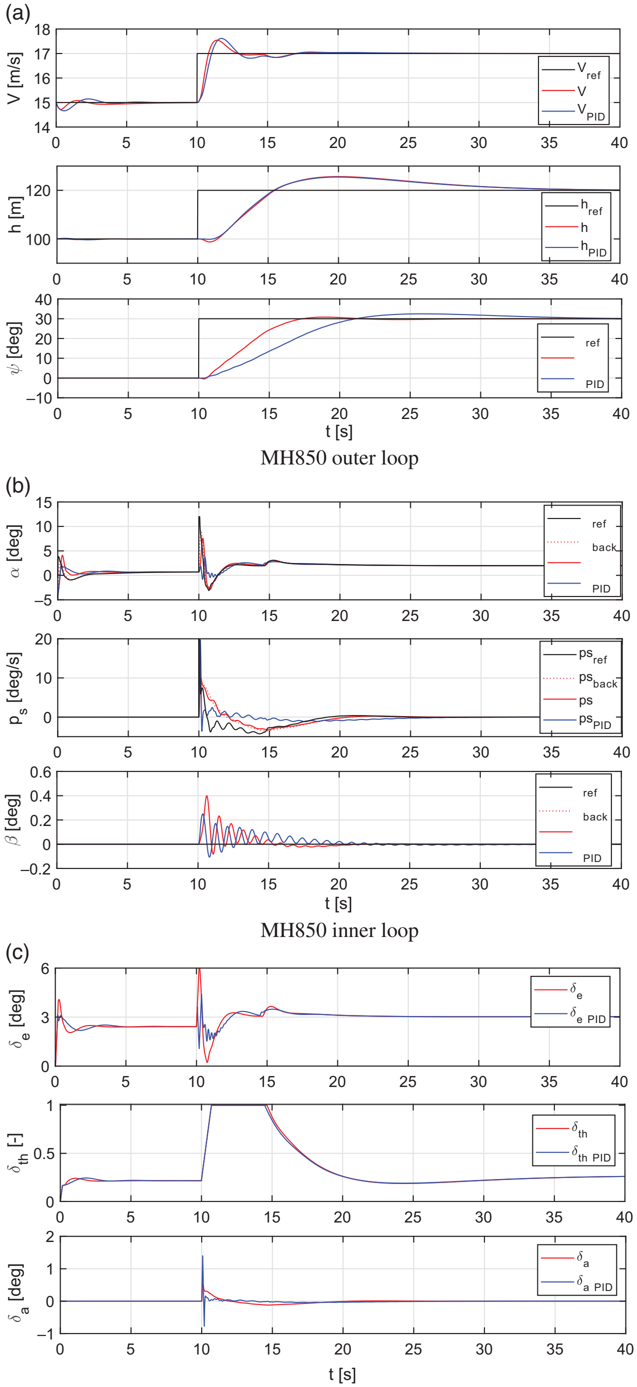
model[33,34]. The Cessna 172P is a single-combustion-engine aircraft with a standard configuration including high wing and fixed, tricycle landing gear. Take-off weight is around 880kg, and wingspan 11m. The aircraft is powered by a Lycoming O-320-D2J engine able to produce 160hp and guaranteeing a cruise speed of 60m/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 (Table 3). Testing the controller on a lower-performance platform allows to prove its universality and identify its limits.

It is interesting to explain how the calculation of the commands for the MH850 rudderless configuration is performed. As pointed out above, the δ_e command is found from the $M(\delta_e)$ equation as $C_{m\delta_a} = 0$. Both L and N moments are functions of the remaining δ_a command, which 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 presented in Fig. 4(a) 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 satisfactory with just aileron control. Figure 4(b) and (c) 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. Accurate velocity tracking is achieved thanks to 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. The 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°, barely noticeable in flight. The deflection of the elevons always remains within the 20° maximum value, and 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 being obtained from the outer PID on ψ. A similar configuration, commonly employed in commercial autopilots, was illustrated in Ref. (35). In the outer loop, the PID performance is almost comparable to backstepping, the V response being slightly more

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press



MH850 outer loop

MH850 inner loop

MH850 commands

Figure 4. Simulink responses for MH850.

