DENVER UNIVERSITY
DU SRI
UNMANNED SYSTEMS RESEARCH INSTITUTE

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

DU2SRI-2013-12-001

Daniele Sartori[1], Fulvia Quagliotti[2],
Matthew J. Rutherford[3], Kimon P. Valavanis[4]

Monday 23rd December, 2013

[1] PhD Student, Department of Mechanical and Aerospace Engineering, Corso Duca degli Abruzzi 24, 10129 Torino, Italy, daniele.sartori@polito.it

[2] Associate Professor, Department of Mechanical and Aerospace Engineering, Corso Duca degli Abruzzi 24, 10129 Torino, Italy, fulvia.quagliotti@polito.it

[3] Assistant Professor, Department of Computer Science, John Greene Hall, 2360 S. Gaylord St., Denver, CO 80208, USA, mjr@cs.du.edu

[4] Professor and Chair, Department of Electrical & Computer Engineering, Knudson Hall, 2390 S. York St., Denver, CO 80208, USA, kvalavan@du.edu

Abstract

Backstepping is one of the most promising advanced control laws developed for fixed-wing Unmanned Aerial Vehicles (UAVs). Its nonlinearity combined with adaptation guarantees adequate performance over the whole flight envelope even when the aircraft model is not exact. In the literature, there are several attempts to apply a backstepping controller to aspects of fixed-wing UAV flight. Few of them attempt a simultaneous longitudinal and latero-directional aircraft control, and the majority of these have not been implemented in a real-time controller. In this paper a backstepping approach able to control longitudinal and latero-directional motions is presented. Rapidly changing inner-loop variables are controlled with non-adaptive backstepping, less dynamic outer-loop variables are controlled with PID gains. The controller is evaluated through software-in-the-loop simulation in both continuous and discrete time domains, in the first case on two aircraft with different capabilities. The behavior with parametric uncertainties in the aircraft model or in presence of noise is also tested. The results of a real-time implementation on a microcontroller are presented and its performance is evaluated through hardware-in-the-loop simulation. Overall, the proposed backstepping controller has good performance on the aircraft evaluated for complex maneuvers involving control of multiple changing variables simultaneously.

Contents

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
$E(\Phi)$	rotation matrix
$\mathbf{F} = (F_x, F_y, F_z)^T$	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
$\mathbf{M} = (M, L, N)^T$	body-axes moment vector, N·m
m	aircraft mass, kg
$\hat{p}$, $\hat{q}$, $\hat{r}$	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
$\mathbf{u}_c = (u_1, u_2, u_3)^T$	control action vector, rad/s^2
u_s	change of variable control action

V_L	global control Lyapunov function
V, V^{ref}	measured and reference linear velocity, m/s
$\mathbf{V} = (u, v, w)^T$	body-axes linear velocity vector, m/s
$W(x_1)$	temporary control Lyapunov function
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	change of variable state
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

Chapter 1

Introduction

Small fixed-wing Unmanned Aerial Vehicle (UAV) flight dynamics are characterized by highly nonlinear behavior. A severe coupling exists between longitudinal and latero-directional dynamics and the sensitivity to external disturbances is considerable. The interest in the problem of 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. The wide range of possible missions, for instance traffic surveillance or firefighter support, is stimulating research and development of unmanned aerial systems of different size and configuration. A significant part of this research is dedicated to the design and development of adequate on-board controllers. Recent surveys by Chao, Cao and Chen [1] and Ollero and Merino [2] illustrate the current technology available for autopilot systems and describe the control laws commonly employed. The use of PID gains is still a popular approach in practice, in particular when dealing with commercial off-the-shelf autopilots such as the MicroPilot MP Series. 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 [3] and [4]. One drawback of the standard PID approach is the inability to deal with the flight envelope that might be required in most advanced mission profiles. As the performance of a PID controller decreases far from the design point, gain scheduling is a common approach to extend the validity of this technique to the whole flight envelope. Another disadvantage of PIDs is that they do not guarantee robustness to model parametric uncertainties which are a common occurrence in small fixed-wing UAVs.

Researchers are currently developing nonlinear, adaptive and robust control laws able to theoretically guarantee satisfying performance over a large flight envelope also in case of uncertainties. For instance the authors of [5] propose a nonlinear model predictive control for fixed-wing UAV path tracking, [6] investigates the feasibility of H_2 and H_∞ autopilots for longitudinal UAV control and [7] presents a combined adaptive control law based on shunting method and passification for an UAV autopilot homing guidance system. Nevertheless, the constraints imposed by real-time implementation often make these algorithms very challenging for the limited computational platforms available for small scale UAVs. As an example, the controller proposed in [5] is successfully implemented in a dedicated onboard computer installed on an experimental fixed-wing UAV and tested with real-time hardware-in-the-loop simulation. The authors, however, underline the need for a compromise between smooth convergence and computational performance in the determination of the receding horizon size. 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, see [8] and [9], is now enabling researchers to implement unmanned aircraft actually flown by self-developed control laws. Whereas several examples have been published for rotorcraft (an excellent survey is [10]) there are relatively few for fixed-wing aircraft. The diffusion of frameworks for control law development (e.g., [11] - [14]) has helped to reduce the barriers to successful implementation. Two examples are [15], where a neural network adaptive controller is used for the transition from horizontal flight to hover, and [16], where a nonlinear dynamic inversion approach is used for formation flight.

Within this context, an autopilot configuration for longitudinal and latero-directional aircraft control based on nonlinear backstepping is presented in this paper: emphasis is given to the control law design and to its real-time hardware/software implementation. Differently from many related studies, the possibility to implement the proposed approach on a microcontroller board allows to actually exploit on a fixed-wing UAV the advantages of the backstepping controller. In fact, the backstepping controller is chosen for its ability to deal with nonlinearities. Unlike from traditional linear control techniques, such as LQ or feedback linearization, a nonlinear control law applied to nonlinear aircraft dynamics guarantees satisfying performance over the whole flight envelope [17]. With backstepping control design, 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 of techniques it has guaranteed convergence of the tracking error and asymptotic stability [19]. The proposed approach is demonstrated to be easily implementable on a microcontroller board suitable for small UAV application.

The paper is organized as follows: Section II describes applications of backstepping controllers to UAVs; Section III briefly presents the aircraft equations of motion and how they are arranged in a suitable form for the backstepping controller. Section IV introduces the control design and strategy. Section V describes the results of software simulation evaluation, Section VI presents the hardware implementation and results of hardware-in-the-loop (HIL) simulations. Finally, Section VII concludes and describes the future work towards achieving flight tests.

Chapter 2

Related Work

A variety of projects illustrate the application of backstepping to fixed-wing aircraft flight control. Unlike with rotorcraft, the difficulty for fixed-wing aircraft is in arranging the equations of motion into the required cascade form. In the literature there are several examples where longitudinal and latero-directional controls are independent. For instance, [20] presents an adaptive backstepping control law for angle of attack tracking, [21] uses adaptive backstepping for UAV velocity and flight path angle control and [22] combines $\mathcal{L}_1$ adaptive methodology with backstepping for longitudinal control of a multi-axis thrust vectoring fighter aircraft. UAV trajectory tracking with adaptive backstepping is described in [23], where velocity and roll angle are controlled. The path-following problem is addressed in [24], where the roll angle command is generated through backstepping with the parameter adaptation technique, hardware-in-the-loop (HIL) simulations validate the results.

The number of works describing combined longitudinal and latero-directional aircraft control is limited. In [25], outer loop variables, incidence, sideslip and roll angles, are controlled by adaptive backstepping with neural networks through body-axes angular rates. In [26], constrained adaptive backstepping with neural adaptation laws is employed for tracking angle of attack, stability-axes roll rate and total velocity while sideslip is maintained at zero.

In contrast to the majority of existing work, an autopilot configuration for combined longitudinal and latero-directional fixed-wing UAV control based on the backstepping technique is presented in this paper. The inner-loop variables angle of attack, sideslip angle and stability-axes roll rate are controlled via the backstepping approach described in [27] and with more details in [28]. 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 will be demonstrated. Nonlinear natural-stabilizing aerodynamic loads are included and employed by the controller. This approach is different from feedback linearization, where these forces are first modeled and then canceled, the nonlinear backstepping approach allows for less accurate knowledge of the aircraft dynamics. Less dynamic outer-loop variables, velocity, altitude and heading, are, for the moment, controlled by PID gains. This choice allows the designer to maintain a clear understanding of the control action, to limit required computational power and to ease the implementation procedure. The main purpose of this work, in fact, is to provide a starting framework for the actual employment of backstepping control technique on microcontrollers for small UAVs. Adaptation and a more advanced outer loop design is beyond the scope of this paper.

