Incremental nonlinear control for attitude tracking of a fixed-wing UAV - simulation and practical implementation

Afonso D'Alberto Gonçalves Canas Mendes
afonsocmendes@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Portugal

November 2017

Abstract

In this work, two well-studied nonlinear control techniques - Nonlinear Dynamics Inversion and Backstepping - are taken into consideration. In particular, they are complemented with an increasingly popular robust control technique that considers the incremental dynamics of the vehicle in order to capture changes in its aerodynamics through angular acceleration feedback instead of complex modelling. The approach yields increased robustness to model uncertainties and improved tracking performance, at the expense of additional sensor information and fast control actuation. Additionally, command filters are considered in order to protect the design against constraining actuator dynamics. Both robust approaches are properly tuned, tested and compared in closed-loop simulations. Robustness tests for one selected technique are performed to validate its integration into the open-source Ardupilot© real-time software. Simulation results using the JSBSIM flight simulator are obtained to further certify the integrated solution for practical experimentation.
Keywords: Nonlinear Dynamics Inversion, Backstepping, Incremental Control, Command Filtering, Ardupilot

1. Introduction

The inherently compact structure, high manoeuvrability and low-cost production of unmanned aerial vehicles (UAVs) represent only some of the reasons why these systems have become increasingly popular in recent years. As the corresponding market expands, it is expected that regulatory authorities establish tighter bounds on the performance of these systems to mostly ensure public safety but also long-term integration into the civil airspace environment. In turn, flight safety is closely related to the performance of the automatic flight controllers.

The most commonly used approach by the industry to design flight controllers is to use linear models of the aircraft and employ the various tools from linear control theory to obtain a linear controller. Since most real systems are inherently nonlinear, linear descriptions are approximations with limited validity and so are the controllers which are developed using those descriptions.

On the other hand, advances in computer and microprocessor technology have rendered the implementation of nonlinear control systems a simple matter. Since this type of control techniques concerns generic nonlinear descriptions of the system to be controlled, it is likely to be more efficient than linear control and valid for a larger set of conditions.

Nonlinear Dynamics Inversion (NDI) is a control methodology that started being developed in the late 1970s by independent authors who first addressed the problem of linearising a nonlinear system through state feedback [6].

Backstepping (BKS) is a systematic, Lyapunov-based control design method for a broad class of nonlinear systems [6]. Its origins can be traced back to the early 1990s, when its development experienced a rapid acceleration due to the ease that the technique showed to have in dealing with unknown parameters and external disturbances.

A common drawback to NDI and BKS designs is the dependency on model knowledge. The nominal character of these techniques makes them, in their simplest forms, sensitive to model uncertainties.

One particular nonlinear and robust control solution that has been continuously gaining momentum in recent years is incremental control. Development of nonlinear control theory based on increments of nonlinear control action can be traced back to the late 1990s by the papers of [9] and others. It has been studied in recent years for atmospheric flight control of fixed-wing aircraft [7] and helicopters [8] and also for space applications [1]. It exploits the advancements in sensor technology (and general hardware) to reduce the control syn-

thesis dependency on nominal aircraft model data and therefore enhance its flexibility and robustness properties, while minimizing the necessity for adaptive elements in the flight control system.

Applications of Incremental BKS (IBKS) can be found in [11] for an F-16 aircraft, in [3] for missile control and in [2] for spacecraft attitude control.

Incremental NDI (INDI) [9] has also found practical applications both for aircraft [7, 8] and spacecraft [1] flight control.

This project fits in the context presented this far as it concerns the study of INDI and IBKS and their applicability to the flight control problem of fixed-wing aircraft. It includes a comparison between the approaches based on simulation results and the promotion of one selected technique for integration in the real-time software Ardupilot© .

This paper is structured as follows: in Section 2, standard NDI and BKS control laws are reviewed, followed by a short presentation on incremental dynamics and command filtering. Section 3 details the fixed-wing UAV rotational model and the derivation of the command filtered INDI and IBKS control laws for attitude control. In Section 4, MATLAB/Simulink® implementation results are discussed and in Section 5 the integration of one incremental controller into the Ardupilot© framework is briefly commented. Section 6 provides the main conclusions and some final remarks.