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

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

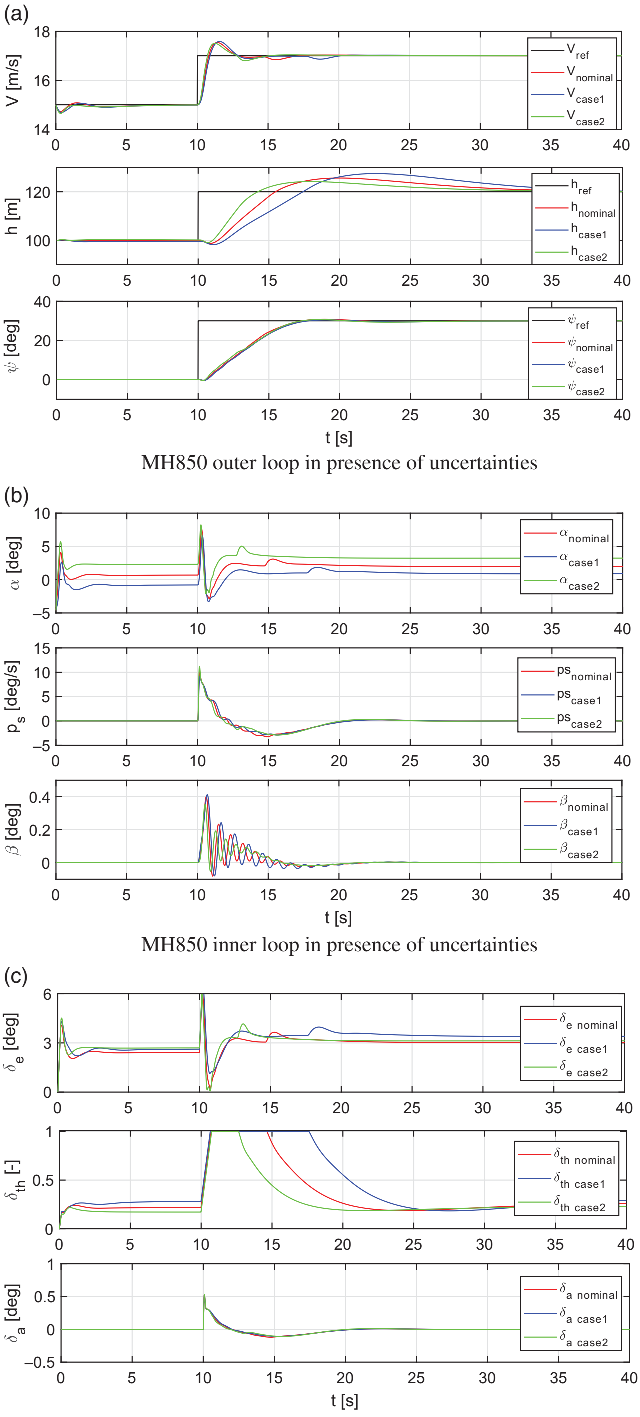
oscillatory while the ψ response is slower and has larger overshoot. Similarly, no significant difference is observable in the inner loop. Instead, the commands δ_e and δ_a from the PID control show a higher oscillatory behaviour, while the altitude and throttle remain basically unchanged.

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 non-linear aircraft model, while the controller remains unchanged. Case 1 considers a heavier aircraft, with higher inertia and 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 and has lower inertia, and its centre 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. Figure 5 shows the obtained results in comparison with the nominal case. In the outer-loop response of Fig. 5(a), 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, and higher overshoot and settling time. In the inner loop, see Fig. 5(b), 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 plot of commands in Fig. 5(c) confirms that full throttle command is required for 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 minimise their values.

The PID controller introduced above is tested for the same perturbed aircraft configurations, and the results are presented in Fig. 6. It is evident that the nominal PID controller is not able to withstand the uncertainties introduced in both cases. The aircraft loses directional control and accelerates while quickly losing altitude, crashing into the ground in less than 10s. 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 is proved to be robust as it guarantees satisfactory performance in all cases.