A constant in all the backstepping approaches summarized above is the combination of backstepping controller with complex adaptive laws. The benefits of combining nonlinear control with advanced adaptation are clear, but the problems of real-time implementation might be considerable. To our knowledge, aside from [24], none of the adaptive backstepping implementations described above have been performed on microcontrollers suitable for small UAVs. The algorithm described in [29], based on adaptive backstepping for directional control in the presence of crosswind, is currently being implemented; this effort is aided by the limited number of controlled variables and the simplicity of the adaptation approach. What seems to be the only application of the backstepping controller on a flying fixed-wing unmanned aircraft is presented in [30]. There, basic roll and pitch angles hold and trimming tasks are achieved through adaptive backstepping implemented on a Procerus Kestrel autopilot. In the present paper an innovative use of microprocessor technology based on cutting-edge transistor computers is employed to support the controller implementation [31]. The combination of this tool with the proposed control layout strongly facilitates the passage from theoretical simulation to practical application. In fact, HIL simulations validate the control scheme, real-time operation is demonstrated with satisfying flight performances.

Chapter 3

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 [32]. Generic body axes are introduced: X$_B$ and Z$_B$ lie in the aircraft plane of symmetry, with X$_B$ parallel to the fuselage reference line and Z$_B$ directed from the upper to the lower surface of the wing airfoil; the Y$_B$ axis is selected so that the coordinate frame is right-handed. Body axes have origin in the center of gravity and are fixed on the aircraft (Fig. 3.1).

The force equation is:

$$m\dot{\mathbf{V}} = \mathbf{F} - \boldsymbol{\omega} \times m\mathbf{V} \tag{3.1}$$

where m is the aircraft mass. **V** $= (u, v, w)^T$ is the linear velocity vector and $\boldsymbol{\omega} = (p, q, r)^T$ is the angular velocity vector, all expressed in body axes. The force vector **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:

$$I\dot{\boldsymbol{\omega}} = \mathbf{M} - \boldsymbol{\omega} \times I\boldsymbol{\omega} \tag{3.2}$$

where I is the body-axes inertia matrix:

$$I = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{bmatrix}$$

The vector **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:

$$\dot{\Phi} = E(\Phi)\boldsymbol{\omega} \tag{3.3}$$

where:

$$E(\mathbf{\Phi}) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix}$$

The vector $\mathbf{\Phi} = (\phi, \theta, \psi)^T$ contains the Euler angles between the body axes and the North-East-Down (NED) axes reference system. These angles are called, respectively, roll, pitch and yaw. NED axes are centered on the aircraft center of mass. The vertical axis Z_N is directed along the local gravity acceleration vector, X_N points towards North, Y_N points towards East. The X_N and Y_N axes belong to a plane parallel to another plane tangent to the Earth surface at zero altitude.

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]:

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\xi_1 \\ \dot{\xi}_1 = f_1(\mathbf{x}, \xi_1, \xi_2) \\ \dot{\xi}_2 = f_2(\mathbf{x}, \xi_1, \xi_2, \xi_3) \\ \vdots \\ \dot{\xi}_{k-1} = f_{k-1}(\mathbf{x}, \xi_1, ..., \xi_k) \\ \dot{\xi}_k = f_k(\mathbf{x}, \xi_1, ..., \xi_k, u_b). \end{cases} \tag{3.4}$$

In Equation (3.4) $\mathbf{x} \in \mathbb{R}^n$ 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 (3.1)-(3.2)-(3.3), as such, cannot assume the structure of (3.4). Since forces and moments in **F** and **M** are, in general, function of the states **V** and $\boldsymbol{\omega}$ and of the aerodynamic angles and control actions, the cascade form is not respected. 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. 3.1. 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 sideslip angle perturbation is a sign of its latero-directional static stability.

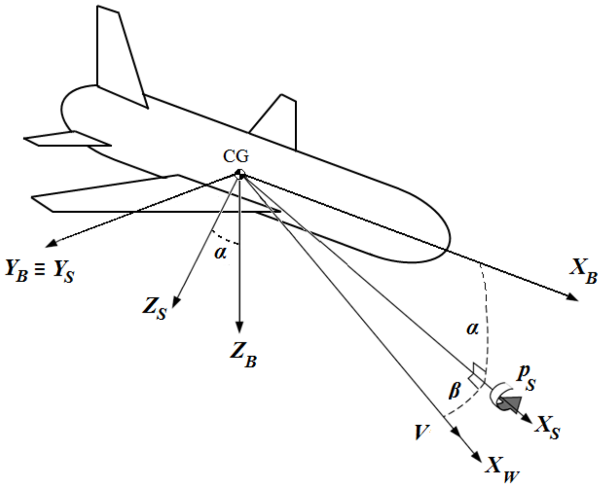


Figure 3.1: Controlled variables and reference axes at t_0

Stability axes are a particular type of body axes where X$_S$ lies along the projection of **V** (at initial reference time t_0) 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 (Fig. 3.1). 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 angular velocities in stability axes $\boldsymbol{\omega}_s = (p_s, q_s, r_s)^T$ as:

$$\boldsymbol{\omega}_s = R_{sb}\boldsymbol{\omega} \tag{3.5}$$

where R_{sb} is the rotation matrix:

$$R_{sb} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}$$

The dynamics considered for the control design are obtained from the force equation written in wind axes, the complete derivation is available in [28]. 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 made:

- **Assumption 1**: The deflection of 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 aerodynamic angles and not on 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, so that control surfaces are far from the aircraft center of gravity [32]. 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 moment derivative 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 Equation (4.11), 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 variation. Assumption 2 is on the conservative side as it targets progressive maneuvers, the ability of backstepping to control aggressive flight will be demonstrated.

The differential equations governing the variation in time of the controlled variables α, β and p_s are now obtained. The definition of the aerodynamic angles is:

$$\alpha = \arctan \frac{w}{u}$$
$$\beta = \arcsin \frac{v}{V}$$

with:

$$V = |\mathbf{V}| = \sqrt{u^2 + v^2 + w^2}$$

The equations relating the derivatives of the aerodynamic angles with the angular velocities and α and β themselves are:

$$
\begin{aligned}
\dot{\alpha} &= q - (p\cos\alpha + r\sin\alpha)\tan\beta + \frac{Z\cos\alpha - (X+T)\sin\alpha + mg_2}{mV\cos\beta} \\
\dot{\beta} &= p\sin\alpha - r\cos\alpha + \frac{Y - T\cos\alpha\sin\beta + mg_3}{mV}
\end{aligned}
\tag{3.6}
$$

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

$$
\begin{aligned}
g_2 &= g(\cos\alpha\cos\theta\cos\phi + \sin\alpha\sin\theta) \\
g_3 &= g(\cos\beta\cos\theta\sin\phi + \sin\beta\cos\alpha\sin\theta - \sin\alpha\sin\beta\cos\theta\cos\phi)
\end{aligned}
\tag{3.7}
$$

where $g = 9.81$ m/s^2 is the gravity acceleration. Equation (3.6) can be written in a more compact and meaningful form. The relationship:

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

is used to include the lift force $Lift$ in the $\dot{\alpha}$ equation. Thanks to Equation (3.5) stability-axes angular rates are introduced in $\dot{\alpha}$ and $\dot{\beta}$ dynamics, the result is:

$$
\begin{aligned}
\dot{\alpha} &= q_s - p_s\tan\beta + \frac{-Lift - T\sin\alpha + mg_2}{mV\cos\beta} \\
\dot{\beta} &= -r_s + \frac{Y - T\cos\alpha\sin\beta + mg_3}{mV}
\end{aligned}
\tag{3.8}
$$

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 Equation (3.8) gives:

$$
\begin{cases}
\dot{p}_s = u_1 \\
\dot{\alpha} = q_s - p_s\tan\beta + \dfrac{-Lift - T\sin\alpha + mg_2}{mV\cos\beta} \\
\dot{q}_s = u_2 \\
\dot{\beta} = -r_s + \dfrac{Y - T\cos\alpha\sin\beta + mg_3}{mV} \\
\dot{r}_s = u_3
\end{cases}
\tag{3.9}
$$

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

Chapter 4

Control Design

4.1 Backstepping Controller

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. Here a controlled variation in the aircraft equilibrium state has a primary effect on the faster dynamics characterizing the attitude, and a secondary effect on the slow-changing variables defining the navigation. Finally, Assumption 4 is very common and generally reasonable, provided that Assumptions 2 and 3 are respected.