2. Nonlinear Control approaches

2.1. Nonlinear Dynamics Inversion

Consider the following affine-in-control, square, multivariable, generic, nonlinear, n-th order system:

$$\dot{\underline{x}} = \underline{f}(\underline{x}) + \underline{G}(\underline{x})\,\underline{u} \qquad , \qquad \underline{y} = \underline{h}(\underline{x}) \tag{1}$$

where $\underline{x} \in \mathbb{R}^n$ is the state vector, $\underline{u} \in \mathbb{R}^m$ is the input vector, $\underline{y} \in \mathbb{R}^m$ is the output vector, $\underline{f}$ and $\underline{h}$ are smooth vector fields and $\underline{G}$ is an $n \times m$ matrix whose column are smooth vector fields in $\mathbb{R}^n$. The NDI control design procedure consists in finding a direct relation between the output $\underline{y}$ and the input $\underline{u}$ and invert it. By differentiating the output with respect to time, one obtains:

$$\begin{aligned}\dot{\underline{y}} &= \frac{\mathrm{d}\underline{h}(\underline{x})}{\mathrm{d}\underline{x}}\,\dot{\underline{x}} = \nabla\underline{h}(\underline{x})\left[\underline{f}(\underline{x}) + \underline{G}(\underline{x})\,\underline{u}\right] \\ &= L_f\underline{h}(\underline{x}) + L_g\underline{h}(\underline{x})\,\underline{u}\end{aligned} \tag{2}$$

where $L_f\underline{h}$ denotes the first-order Lie derivative of $\underline{h}$ in the direction of $\underline{f}$ and is used to simplify the notation. If $L_g\underline{h}(\underline{x})$ is invertible, the control law:

$$\underline{u} = L_g\underline{h}(\underline{x})^{-1}\left[\underline{\nu} - L_f\underline{h}(\underline{x})\right] \tag{3}$$

linearises the nonlinear system from the virtual control $\underline{\nu}$ to $\underline{y}$, i.e., $\underline{\nu} = \dot{\underline{y}}$, which can be used to render the closed-loop asymptotically stable with linear control, for example. If the reference for $\underline{y}$ is denoted by $\underline{y}_d$ and the tracking error by $\underline{e} = \underline{y}_d - \underline{y}$, then the linear control law with feedforward term $\underline{\nu} = \underline{K}\underline{e} + \dot{\underline{y}}_d$ yields:

$$\dot{\underline{e}} + \underline{K}\underline{e} = \underline{0} \tag{4}$$

If $\underline{K}$ is a positive definite matrix, then the error dynamics is asymptotically stable.

If $L_g\underline{h}(\underline{x})$ is not invertible, consecutive time-differentiations are performed until an explicit relation between every output $y_i \in \underline{y}$ and the input $\underline{u}$ is found. If the total number of differentiations is not equal to n, then the system possesses so-called internal dynamics, which cannot be controlled by the input vector and their behaviour must be considered in order to guarantee closed-loop stability.

For higher-order, cascaded systems, a common approach to simplify NDI design is to study the existence of a time-scale separation between the different subsystems and exploit that separation to design NDI control laws individually, as explained in [7]. This simplification reduces the number of differentiations that need to be performed to obtain feedback linearisation, facilitating the design.

2.2. Backstepping

For the sake of simplicity, consider the following generic, strict-feedback, nonlinear system:

$$\begin{aligned}\dot{\underline{x}}_1 &= \underline{f}_1(\underline{x}_1) + \underline{G}_1(\underline{x}_1)\,\underline{x}_2 \\ &\vdots \\ \dot{\underline{x}}_i &= \underline{f}_i(\underline{x}_1, \cdots, \underline{x}_i) + \underline{G}_i(\underline{x}_1, \cdots, \underline{x}_i)\,\underline{x}_{i+1} \\ &\vdots \\ \dot{\underline{x}}_k &= \underline{f}_k(\underline{x}_1, \cdots, \underline{x}_k) + \underline{G}_k(\underline{x}_1, \cdots, \underline{x}_k)\,\underline{u}\end{aligned} \tag{5}$$

where it is assumed that every $\underline{x}_i$-subsystem is of the same order n and that every state in each subsystem has a relative degree of one. Then, using Lyapunov stability concepts, it can be shown [6] that for a candidate Lyapunov function of the form:

$$V_k = \frac{1}{2}\sum_{i=1}^{k} \underline{z}_i^\top\,\underline{z}_i \tag{6}$$

where $\underline{z}_i$ are the tracking tracking errors, given by:

$$\underline{z}_i = \underline{x}_i - \underline{\alpha}_{i-1} \tag{7}$$

the system in Eq.(5) can achieve global asymptotic tracking of a smooth and bounded reference signal, with the first n time-derivatives known and bounded, $\underline{x}_d$ for $\underline{x}_1$, if the following stabilizing control laws are used:

$$\begin{aligned}\underline{\alpha}_1 &= \underline{G}_1^{-1}\left[-\underline{C}_1\underline{z}_1 - \underline{f}_1 + \dot{\underline{x}}_d\right] \\ \underline{\alpha}_i &= \underline{G}_i^{-1}\left[-\underline{C}_i\underline{z}_i - \underline{f}_i + \dot{\underline{\alpha}}_{i-1} - \underline{G}_{i-1}^\top\underline{z}_{i-1}\right]\end{aligned} \tag{8}$$

for $i = 2, \cdots, k$, with each $\underline{C}_i$ and $\underline{C}_1$ being diagonal positive definite matrices and where the final Backstepping control law is given by $\underline{u} = \underline{\alpha}_k$.

2.3. Incremental control

Consider now a generic, n-th order, nonlinear system given by:

$$\dot{\underline{x}} = \underline{\mathscr{f}}\left(\underline{x}, \underline{u}\right) \qquad , \qquad \underline{y} = \underline{x} \tag{9}$$

Take the first-order Taylor series expansion of Eq.(9) around the current augmented state $(\underline{x}_0, \underline{u}_0)$:

$$\begin{aligned}\dot{\underline{x}} &\approx \dot{\underline{x}}_0 + \left.\frac{\partial \underline{\mathscr{f}}}{\partial \underline{x}}\right|_{(\underline{x}_0, \underline{u}_0)} (\underline{x} - \underline{x}_0) + \left.\frac{\partial \underline{\mathscr{f}}}{\partial \underline{u}}\right|_{(\underline{x}_0, \underline{u}_0)} (\underline{u} - \underline{u}_0) \\ &= \dot{\underline{x}}_0 + \underline{F}_x \left(\underline{x} - \underline{x}_0\right) + \underline{F}_u \left(\underline{u} - \underline{u}_0\right)\end{aligned} \tag{10}$$

For high sampling frequencies and fast control actuation, the state variation can be neglected and the former result reduces to:

$$\dot{\underline{x}} \approx \dot{\underline{x}}_0 + \underline{F}_u \left(\underline{u} - \underline{u}_0\right) \tag{11}$$

assuming that it is possible to obtain the state derivative feedback $\dot{\underline{x}}_0$ and that the actuator state is observable. This results allows modifying the NDI control law of equation Eq.(3) to:

$$\underline{u} = \underline{u}_0 + \underline{F}_u^{-1} \left[\underline{\nu} - \dot{\underline{x}}_0\right] \tag{12}$$

and the final Backstepping control law of Eq.(8) to:

$$\underline{u} = \underline{u}_0 + \underline{F}_u^{-1} \left[-\underline{C}_k \underline{z}_k - \dot{\underline{x}}_0 + \dot{\underline{\alpha}}_{k-1} - \underline{G}_{k-1}^{\top} \underline{z}_{k-1}\right] \tag{13}$$

where $\underline{x}_0$ is used to designate the current observation of the state $\underline{x}_k$.

It is noticeable that the effect of considering incremental dynamics removes the necessity of a model for the state-dependent dynamics of the plant, $\underline{f}\left(\underline{x}\right)$, both in the NDI and in the BKS control laws, which results in increased robustness against model mismatch. This simplification is achieved at the expense of requirements of fast control actuation, high sampling frequency and increased state observability.

2.4. Command filtering

Since the end goal of this work is practical implementation of an incremental NDI (INDI) or incremental BKS (IBKS) controller, actuator dynamics are an important aspect to consider. In reality, they can include magnitude bounds, rate limitations and other phenomena that deviates them from an ideal, instant response behaviour. Therefore, command filters are considered now to take those limitations into account and protect the closed-loop system against reference signals that force the system beyond its capabilities.

2.4.1 Command filtered Backstepping

Command filtered (CF) Backstepping is formally presented in [5] and consists, basically, in defining modified tracking errors

$$\tilde{\underline{z}}_i = \underline{z}_i - \underline{\chi}_i \tag{14}$$

and guaranteeing global asymptotic regulation of those modified errors instead. The variables $\underline{\chi}_i$ are used as estimates of the effect of the command filters and are obtained from stable linear filters:

$$\dot{\underline{\chi}}_i = -\underline{C}_i \underline{\chi}_i + \underline{G}_i \left[\underline{x}_{i+1,r} - \underline{\alpha}_i + \underline{\chi}_{i+1}\right] \tag{15}$$