To demonstrate the ability of the backstepping controller to withstand sensor noise, the same simulation is performed but 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 a magnetometer with $\sigma_\psi = 1$deg. A Kalman filter

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press



MH850 outer loop in presence of uncertainties

MH850 inner loop in presence of uncertainties

MH850 commands in presence of uncertainties

Figure 5. Simulink responses for MH850 in presence of uncertainties.

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

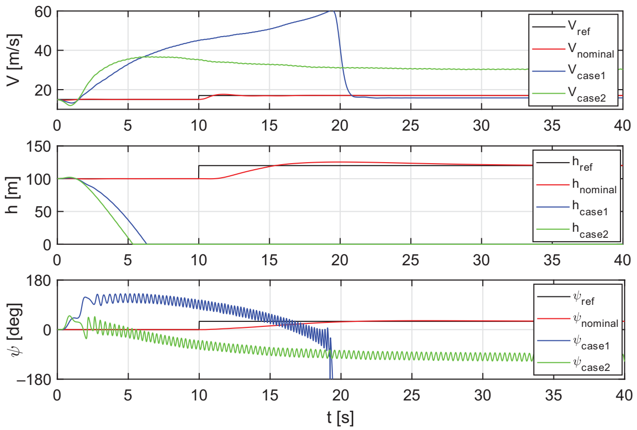
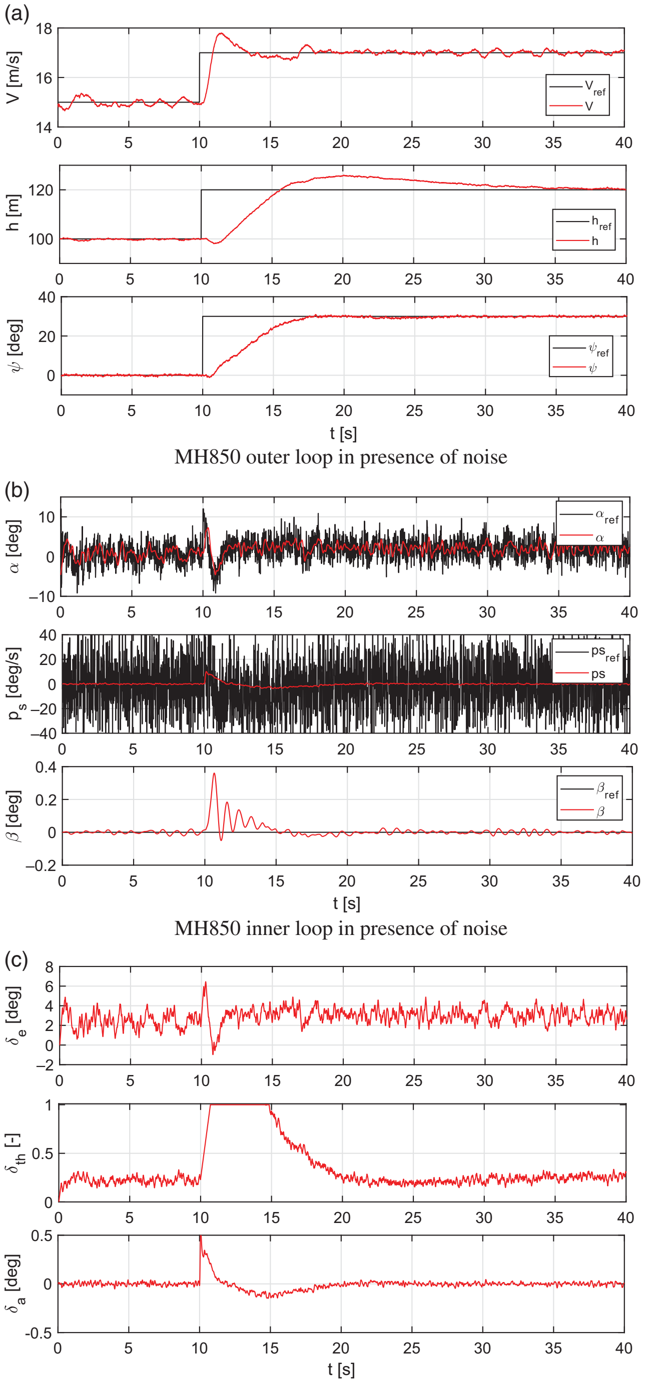


Figure 6. Simulink responses for MH850 with PID controller in presence of uncertainties.