Equation (3.9) 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:

$$\dot{p}_s = u_1 \tag{4.1}$$

$$\begin{cases} \dot{\alpha} = q_s - p_s \tan\beta + \dfrac{-Lift - T\sin\alpha + mg_2}{mV\cos\beta} \\ \dot{q}_s = u_2 \end{cases} \tag{4.2}$$

$$\begin{cases} \dot{\beta} = -r_s + \dfrac{Y - T\cos\alpha\sin\beta + mg_3}{mV} \\ \dot{r}_s = u_3 \end{cases} \tag{4.3}$$

three sub-controllers stabilizing the desired states α, β and p_s can be defined. These control laws are designed for a simultaneous action on these three variables, also taking into account cross-coupling effects. In fact, it is possible to observe the presence of p_s and β in the α dynamics and, at the same time, the presence of α in the β dynamics. Because of this coupling, the computation of a control action needs to consider, 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 p_s, controlled by u_1, and β controlled by u_3. This occurrence is beneficial when dealing with maneuvers where strong coupling between longitudinal and latero-directional planes exists.

A simple proportional controller is chosen for p_s, Equation (4.1), while the cascade form of Equation (4.2) and Equation (4.3) allows the application of backstepping controller for α and β.

Note that Equations (4.2) and (4.3) have similar structure:

$$\begin{cases} \dot{\omega}_1 = f(\omega_1, y) + \omega_2 \\ \dot{\omega}_2 = u_s \end{cases} \tag{4.4}$$

A single backstepping controller designed for Equation (4.4) is suitable for Equations (4.2) and (4.3). As it is preferable to have the origin as the desired equilibrium point, a change of variables is 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:

$$\begin{cases} \dot{x}_1 = \Omega(x_1) + x_2 \\ \dot{x}_2 = u_s \end{cases} \tag{4.5}$$

The external control input u_s controls x_2 that, in cascade, acts as virtual control to stabilize x_1. Table 4.1 summarizes the relationships between the variables used in the new and in the original systems. The functions $f_\alpha(\alpha, y_\alpha)$ and $f_\beta(\beta, y_\beta)$ are:

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

Table 4.1: Change of variable relationships

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

As fully demonstrated in [28] through Lyapunov stability theory, a simple globally stabilizing control law for the system of Equation (4.5) is:

$$u_s = -k_u(x_2 + \Psi(x_1)) \tag{4.6}$$

if, for all $x_1 \neq 0$, a constant k_u exists such that:

$$k_u \geq \frac{\Omega(x_1)}{x_1}$$

The function $\Psi(x_1)$ is built so that $\Psi(x_1) = -x_2^{des}$, where x_2^{des} is the desired value for the state x_2 acting as virtual control input for the subsystem x_1. This choice guarantees asymptotic stability for the subsystem x_1:

$$\dot{W}(x_1)|_{x_2=x_2^{des}} = (\Omega(x_1) - \Psi(x_1))\, x_1 < 0, \quad x_1 \neq 0$$

having chosen as temporary control Lyapunov function:

$$W(x_1) = \frac{1}{2}x_1^2$$

Furthermore it can be demonstrated that $\Psi'(x_1)$, the time derivative of $\Psi(x_1)$, is bounded:

$$0 < \Psi'(x_1) < k_u$$

The global control Lyapunov function used to define the control law of Equation (4.6) is:

$$V_L = \int_0^{x_1} -\Psi'(y)\,(\Omega(y) - \Psi(y))\, dy + \frac{1}{2}\,(x_2 + \Psi(x_1))^2$$

which satisfies:

$$\dot{V}_L = -\Psi'(x_1)\left(\Omega(x_1) - \Psi(x_1)\right)^2 - \left(k_u - \Psi'(x_1)\right)\widetilde{x}_2^2$$

where $\widetilde{x}_2 = x_2 - x_2^{des}$.

A linear control is chosen assigning $\Psi(x_1) = k_1 x_1$ so that:

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

with $k_2 > k_1 > \max\{0, k_u\}$. For $k_2 > 2k_1$ the controller is optimal as it minimizes the cost function:

$$\int_0^\infty \left(k_1(\Omega(x1) - k_1 x_1)^2 + \left(\frac{k_2}{2} - k_1\right)(x_2 + k_1 x_1)^2 + \frac{u_s^2}{2k_2}\right) dt$$

Using the relationships of Table 4.1, the control laws for the systems of Equations (4.2) and (4.3) are obtained:

$$\begin{aligned} u_2 &= -k_{\alpha,2}\left(q_s + k_{\alpha,1}\left(\alpha - \alpha^{ref}\right) + f_\alpha(\alpha^{ref}, y_\alpha)\right) \\ u_3 &= k_{\beta,2}\left(-r_s + k_{\beta,1}\beta + f_\beta(0, y_\beta)\right) \end{aligned} \tag{4.7}$$

with:

$$\begin{aligned} k_{\alpha,2} &> 2k_{\alpha,1}, \quad k_{\alpha,1} > \max\{0, k_\alpha\} \\ k_{\beta,2} &> 2k_{\beta,1}, \quad k_{\beta,1} > \max\{0, k_\beta\} \end{aligned} \tag{4.8}$$

where:

$$\begin{aligned} k_\alpha &= \max_{\alpha, y_\alpha} \frac{\partial f_\alpha(\alpha, y_\alpha)}{\partial \alpha} \\ k_\beta &= \max_{\beta, y_\beta} \frac{f_\beta(\beta, y_\beta) - f_\beta(0, y_\beta)}{\beta} \end{aligned}$$

Finally, a proportional control is adopted for p_s:

$$u_1 = k_{p_s}\left(p_s^{ref} - p_s\right), \quad k_{ps} > 0 \tag{4.9}$$

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

$$\mathbf{M}(\boldsymbol{\delta}) = I\left(R_{sb}^T \mathbf{u_c} + \dot{R}_{sb}^T \boldsymbol{\omega}_s\right) + \boldsymbol{\omega} \times I\boldsymbol{\omega} \tag{4.10}$$

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 connected with the attitude of the aircraft. 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 in the inner-loop, three PID controllers act on V, h and ψ in the outer-loop. This approach separates the fast dynamics, characterizing aircraft attitude, from the slower dynamics, characterizing aircraft navigation. The prompt response of the backstepping controller is necessary when dealing with fast-changing inner-loop variables. These, in fact, are of prime importance for the aircraft safety. For instance, an immediate control of α for a UAV affected by vertical gust could prevent the stall and dangerous flight regimes. Consistent with Assumption 3, slower variation of the navigation variables can be successfully handled using traditional PID technique. PID gains are tuned manually following a trial and error approach. The goal is optimizing the response in terms of values of overshoot, rise time, settling time and ringing compatible with aircraft behavior.

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. In particular, standard values for the imposed saturation are the stall angle of attack for α^{ref} and typical roll rate for p^{ref}. Note that the desired roll rate is expressed in body axes (p^{ref}), the conversion to stability axes (p_s^{ref}) is performed with Equation (3.5). 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 Equation (4.10) the most general expressions of the moments:

$$\begin{aligned}
L(\delta_a, \delta_e, \delta_r) &= \frac{1}{2}\rho V^2 Sb \left(C_{l\beta}\beta + C_{l\dot{\beta}}\dot{\beta} + C_{lp}\hat{p} + C_{lr}\hat{r} + C_{l\delta_a}\delta_a + C_{l\delta_e}\delta_e + C_{l\delta_r}\delta_r\right) \\
M(\delta_a, \delta_e, \delta_r) &= \frac{1}{2}\rho V^2 Sc \left(C_{m0} + C_{m\alpha}\alpha + C_{m\dot{\alpha}}\dot{\alpha} + C_{mq}\hat{q} + C_{m\delta_a}\delta_a + C_{m\delta_e}\delta_e + C_{m\delta_r}\delta_r\right) \\
N(\delta_a, \delta_e, \delta_r) &= \frac{1}{2}\rho V^2 Sb \left(C_{n\beta}\beta + C_{n\dot{\beta}}\dot{\beta} + C_{np}\hat{p} + C_{nr}\hat{r} + C_{n\delta_a}\delta_a + C_{n\delta_e}\delta_e + C_{n\delta_r}\delta_r\right)
\end{aligned} \tag{4.11}$$

and solving the resulting linear system with three equations and three unknowns. The nondimensional angular rates $\hat{p}$, $\hat{q}$ and $\hat{r}$ calculated 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 surface. 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, 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, in fact, acts through angular rates. The outer-loop strategy, where control surfaces (as a matter of fact the elevator) control airspeed and throttle controls altitude, is a standard autopilot mode. As explained in [33], this approach guarantees better tracking of the airspeed which is a key parameter for an unmanned aircraft. Table 4.2 summarizes the controlled variables, their command, and the control method.