where $\underline{x}_{i,r}$ is the output of the i-th CF and represents a filtered estimation of the reference signal for the $\underline{x}_i$ state, $\underline{\alpha}_{i-1}$, that takes into account the known limitations of the system. The reader is referred to [5] for the proof of how the control laws:

$$\begin{aligned}\underline{\alpha}_1 &= \underline{G}_1^{-1} \left[-\underline{C}_1 \underline{z}_1 - \underline{f}_1 + \dot{\underline{x}}_d\right] \\ \underline{\alpha}_i &= \underline{G}_i^{-1} \left[-\underline{C}_i \underline{z}_i - \underline{f}_i + \dot{\underline{x}}_{i,r} - \underline{G}_{i-1}^{\top} \tilde{\underline{z}}_{i-1}\right]\end{aligned} \tag{16}$$

for $i = 2, \cdots, k$ can be used in combination with the CFs to obtain a globally asymptotically stable closed -loop system in terms of the modified tracking errors. The CFs are also useful to provide the derivatives of these control laws.

2.4.2 Command filtered NDI

The approach of command filtering for NDI is similar to that of BKS and is even more relevant for multi-loop designs that exploit the time-scale separation principle because then the CFs can also be used to obtain the feedforward terms for the linear control laws. For the system in equation Eq.(1), the control signal obtained from the dynamics inversion in Eq.(3), here denoted by $\underline{u}^0$, is filtered by the CF before being directed to the nonlinear system. Then, the stable linear filter:

$$\dot{\chi} = -\underline{K}\underline{\chi} + L_g \underline{h}\left(\underline{x}\right) \left(\underline{u} - \underline{u}^0\right) \tag{17}$$

is used to estimate the loss of authority of the controller due to constraints imposed by the CF and the tracking error is hedged with $\underline{\chi}$ to obtain a modified tracking error that is then used by the linear control law. It can be shown that the dynamics of this modified error $\tilde{\underline{e}} = \underline{e} - \underline{\chi}$ becomes:

$$\dot{\tilde{\underline{e}}} = -\underline{K}\tilde{\underline{e}} + \underline{K}\underline{\chi} \tag{18}$$

revealing that it converges asymptotically to zero when the CF does not impose any constraints on the control signal and is bounded otherwise.

3. Fixed-wing attitude control

3.1. Rotational model

In vector notation, the Euler equations of rotational motion for a rigid body with constant inertial mass can be expressed by:

$$\underline{M} = \underline{J}\dot{\underline{\omega}} + \underline{\omega} \times \underline{J}\underline{\omega} \tag{19}$$

where $\underline{M}$ is the total external moment acting on the system, $\underline{\omega} = \left[p\,, q\,, r\right]^{\top}$ is its angular velocity in the body frame and $\underline{J}$ is the inertia tensor. The external moment is assumed to be decomposed in

contributions from an aerodynamic model, $\underline{M}_{aero}$, and the propulsion system, $\underline{M}_{prop}$. In turn, the aerodynamic moments are given by:

$$\underline{M}_{aero} = \underline{M}_a + \underline{r}_{ac} \times \underline{F}_{a,b} \tag{20}$$

where $\underline{r}_{a,c}$ is the relative position of the mean aerodynamic centre of the aircraft with respect to its CG, $\underline{M}_a$ are pure aerodynamic torques and $\underline{F}_{a,b}$ are the aerodynamic forces written in the body frame, which can be obtained from the wind-axes forces $\underline{F}_{a,w}$ through a simple rotation matrix [4]:

$$\underline{F}_{a,b} = \underline{T}_{bw}(\alpha, \beta)\, \underline{F}_{a,w} \tag{21}$$

with α being the angle of attack and β the sideslip angle. The propulsion moment has a similar decomposition. Defining the control input as $\underline{u} = [\delta_a, \delta_e, \delta_r]^\top$, with δ_a, δ_e and δ_r being the aileron, elevator and rudder deflections, respectively, and assuming a linear relation between the aerodynamic forces and moments and $\underline{u}$, as in:

$$\underline{M}_a = \underline{M}_a^0 + \underline{M}_u \underline{u} \quad \text{and} \quad \underline{F}_{a,w} = \underline{F}_{a,w}^0 + \underline{F}_u \underline{u} \tag{22}$$

the rotational dynamics equation becomes:

$$\begin{aligned}\dot{\underline{\omega}} = {} & \underline{J}^{-1}\left[\underline{M}_a^0 + \tilde{\underline{R}}_{ac}\underline{T}_{bw}\underline{F}_{a,w}^0 + \underline{M}_{prop} - \tilde{\underline{\Omega}} J \underline{\omega}\right] \\ & + \underline{J}^{-1}\left[\underline{M}_u + \tilde{\underline{R}}_{ac}\underline{T}_{bw}\underline{F}_u\right]\end{aligned} \tag{23}$$

with $\tilde{\underline{R}}_{ac}$ denoting $(\underline{r}_{ac}\times)$ and $\tilde{\underline{\Omega}}$ denoting $(\underline{\omega}\times)$. In this work, the attitude vector will be considered to include the pitch θ and roll ϕ angles and the sideslip angle instead of the yaw angle. Using kinematic relations [4, 11], the final rotational model is given by:

$$\dot{\underline{x}}_1 = \underline{f}_1(\underline{x}_1) + \underline{G}_1(\underline{x}_1)\,\underline{x}_2 \tag{24a}$$

$$\dot{\underline{x}}_2 = \underline{f}_2(\underline{x}_1, \underline{x}_2) + \underline{G}_2(\underline{x}_1, \underline{x}_2)\,\underline{u} \tag{24b}$$

where the state vectors are $\underline{x}_1 = [\phi, \theta, \beta]^\top = \underline{\varphi}_\beta$ and $\underline{x}_2 = \underline{\omega}$ and:

$$\begin{aligned}\underline{f}_1 &\equiv \begin{bmatrix}0 & 0 & f_\beta\end{bmatrix}^\top \\ \underline{G}_1 &\equiv \begin{bmatrix} 1 & \mathrm{s}_\phi \mathrm{t}_\theta & \mathrm{c}_\phi \mathrm{t}_\theta \\ 0 & \mathrm{c}_\phi & -\mathrm{s}_\phi \\ \dfrac{w}{\sqrt{u^2+w^2}} & 0 & \dfrac{-u}{\sqrt{u^2+w^2}} \end{bmatrix} \\ \underline{f}_2 &\equiv \underline{J}^{-1}\left[\underline{M}_a^0 + \tilde{\underline{R}}_{ac}\underline{T}_{bw}\underline{F}_{a,w}^0 + \underline{M}_{prop} - \tilde{\underline{\Omega}} J \underline{\omega}\right] \\ \underline{G}_2 &\equiv \underline{J}^{-1}\left[\underline{M}_u + \tilde{\underline{R}}_{ac}\underline{T}_{bw}\underline{F}_u\right]\end{aligned} \tag{25}$$

where u, v and w are the body velocity vector components and where s_ϕ and c_ϕ is used to denote sine and cosine and where:

$$f_\beta = \frac{1}{\sqrt{u^2+w^2}}\left(f_x + f_y + f_z\right) \tag{26a}$$

$$f_x = -\frac{uv}{V_t^2}\left[a_x - g\mathrm{s}_\theta\right] \tag{26b}$$

$$f_y = \left[1 - \left(\frac{v}{V_t}\right)^2\right]\left[a_y + g\mathrm{s}_\phi \mathrm{c}_\theta\right] \tag{26c}$$

$$f_z = -\frac{vw}{V_t^2}\left[a_z + g\mathrm{c}_\phi \mathrm{c}_\theta\right] \tag{26d}$$

with V_t being the total velocity $\sqrt{u^2+v^2+w^2}$ and a_x, a_y and a_z the specific forces in the body frame.

3.2. NDI control laws

Based on the time-scale separation principle [8], the system can be controlled separately by an NDI law for the dynamics equation and another for the kinematics. Using standard NDI with command filtering, the rate control problem can then be solved by:

$$\underline{u}^0 = \underline{T}_u^{-1}\underline{J}\left[\underline{\nu}_2 - \underline{J}^{-1}\left[\underline{T}^0 - \tilde{\underline{\Omega}} J \underline{\omega}\right]\right] \tag{27a}$$

$$\underline{\nu}_2 = \underline{K}_2\left(\underline{\omega}_d - \underline{\omega} + \underline{\chi}_2\right) + \dot{\underline{\omega}}_d \tag{27b}$$

$$\dot{\underline{\chi}}_2 = -\underline{K}_2\underline{\chi}_2 + \underline{T}_u\underline{J}^{-1}\left(\underline{u} - \underline{u}^0\right) \tag{27c}$$

$$\underline{T}_u = \underline{M}_u + \tilde{\underline{R}}_{ac}\underline{T}_{bw}\underline{F}_u \tag{27d}$$