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 satisfactory response. In fact, when steady state is achieved, the standard deviation is just 0.074m/s. The inner loop is more affected by noise because of the derivative operation 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 with 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 satisfactory reference tracking, albeit with a more disturbed response. The calculated commands shown in Fig. 7(c), in particular for the elevator, are also affected by noise but remain compatible with the dynamic response of the actuators.

The C172P responses to ramp inputs are shown in Fig. 8(a). 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 aircrafts: 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 behaviour of the inner-loop variables, as shown in Fig. 8(b). 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 thus 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 in Fig. 8(c).

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press



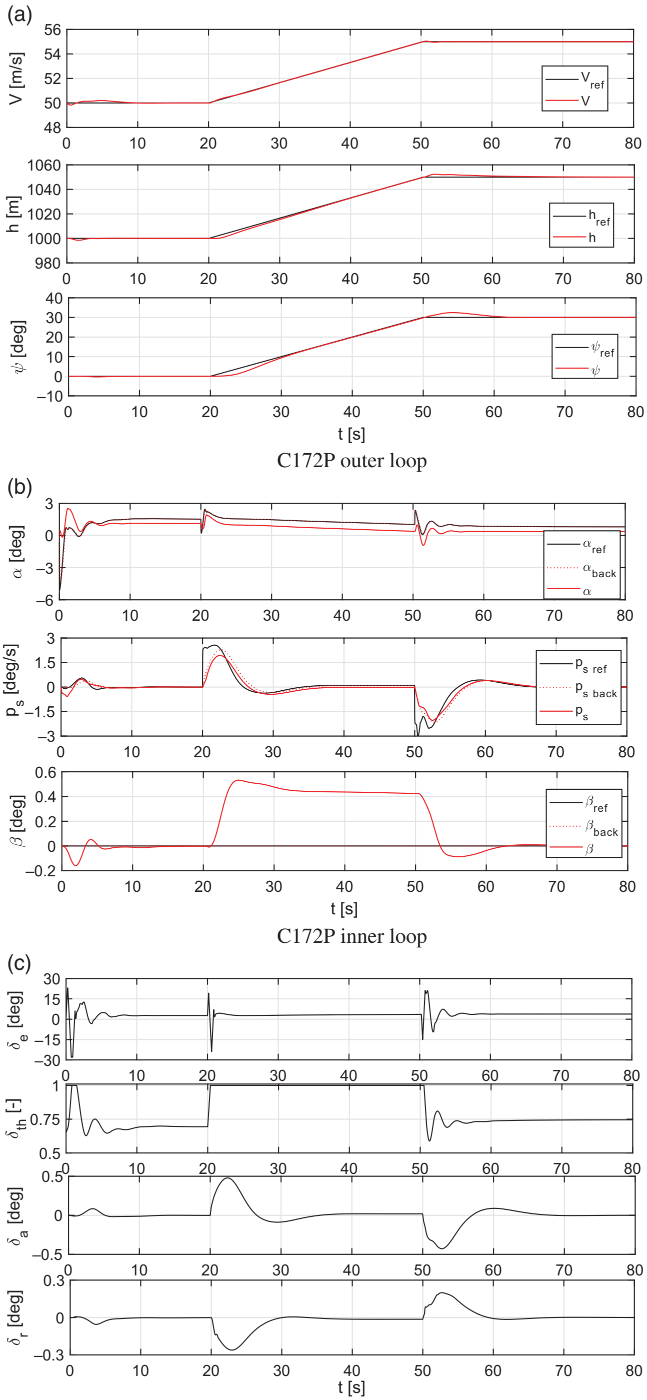
MH850 outer loop in presence of noise

MH850 inner loop in presence of noise

MH850 commands in presence of noise

Figure 7. Simulink responses for MH850 in presence of noise.