Table 4.2: 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 and implemented control scheme is shown in Fig. 4.1. The resulting elevator, aileron and throttle control inputs act on the aircraft model. The measures of the controlled states, total speed, altitude and heading angle, are the feedback variables. The 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 inside the backstepping controller with good degree of accuracy integrating Equations (4.1)-(4.3), as later demonstrated in Fig. 5.2.(b). A support to the accurate estimation of α

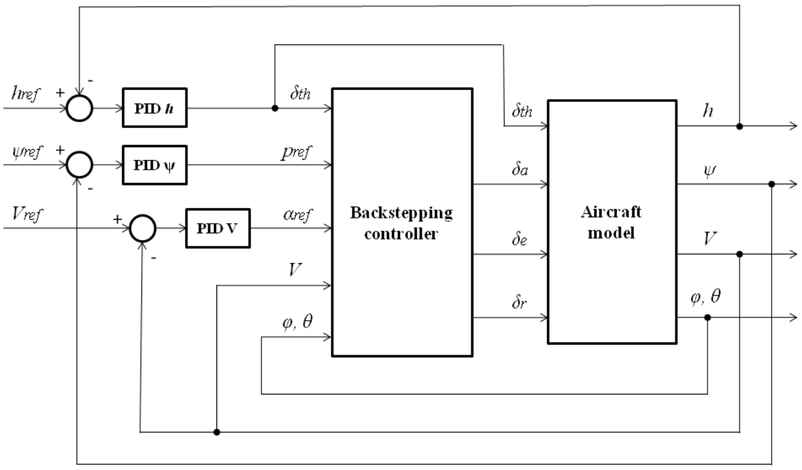


Figure 4.1: Backstepping control strategy for fixed-wing aircraft

and β is provided by the feedback of θ and ψ, which could be provided by an Inertial Measurement Unit. The reason for this unconventional solution lies in the intention of implementing and testing in flight 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.

Chapter 5

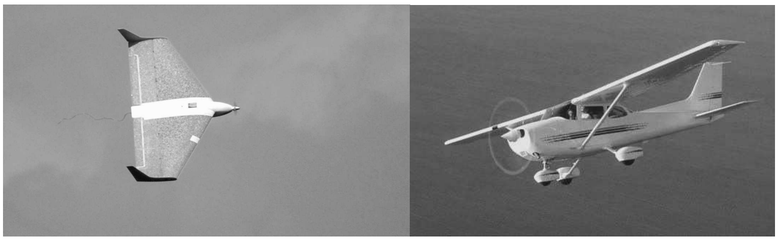
Software-in-the-loop Simulations

Two sets of software-in-the-loop (SIL) simulations are illustrated in this section. The first one is performed in continuous time and demonstrates the capability of the proposed approach to accommodate aircraft of different size and configuration, also in presence of noise. The second set is performed in discrete time on a single aircraft. The ability of the controller to work with slow sampling rate is demonstrated and the simulation tool that will be used for HIL is introduced.

5.1 Continuous Time Simulations

Continuous time simulations are performed in a Matlab Simulink environment. The integration of the equations is performed through a 2nd order Heun method with 0.01 seconds time step. The block scheme follows the structure of Fig. 4.1. The nonlinear equations of motion of [32] are adopted in the aircraft block. Here, actuator transfer functions and a simplified linear motor model are included. Standard Simulink PID blocks are employed. Note that these blocks contain a low pass filter in the derivative action. Its coefficient can be manually set, the default value of $C_{PID} = 100$ is maintained.

At this stage the controller is applied to two nonlinear aircraft models: the MH850 UAV and the full-scale Cessna 172P, Fig. 5.1. The MH850 is characterized by tailless configuration, electric propulsion and non-movable vertical fins at wingtips [34] - [36]. The wingspan is 85 cm, the approximate mass 1 kg and the cruise speed 15 m/s. Aircraft aerodynamic control is achieved with elevons: they control longitudinal motion when symmetrically deflected and latero-directional motion when antisymmetrically deflected. A numerically derived database comprehensive of all aerodynamic derivatives is available to build the nonlinear aircraft model [35] [36]. The Cessna 172 is a single combustion engine aircraft with standard configuration including high-wing and fixed tricycle landing gear. Take off weight is around 880 kg, wingspan 11 m and the length is 8.3 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 include aileron, elevator and rudder. Its choice is motivated by two reasons: i) it is a popular aircraft with much technical data available; and ii) the aircraft is available in the simulator tool employed for HIL simulations. The aircraft characteristics differ considerably not only in terms of absolute weight, dimension and power. Relative values 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 5.1. Testing the controller on a less performing platform is done to prove its universality and identify its limits.



(a) MH850 UAV (b) Cessna 172P

Figure 5.1: Aircraft employed for continuous time simulations

Table 5.1: Aircraft specific properties

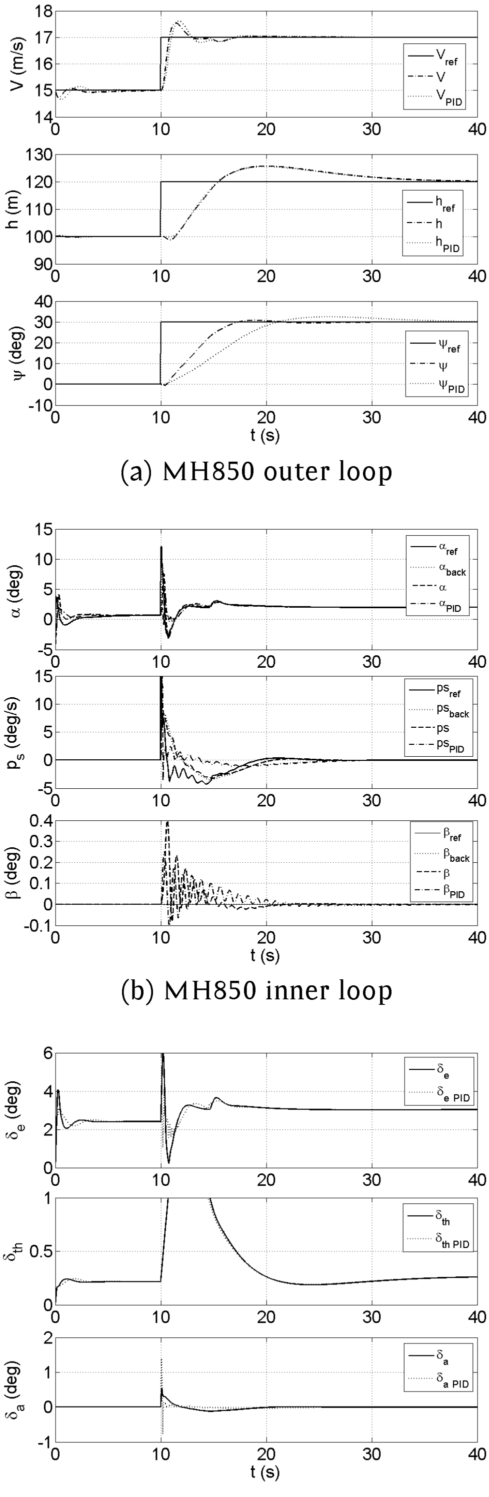
Aircraft	Wing loading [kg/m^2]	Power-to-weight ratio [W/kg]
MH850	4	170
C172P	54.2	134

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 command δ_a, this generates a system of two equations with one unknown which cannot be solved. 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. In fact, for the MH850 UAV $C_{l\delta_a} \approx 10 \cdot C_{n\delta_a}$.

Initially the MH850 response to contemporary step inputs is tested. Reference values are defined as $V^{ref} = 17$ m/s, $h^{ref} = 120$ m and ψ is set to thirty degrees, these are typical figures expected in standard flight conditions. Longitudinal and latero-directional commands are applied at the same time. The outer loop responses are represented in Fig. 5.2.(a). The capability of the controller to effectively control the inner-loop variables with good tracking and short settling time is demonstrated. Note that although no rudder is used, the response on the heading angle ψ is still satisfying with aileron control. Fig. 5.2.(b) and 5.2.(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 real aircraft state. The accurate outer loop velocity tracking is achieved thanks to an excellent angle of attack controlled response in the inner loop. In this case the absolute value of α is bounded to 12 degrees in order to avoid near-stall conditions. Sideslip angle β shows some oscillations originating at the moment of transition. The limited directional damping provided by the vertical fins at the wingtips of the MH850 might be responsible for this. In any case the magnitude of the oscillations is minimal, less than 0.4 degrees, barely noticeable in a real aircraft application. The elevons deflections always remain within the 20 degrees maximum value, throttle saturation is measured only for few seconds after the step time.

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 PID gains. The first determines δ_e according to the output of the PID on V, the other defines δ_a based on the output of the PID on ψ. In the outer loop response the PID performance is almost comparable to backstepping, the V response in slightly more oscillatory while the ψ response is slower and has larger overshoot. Similarly, in the inner loop no significant difference is observable. The commands of the PID controller show, instead, a higher oscillatory behavior for δ_e and δ_a. Altitude and throttle remain basically unchanged because controlled like in the backstepping case. The benefit of backstepping will be clear after the next simulation case.

A validation of the control scheme robustness to aircraft parameters uncertainties is performed with two test cases. In both of them significant variations in aircraft mass, inertia and static margin are considered. The altered parameters are introduced in the nonlinear aircraft model while the controller settings remain unchanged from the nominal case. Case 1



(a) MH850 outer loop

(b) MH850 inner loop

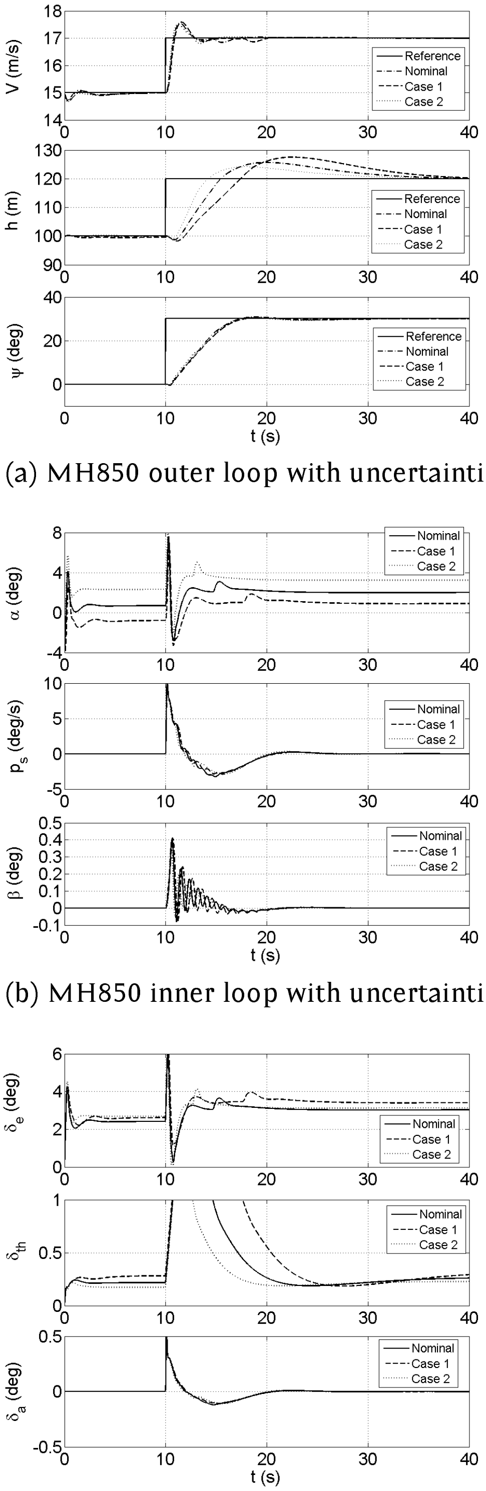
(c) MH850 commands

Figure 5.2: Simulink responses for MH850

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. In both cases the variation in m, I, $C_{m\alpha}$ and $C_{m\delta_e}$ is $\pm 30\%$ from the nominal value. Fig. 5.3 shows the obtained results in comparison with the nominal case. In the outer loop response of Fig. 5.3.(a) V and ψ tracking, which are controlled through backstepping, remain almost unchanged despite the considerable variation of the parameters. A slightly higher oscillatory behavior in V is observable for Case 1 motivated by a lower pitch damping. The altitude response, controlled with throttle 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, Fig. 5.3.(b), lateral oscillations are emphasized in Case 1. It is interesting to observe how the α trim condition changes in the two cases. The commands plot of Fig. 5.3.(c) confirms that full throttle command is required longer for the heavier aircraft of Case 1.

The PID controller previously introduced is tested for the same perturbed aircraft configurations, results are presented in Fig. 5.4. It is evident that the nominal PID controller is not able to withstand the uncertainties introduced in both Case 1 and Case 2. The aircraft loses directional control, it accelerates while quickly losing altitude. It crashes to the ground in less than 10 seconds. Despite backstepping and PID controllers were 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 performances in all cases.

In order to demonstrate the ability of the backstepping controller to withstand sensor noise, the same simulation is performed including this kind of 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. Results of these simulations appear in Fig. 5.5. A comparison with Fig. 5.2 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, in the last 20 seconds of simulation when steady state is achieved, the standard deviation is just 0.074 m/s from



(a) MH850 outer loop with uncertainties

(b) MH850 inner loop with uncertainties

(c) MH850 commands with uncertainties

Figure 5.3: Simulink responses for MH850 in presence of uncertainties

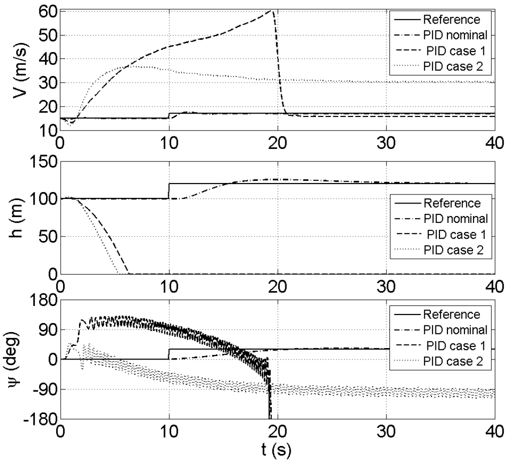
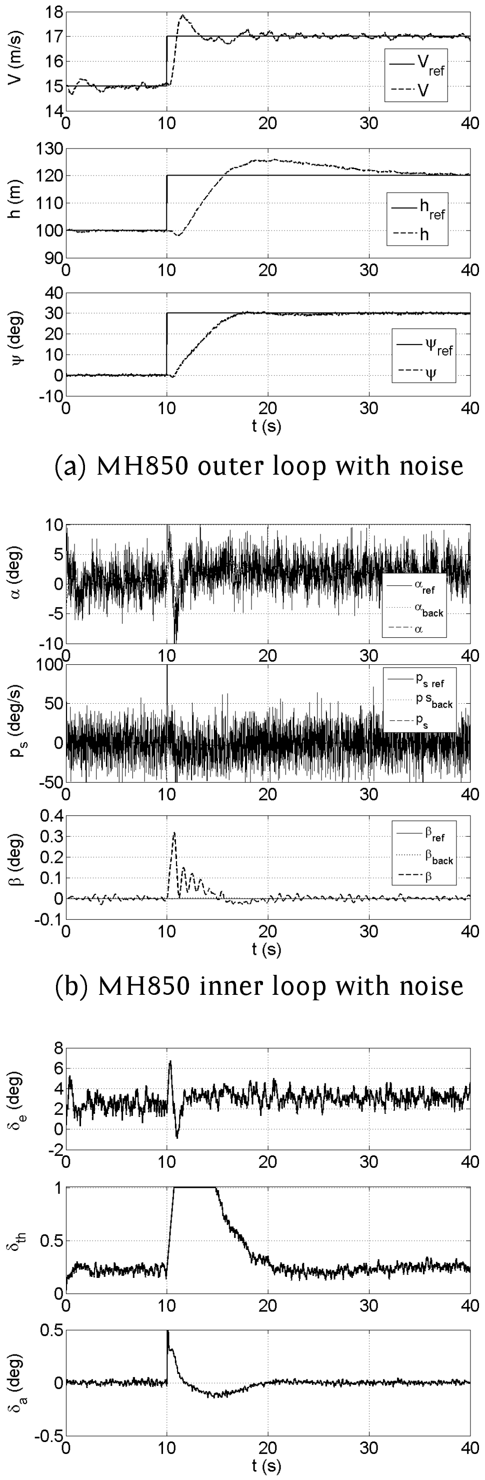


Figure 5.4: Simulink responses for MH850 with PID controller in presence of uncertainties

the mean value, well below the value from the sensor measure. The inner loop is more affected by noise because of the derivative term 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 measured for the aircraft or estimated in the backstepping controller. It is interesting to note that the PID and backstepping gains for the noisy example are very similar to the noise-free case. The major change is the reduction of the derivative gains for the V and h loops. Also the calculated commands, in particular the elevator, are affected by noise, but they still remain compatible with the actuators dynamic response.