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press



C172P outer loop

C172P inner loop

C172P commands

Figure 8. Simulink responses for C172P.

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

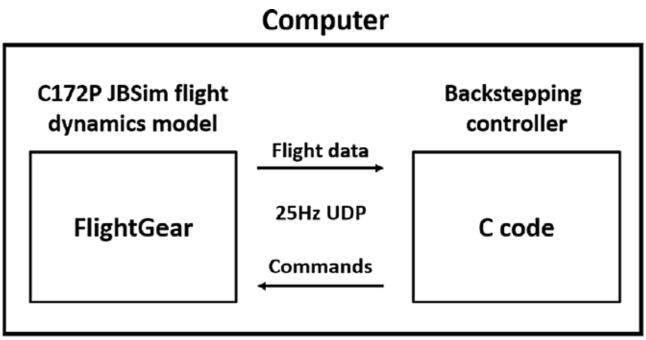


Figure 9. SIL layout.

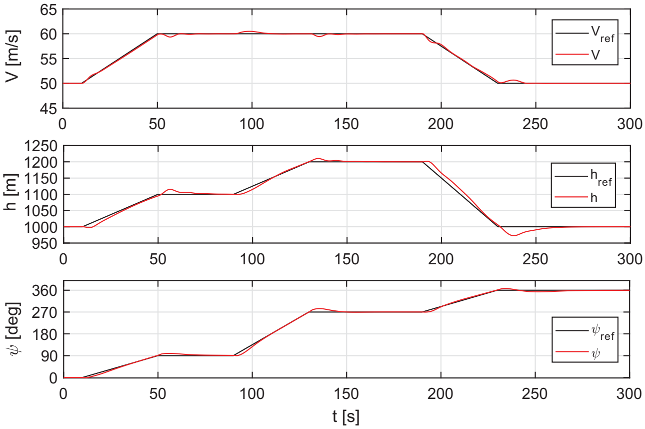


Figure 10. FlightGear SIL simulated manoeuvre for C172P.

5.2 Software-in-the-loop simulations

As a first step towards achieving 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 the same features as described in Section 5.1. The adopted architecture is presented in Fig. 9.

FlightGear is a freeware open-source flight simulator developed by volunteers around the world that offers academic developers a well-used tool to test their aircraft models and control laws; see for instance Refs (36), (37). 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 that characterise aircraft motion. Data transfer between the C application and FlightGear is performed via User Datagram Protocol (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 25Hz 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 100Hz.

Figure 10 shows the results of a complex manoeuvre. The aircraft is requested first to climb and turn while accelerating, then to maintain the speed while climbing and turning more

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

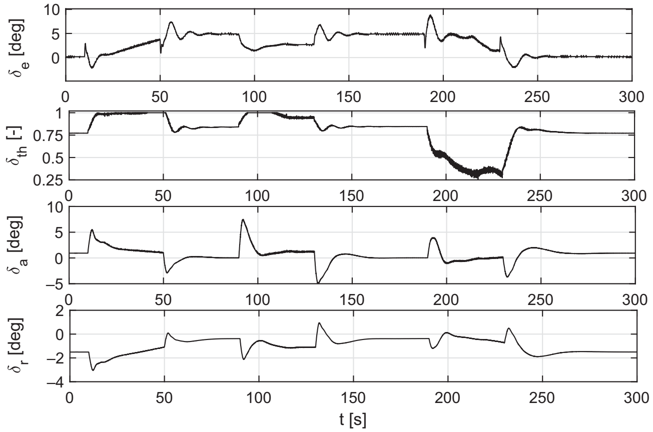


Figure 11. FlightGear SIL simulated manoeuvre commands for C172P.

aggressively, finally to decelerate while performing another turn and rapidly losing altitude. All variables are tracked with good accuracy in every phase of the manoeuvre. Speed control performs the best, where 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 penalised 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, viz. 20° for elevator and aileron or 16° for rudder, while a rate limiter is imposed. The motor instead goes full throttle during the climbing phases.

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[38]. This board is characterised 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[27]. An advantage of using the XMOS technology is the ease of programming the board. The coding language is called XC, and its strong similarity with C allows a fast and simple implementation. The low cost of the board and its limited weight (19g) and dimensions (50 × 50mm) make it suitable for small UAV applications.

The hardware-in-the-loop (HIL) setup and scheme are presented in Figs 12 and 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 a serial connection 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 25Hz. For serial communication, a baud rate of 153,600Bd is chosen to maximise the data transmission speed and avoid overlap between send and receive

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

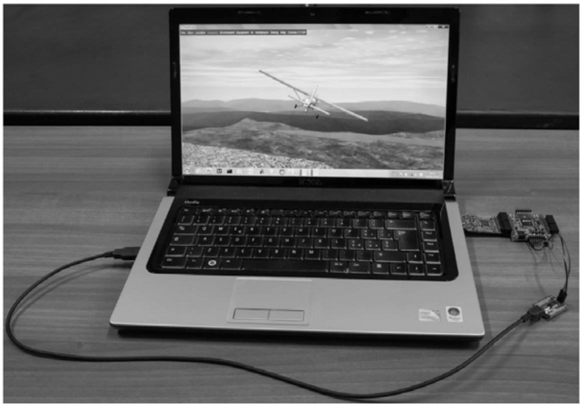


Figure 12. HIL setup.

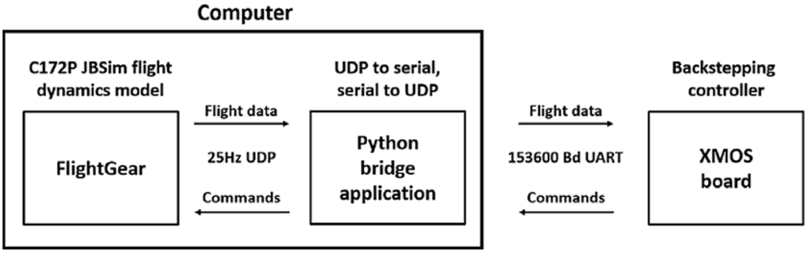


Figure 13. HIL layout.

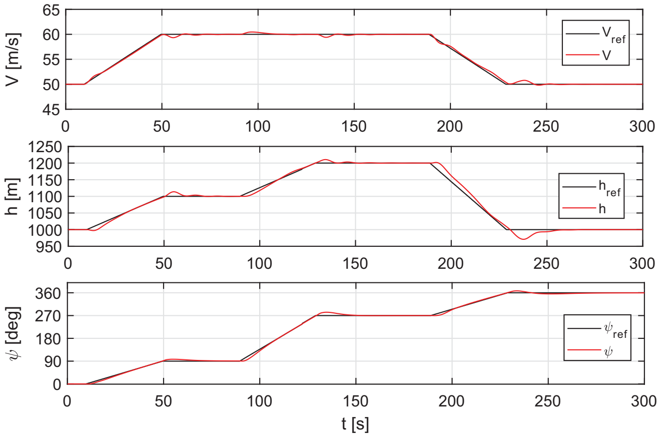


Figure 14. FlightGear HIL simulated manoeuvre for C172P.

tasks. The controller computation time is slightly less than 0.004s on 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 with the backstepping and PID gains unchanged. Figure 14 validates the real-time