C172P responses to ramp inputs are shown in Fig. 5.6.(a). Excellent tracking performance is achieved, in particular for the speed, as previously considered. The different nature of the reference to be followed 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 steady-state altitude error but with a larger error in the climbing phase. This is not due to a problem with the controller but by the lack of power of the C172P. Despite these differences, the same controller configuration is demonstrated to work for aircraft different in size and configuration with adequate results. In particular, in this 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, Fig. 5.6.(b). Note that the minor error in the α tracking, about 0.4 degrees, is motivated by the effect on the elevator of the induced ve-



(a) MH850 outer loop with noise

(b) MH850 inner loop with noise

(c) MH850 commands with noise

Figure 5.5: Simulink responses for MH850 in presence of noise

locity generated by the propeller. The different velocity on the elevator changes its contribution to the aircraft equilibrium, and so it changes the trim angle of attack. This physical phenomenon is included in the aircraft model but not in the simplified backstepping controller aircraft scheme. Commands saturation is achieved for the elevator and the throttle, Fig. 5.6.(c).

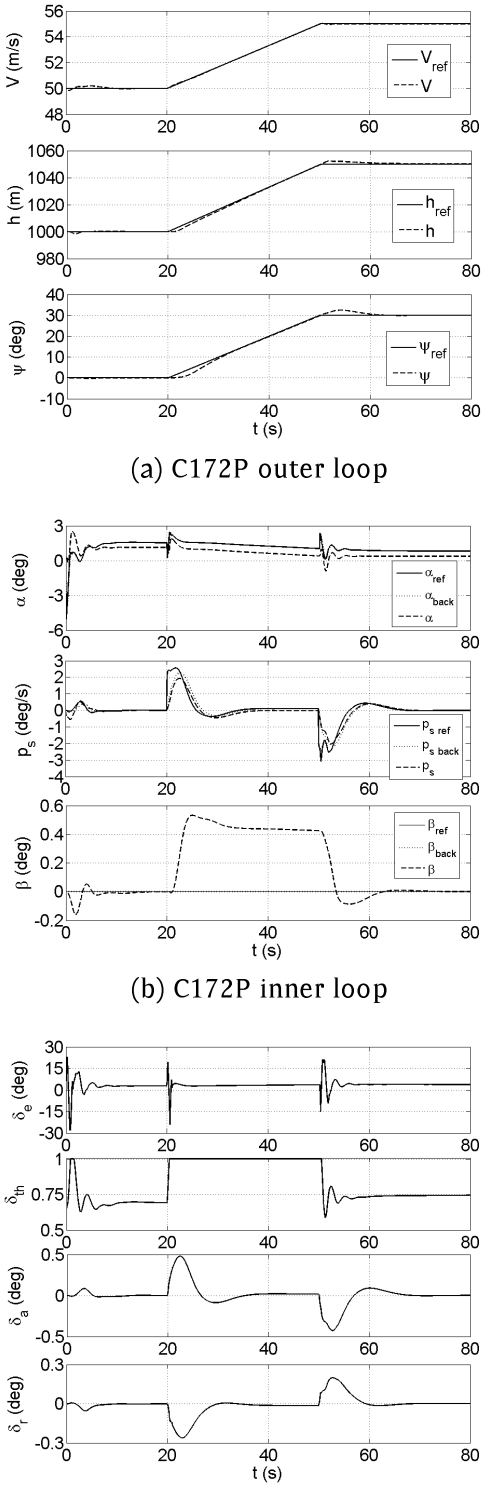
5.2 Discrete Time Simulations

The discretization of the controller is the first necessary step towards real-time implementation. Discrete time simulations are performed with a different approach. The control law is implemented in C code and applied to the C172P model in the FlightGear simulator, see Fig. 5.7

FlightGear is a freeware open-source flight simulator. Developed by volunteers around the world, it offers to academic developers an experienced tool to test their aircraft models and control laws, see for instance [37] - [39]. FlightGear version 2.6.0 is used and the JSBSim flight dynamics library is employed for the C172P. JSBSim is an open source flight dynamics model defining the six-degree of freedom equations which characterize the aircraft motion.

The data transfer between the C application and FlightGear is performed through User Datagram Protocol (UDP). FlightGear provides the value of the feedback variables V, h and ψ, the backstepping controller returns the surface deflections δ_e and δ_a. A 25 Hz frequency is chosen in order to guarantee a consistent data rate compatible with real sensors. Integration is performed with a 2nd order Heun method with 0.01 seconds (100 Hz) time step.

In Fig. 5.8 the results of a complex maneuver are shown. 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 tracking performs the best, the quick response is guaranteed by the choice of using the elevator instead of the throttle for its control. 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, note also the difference in slope between reference and actual values. The slower engine response and the low power-to-weight ratio are responsible for this. In this paper, envisaging the UAV application, priority is given to the speed which is a



(a) C172P outer loop

(b) C172P inner loop

(c) C172P commands

Figure 5.6: Simulink responses for C172P

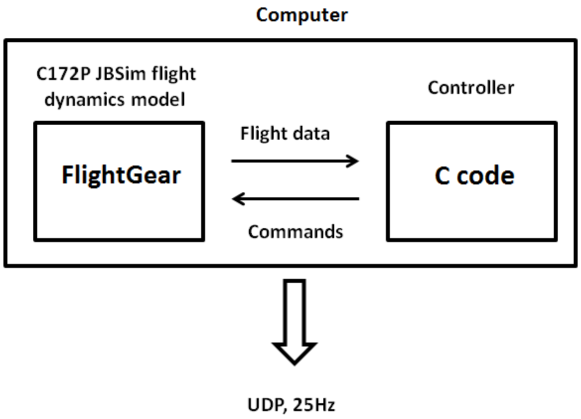


Figure 5.7: Discrete time SIL layout

sensible parameter to avoid stall. Finally, it is interesting to observe how the changes in altitude affect the speed. The corresponding commands are illustrated in Fig. 5.9. The surfaces deflections always remain well within the saturation limits, around 20 degrees for elevator and aileron, 16 degrees for the rudder. The motor instead goes full throttle during the climbing phases.

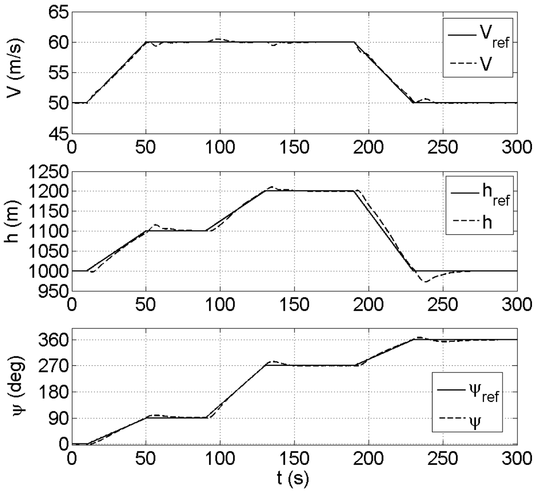


Figure 5.8: FlightGear SIL simulated maneuver for C172P

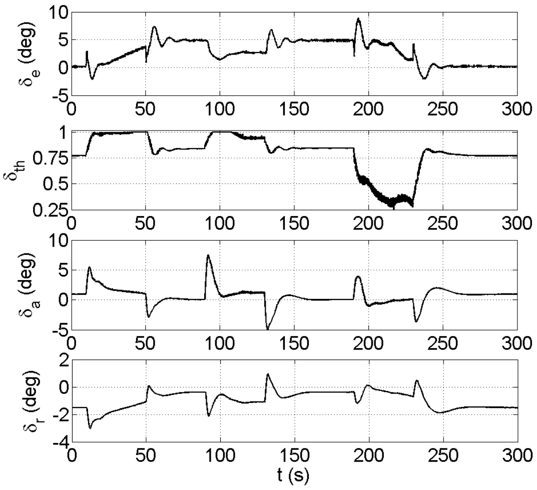


Figure 5.9: FlightGear SIL simulated maneuver commands for C172P

Chapter 6

Hardware-in-the-loop Simulations

The controller is implemented on a control board and tested in real time with the simulator. The chosen microcontroller is the XMOS XK-1A development board, a technology by XMOS Ltd. 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 [31]. An advantage in using the XMOS technology is the ease of programming the board. The language is called XC, the strong similarity with C allows a fast and simple code implementation. The board low cost, limited weight (19 g) and dimensions (50 x 50 mm) make it suitable for small UAV applications.