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

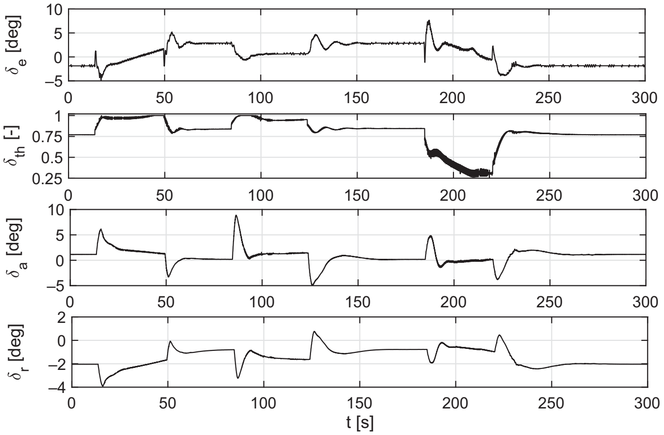
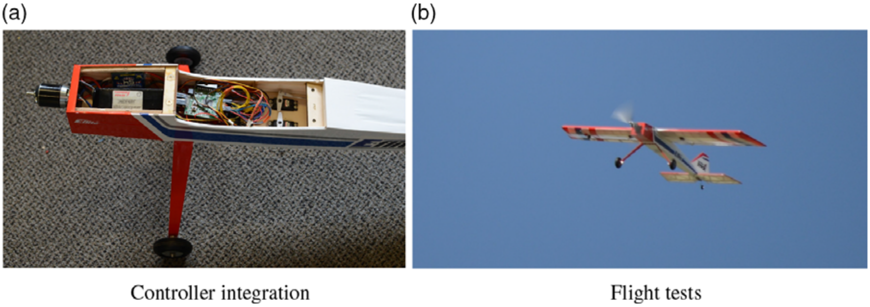


Figure 15. FlightGear HIL simulated manoeuvre commands for C172P.



Controller integration

Flight tests

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

implementation; the tracking achieved with the microcontroller is accurate and virtually identical to that obtained in the SIL case. The commands for this simulation are represented in Fig. 15, showing an excellent matching with the commands from the SIL.

The controller has been integrated onto an Ultra Stick 25e RC aircraft model (Fig. 16), and preliminary validation flight tests have been conducted(39). The upgrade of the controller is under consideration. The initial control strategy was deliberately kept simple to demonstrate the feasibility of the real-time implementation and perform flight tests. Adaptive backstepping and substitution of the PIDs with more advanced laws are the changes investigated.

7.0 CONCLUSIONS

An autopilot configuration combining non-linear control with the traditional PID technique is presented. The backstepping controller is employed to stabilise fast inner-loop variables characterising the aircraft attitude and aerodynamic angles, while PID gains control more slowly changing navigation variables. The backstepping method is chosen for its ability to deal with the non-linearities that characterise small fixed-wing UAV dynamics. This method requires a fairly rich knowledge of the aircraft characteristics but in return ensures good performance over a large flight envelope.

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

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 manoeuvres, it is demonstrated that, aircraft allowing, aggressive flight is achievable. Complex manoeuvres characterised by severe coupling are performed with limited tracking error. The simple solution adopted differs from the standard adaptive backstepping approaches popular in literature but 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, revealing satisfactory performance on complex manoeuvres. The control strategy described herein is believed to be implementable on any microcontroller board for small UAV applications.

SUPPLEMENTARY MATERIAL

To view supplementary material for this article, please visit https://doi.org/10.1017/aer.2021.53

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 Proc. Vol.*, 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, September 2020, www.pixhawk.org
6. COMINOS, P. and MUNRO, N. PID controllers: Recent tuning methods and design to specification, *IEEE Proc. Cont. Theory Appl.*, 2002, **149**, (1), pp 46–53.
7. ASTRÖM, K.J., and HÄGGLUND, T. *PID Controllers: Theory, Design, and Tuning*, Instrument Society of America, 1995.
8. POKSAWAT, P., WANG, L. and MOHAMED, A., Gain scheduled attitude control of fixed-wing UAV with automatic controller tuning, *IEEE Trans. Cont. Syst. Technol.*, 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 Trans. Cont. Syst. Technol.*, 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 Cont. Theory Appl.*, 2008, **2**, (7), pp 583–594.
11. CASTAÑEDA, H., SALAS-PEÑA, O.S. and DE LEÒN-MORALES, J. Extended observer based on adaptive second order sliding mode control for a fixed wing UAV, *ISA Trans.*, 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, 22^{nd} *Mediterranean Conference on Control and Automation*, 2014, pp 1159–1165.
14. KRSTIĆ, M., KANELLAKOPOULOS, I. and KOKOTOVIĆ, P. *Nonlinear and Adaptive Control Design*, Wiley, 1995.
15. KIM, K.-S and KIM, Y. Robust backstepping control for slew maneuver using nonlinear tracking function, *IEEE Trans. Cont. Syst. Technol.*, 2003, **11**, (6), pp 822–829.

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press

16. JU, H. and TSAI, C. Longitudinal axis flight control law design by adaptive backstepping, *IEEE Trans. Aerosp. Electron. Syst.*, 2007, 43, (1), pp 311–329.
17. 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.
18. GAVILAN, F., ACOSTA, J. and VAZQUEZ, R. Control of the longitudinal flight dynamics of an UAV using adaptive backstepping, *18th IFAC World Cong.*, 2011, **44**, (1), pp 1892–1897.
19. 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.
20. JUNG, D. and TSIOTRAS, P. Bank-to-turn control for a small UAV using backstepping and parameter adaptation, *17th IFAC World Cong.*, 2008, **41**, (2), pp 4406–4411.
21. NØRGAARD, M.E., , HANSEN, S., BREIVIK, M. and BLANKE, M. Performance comparison of controllers with fault-dependent control allocation for UAV, *J. Intell. Robot. Syst.*, 2017, **87**, pp 187–207.
22. LEE, T. and KIM, Y. Nonlinear adaptive flight control using backstepping and neural networks controller, *J. Guid. Cont. Dyn.*, 2001, **24**, (4), pp 675–682.
23. SONNEVELDT, L., CHU, Q.P. and MULDER, J.A. Nonlinear flight control design using constrained adaptive backstepping, *J. Guid. Cont. Dyn.*, 2007, **30**, (2), pp 322–336.
24. BREZOESCU, A., ESPINOZA, T., CASTILLO, P. and LOZANO, R. Adaptive trajectory following for a fixed-wing UAV in presence of crosswind, *J. Intell. Robot. Syst.*, 2013, **69**, (1–4), pp 257–271.
25. 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
26. ESPINOZA-FRAIR, 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), *Int. J. Cont. Automat. Syst.*, 2018, **16**, pp 392–396.
27. MARTINS, G, MOSES, A., RUTHERFORD, M.J. and VALAVANIS, K.P. Enabling intelligent unmanned vehicles through XMOS technology, *J. Defense Model. Simul.*, 2012, **9**, (1), pp 71–82.
28. FlightGear FlightGear flight simulator, June 2020, www.flightgear.org
29. ETKIN, B. and REID, L.D. *Dynamics of Flight: Stability and Control*, Wiley, 1996.
30. GUGLIERI, G. Effect of autopilot modes on flight performances of electric mini-UAVs, *Aeronaut. J.*, 2013, **117**, (1187), pp 57–69.
31. GUGLIERI, G. and SARTORI, D. Experimental characterization of actuators for micro air vehicles, *Int. J. Micro Air Veh.*, 2011, **3**, (2), pp 49–59.
32. 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.
33. CAPELLO, E., GUGLIERI, G., MARGUERETTAZ, P. and QUAGLIOTTI, F.B. Preliminary assessment of flying and handling qualities for mini-UAVs, *J. Intell. Robot. Syst.*, 2012, **65**, (1–4), pp 43–61.
34. CAPELLO, E., GUGLIERI, G., QUAGLIOTTI, F.B. and SARTORI, D. Design and validation of an $\mathcal{L}_1$ adaptive controller for mini-UAV autopilot, *J. Intell. Robot. Syst.*, 2013, **69**, (1-4), pp 109–118.
35. CAPELLO, E., Sartori, D.Guglieri, G. and Quagliotti, F.B. Robust assessment for the design of multi-loop proportional integrative derivative autopilot, *IET Cont. Theory Appl.*, 2012, **6**, (11), pp 1610–1619.
36. 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.
37. 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.
38. XMOS XMOS Technology, May 2020, www.xmos.com
39. SARTORI, D., QUAGLIOTTI, F.B., RUTHERFORD, M.J. and VALAVANIS, K.P., Implementation and testing of a backstepping controller autopilot for fixed-wing UAVs, *J. Intell. Robot. Syst.*, 2014, **76**, (3), pp 505–525.

https://doi.org/10.1017/aer.2021.53 Published online by Cambridge University Press