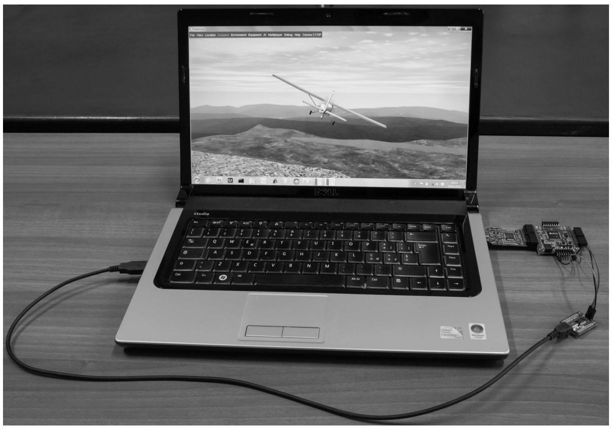


Figure 6.1: HIL setup

The hardware-in-the-loop (HIL) setup and scheme are represented in Fig. 6.1 and Fig. 6.2. 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 153 600 Bd is chosen to maximize data transmission speed and avoid the overlapping of send and receive tasks. The controller computation time is, on the average, slightly less than 0.004 seconds. This result is obtained thanks to the second order Heun integration method and to the simplicity of the operation performed by the control law.

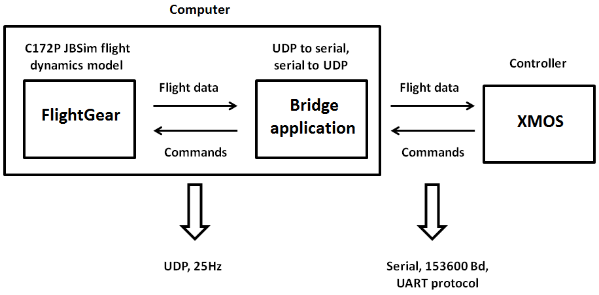


Figure 6.2: HIL layout

HIL simulations are performed with the same reference variables tested for the discrete SIL simulations, backstepping and PID gains are unchanged. Fig. 6.3 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. 6.4, an excellent matching with the commands from the SIL is evident.

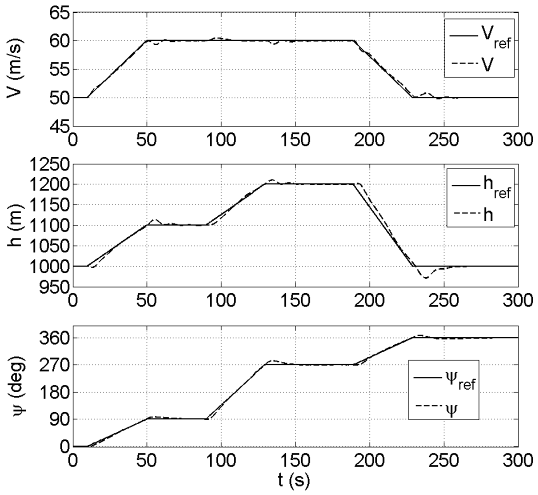


Figure 6.3: FlightGear HIL simulated maneuver for C172P

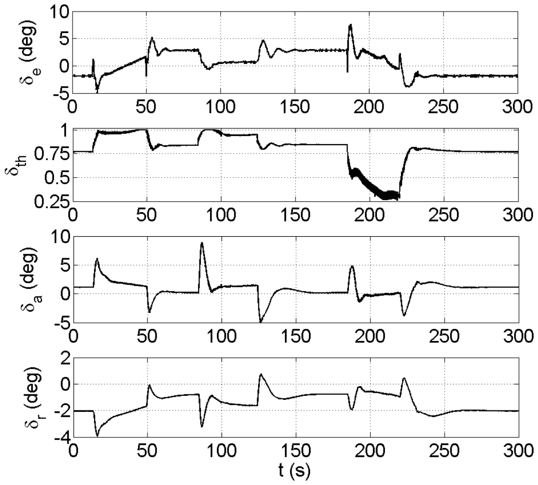


Figure 6.4: FlightGear HIL simulated maneuver commands for C172P

Chapter 7

Discussion, Conclusions and Future Work

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, 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 latero-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 target smooth and progressive maneuvers it is demonstrated that, aircraft allowing, aggressive flight is achievable. Complex maneuvers characterized by severe coupling are performed with little tracking error.

The simple solution adopted differs from the standard adaptive backstepping approach, but it guarantees simple implementation and low computational power without loss of efficacy or robustness. In fact, an innovative real-time implementation on an autopilot board is also demonstrated for a complex maneuver and its performance is satisfying. The control strategy described herein is believed to be implementable on any microcontroller board suitable for small UAV application.

The controller is currently under integration on the Ultra Stick 25e aircraft model, Fig. 7.1. The board successfully communicates with servos and sensors, flight tests are in progress. The upgrade of the controller is also under consideration. The initial control strategy was deliberately kept simple in order to demonstrate the feasibility of the real-time implementation. Adaptive backstepping and substitution of the PID gains with more advanced laws are the changes considered.

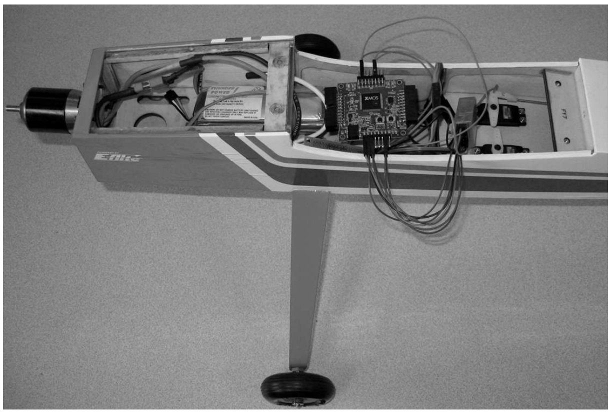


Figure 7.1: Controller integration on the Ultra Stick 25e aircraft

Bibliography

[1] Chao, H. Y., Cao, Y. C., and Chen, Y. Q., "Autopilots for Small Unmanned Aerial Vehicles: A Survey", *International Journal of Control Automation and Systems*, Vol. 8, No. 1, Feb. 2010, pp. 36-44. doi: 10.1007/s12555-010-0105-z

[2] Ollero, A., and Merino, L., "Control and Perception Techniques for Aerial Robotics", *Annual Reviews in Control*, Vol. 28, No. 2, 2004, pp. 167-178. doi: 10.1016/j.arcontrol.2004.05.003

[3] Cominos, P., and Munro, N., "PID Controllers: Recent Tuning Methods and Design to Specification", *IEE Proceedings on Control Theory and Applications*, Vol. 149, No. 1, Jan. 2002, pp. 46-53. doi: 10.1049/ip-cta:20020103

[4] Åström, K. J., and Hägglund, T., *PID Controllers: Theory, Design, and Tuning*, 2nd ed., ISA, Durham, NC, USA, 1995.

[5] Kang, Y., and Hedrick, J. K., "Linear Tracking for a Fixed-Wing UAV Using Nonlinear Model Predictive Control", *IEEE Transactions on Control Systems Technology*, Vol. 17, No. 5, Sept. 2009, pp. 1202-1210. doi:10.1109/TCST.200820044878

[6] Santoso, F., Liu, M., and Egan, G., "H_2 and H_∞ robust autopilot synthesis for longitudinal flight of a special unmanned aerial vehicle: a comparative study", *IET Control Theory & Applications*, Vol. 2, No. 7, July 2008, pp. 583-594. doi: 10.1049/iet-cta:20070415

[7] Fradkov, A., and Andrievsky, B., “Combined Adaptive Controller for UAV Guidance”, *European Journal of Control*, Vol. 11, No. 1, 2005, pp. 71-79.

[8] Tu, H., and Du, X., “The Design of Small UAV Autopilot Hardware System Based on DSP”, *Proceedings of the 2010 Intenational Conference on Intelligent Computation Technology and Automation (ICICTA)*, Vol. 3, Changsha, China, May 2010, pp. 780-783. doi: 10.1109/ICICTA.2010.648

[9] Guanglin, H., Rujun, G., and Shi, Y., “Application of FPGA in Small UAV Autopilot Based on Embedded Linux System, *Proceeedings of the 33rd Annual Conference on the IEEE Industrial Electronics Society*, Taipei, Taiwan, Nov. 2007, pp. 731-734. doi: 10.1109/IECON.2007.4459959

[10] Kendoul, F., “Survey of Advances in Guidance, Navigation, and Control of Unmanned Rotorcraft Systems”, *Journal of Field Robotics*, Vol. 29, No. 2, Mar./Apr. 2012, pp. 315-378. doi: 10.1002/rob.20414

[11] Paw, Y. C., and Balas, G. J., “Development and Application of an Integrated Framework for Small UAV Flight Control Development”, *Mechatronics*, Vol. 21, No. 5, Aug. 2011, pp. 789-802. doi: 10.1016/j.mechatronics.2010.09.009

[12] Lizarraga, M. I., Elkaim, G. H., Horn, G. M., Curry, R., Dobrokhodov, V., and Kaminer, I., “Low Cost Rapidly Reconfigurable UAV Autopilot For Research and Development of Guidance, Navigation and Control Algorithms”, *Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Vol. 3, San Diego, CA, USA, Aug./Sept. 2009, pp. 585-594.

[13] Jung, D., 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 and 2005 European*

Control Conference, Seville, Spain, Dec. 2005, pp. 2739-2744. doi: 10.1109/CDC.2005.1582577

[14] 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*, Vol. 4, No. 5, May 2007, pp. 816-835. doi: 10.2514/1.26553

[15] Johnson, E. N., Turbe, M. A., Wu, A. D., Kannan, S. K., and Neidhoefer, J. C., "Flight Test Results of Autonomous Fixed-Wing UAV Transitions to and from Stationary Hover", *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, USA, Aug. 2006. doi: 10.2514/6.2006-6775

[16] 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*, Vol. 14, No. 6, Nov. 2006, pp. 1105-1112. doi: 10.1109/TCST.2006.880203

[17] Härkegård, O., "Backstepping Designs for Aircraft Control - What is there to gain?", Division of Automatic Control, Department of Electrical Engineering Linköpings Universitet, Sweden, Tech. Rep. LiTH-ISY-R-2339, Mar. 2001.

[18] M. Krstić, I. Kanellakopoulos and P. Kokotović, *Nonlinear and Adaptive Control Design*, 1st ed., Wiley, New York, 1995.

[19] Kim, K-S., and Kim, Y., "Robust Backstepping Control for Slew Maneuver Using Nonlinear Tracking Function", *IEEE Transactions on Control Systems Technology*, Vol. 11, No. 6, Nov. 2003, pp. 822-829. doi: 10.1109/TCST.2003.815608

[20] Ju, H-S., and Tsai, C-C., "Longitudinal Axis Flight Control Law Design by Adaptive Backstepping", *IEEE Transactions on Aerospace*

and Electronic Systems, Vol. 43, No. 1, Jan. 2007, pp. 311-329. doi: 10.1109/TAES.2007.357136

[21] Gavilan, F., Acosta, J. Á., and Vazquez, R., "Control of the longitudinal flight dynamics of an UAV using adaptive backstepping", *Proceedings of the 18th IFAC World Congress*, Vol. 18, No. 1, Milan, Italy, Aug./Sept. 2011, pp. 1892-1897. doi: 10.3182/20110828-6-IT-1002.01876

[22] Liu, K., Zhu, J., and Yu, B., "Longitudinal Controller Design for a Fighter Aircraft Using $\mathcal{L}_1$ Adaptive Backstepping", *Proceedings of the 2011 9th World Congress on Intelligent Control and Automation*, Taipei, Taiwan, June 2011, pp. 341-346. doi: 10.1109/WCICA.2011.5970531

[23] Ren, W., and Atkins, E., "Nonlinear Trajectory Tracking for Fixed Wing UAVs via Backstepping and Parameter Adaptation", *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, CA, USA, Aug. 2005. doi: 10.2514/6.2005-6196

[24] Jung, D., and Tsiotras, P., "Bank-to-Turn Control for a Small UAV Using Backstepping and Parameter Adaptation", *Proceeding of the 17th IFAC World Congress*, Vol. 17, No. 1, Seoul, South Korea, July 2008, pp. 4406-4411. doi: 10.3182/20080706-5-KR-1001.00742

[25] Lee, T., and Kim, Y., "Nonlinear Adaptive Flight Control Using Backstepping and Neural Networks Controller", *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 4, July 2001, pp. 675-682. doi: 10.2514/2.4794

[26] Sonneveldt, L., Chu, Q. P., and Mulder, J. A., "Nonlinear Flight Control Design Using Constrained Adaptive Backstepping", *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 2, Mar. 2007, pp. 322-336. doi: 10.2514/1.25834

[27] Härkegård, O., Torkel Glad S., "Flight Control Design Using Backstepping", *Proceedings of the 5^th IFAC Symposium*, St. Petersburg, Russia, July 2001.

[28] Härkegård, O., "Flight Control Design Using Backstepping", Ph.D. dissertation, Dep. of Electrical Eng., Linköping Univ., Linköping, Sweden, 2001.

[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*, Vol. 69, No. 1-4, Jan. 2013, pp. 257-271. doi: 10.1007/s10846-012-9756-8

[30] Matthews, J. S., Knoebel, N. B., Osborne, S. R., Beard, R. W., and Eldredge, A., "Adaptive Backstepping Control for Miniature Air Vehicles", *Proceedings of the American Control Conference*, Minneapolis, MN, USA, June 2006. doi: 10.1109/ACC.2006.1657678

[31] Martins, G., Moses, A., Rutherford, M. J., and Valavanis, K. P., "Enabling intelligent unmanned vehicles through XMOS Technology", *Journal of Defense Modeling and Simulation Applications, Methodology, Technology*, Vol. 9, No. 1, Jan. 2012, pp. 71-82. doi: 10.1177/1548512910388197

[32] Etkin, B., and Reid, L. D., *Dynamics of Flight: Stability and Control*, 3^{rd} ed. Wiley, New York, 1996.

[33] Guglieri, G., "Effect of Autopilot Modes on Flight Performances of Electric Mini-UAVs", *The Aeronautical Journal*, Vol. 117, No. 1187, Jan. 2013, pp. 57-69.

[34] 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*, Paris, France, June 2010.

[35] 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*, Vol. 65, No. 1-4, Jan. 2012, pp. 43-61. doi: 10.1007/s10846-011-9565-5

[36] 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*, Vol. 69, No. 1-4, Jan. 2013, pp 109-118. doi: 10.1007/s10846-012-9717-2

[37] Park, H., Kim, D-H., Kim, J-G., and Chang, C-H., "Experimental Evaluation of Unmanned Aerial Vehicle System Software Based on the TMO Model", *Journal of Computing Science and Engineering*, Vol. 2, No. 4, Dec. 2008, pp. 357-374.

[38] Sorton, E. F., and Hammaker, S., "Simulated Flight Testing of an Autonomous Unmanned Aerial Vehicle Using FlightGear", *Proceedings of AIAA Infotech@Aerospace*, Arlington, VA, USA, Sept. 2005. doi: 10.2514/6.2005-7083

[39] Jung, D., and Tsiotras, P., "Modeling and Hardware-in-the-Loop Simulation for a Small Unmanned Aerial Vehicle", *Proceedings of AIAA Infotech@Aerospace*, Rohnert Park, CA, USA, May 2007. doi: 10.2514/6.2007-2768