$$\underline{T}^0 = \underline{M}_a^0 + \tilde{\underline{R}}_{ac}\underline{T}_{bw}\underline{F}_{a,w}^0 + \underline{M}_{prop} \tag{27e}$$

where $\underline{K}_2$ is the proportional gain of the rate control loop (the inner loop), the feedforward term $\dot{\underline{\omega}}_d$ is provided by the outer loop and where the actual control signal $\underline{u}$ is obtained at the output of the CF. It is clear that model dependency results in a complex control law. Under the assumptions mentioned before, a simplification can be obtained with the employment of incremental dynamics:

$$\dot{\underline{\omega}} \approx \dot{\underline{\omega}}_0 + \underline{J}^{-1}\left[\underline{M}_u + \tilde{\underline{R}}_{ac}\underline{T}_{bw}\underline{F}_u\right] \tag{28}$$

which is simpler than Eq.(23), resulting in the following INDI control law:

$$\underline{u}^0 = \underline{u}_0 + \underline{T}_u^{-1}\underline{J}\left[\underline{K}_2\left(\underline{\omega} - \underline{\omega}_d + \underline{\chi}_2\right) + \dot{\underline{\omega}}_d - \dot{\underline{\omega}}_0\right] \tag{29}$$

For the (outer) control loop, incremental control is not considered since it is not expected to bring about any clear advantage. The command filtered NDI control law for the outer loop is then given by:

$$\underline{\omega}_d^0 = \underline{G}_1^{-1}\left[\underline{\nu}_1 - \underline{f}_1\right] \tag{30a}$$

$$\underline{\nu}_1 = \underline{K}_1\left(\underline{\varphi}_{\beta,d} - \underline{\varphi}_\beta + \underline{\chi}_1\right) + \dot{\underline{\varphi}}_{\beta,d} \tag{30b}$$

$$\dot{\underline{\chi}}_1 = -\underline{K}_1\underline{\chi}_1 + \underline{G}_1\left(\underline{\omega}_d - \underline{\omega}_d^0\right) \tag{30c}$$

where the feedforward term $\dot{\underline{\varphi}}_{\beta,d}$ is assumed to be provided externally.

3.3. BKS control laws

Since the dynamic system represented by Eq.(24) and Eq.(25) is in strict-feedback form, the BKS design procedure can be readily applied to solve the attitude control problem. The adaptation of the generic laws in Eq.(16) is straightforward and the command filtered BKS control laws for attitude control read:

$$\underline{\alpha}_1 = \underline{G}_1^{-1}\left[\underline{C}_1\left(\underline{\varphi}_{\beta,d} - \underline{\varphi}_\beta\right) + \dot{\underline{\varphi}}_{\beta,d} - \underline{f}_1\right] \tag{31a}$$

$$\dot{\underline{\chi}}_1 = -\underline{C}_1\underline{\chi}_1 + \underline{G}_1\left(\underline{\omega}_d - \underline{\alpha}_1 + \underline{\chi}_2\right) \tag{31b}$$

$$\underline{u}^0 = \underline{T}_u^{-1}\underline{J}\Big[\underline{C}_2\left(\underline{\omega}_d - \underline{\omega}\right) + \dot{\underline{\omega}}_d - \underline{J}^{-1}\left[\underline{T}^0 - \underline{\tilde{\Omega}}\underline{J}\underline{\omega}\right] + \underline{G}_1^\top\left(\underline{\varphi}_\beta - \underline{\varphi}_{\beta,d} - \underline{\chi}_1\right)\Big] \tag{31c}$$

$$\dot{\underline{\chi}}_2 = -\underline{C}_2\underline{\chi}_2 + \underline{T}_u\underline{J}^{-1}\left(\underline{u} - \underline{u}^0\right) \tag{31d}$$

where $\dot{\underline{\varphi}}_{\beta,d}$ is assumed to be provided externally, $\underline{\omega}_d$ and $\dot{\underline{\omega}}_d$ are the outputs of the CF of the outer loop and the final control input $\underline{u}$ is obtained by filtering the signal $\underline{u}^0$ with the CF of the inner loop. As for INDI, the expression for the latter can be simplified by considering the incremental dynamics:

$$\underline{u}^0 = \underline{T}_u^{-1}\underline{J}\Big[\underline{C}_2\left(\underline{\omega}_d - \underline{\omega}\right) + \dot{\underline{\omega}}_d - \dot{\underline{\omega}}_0 + \underline{G}_1^\top\left(\underline{\varphi}_\beta - \underline{\varphi}_{\beta,d} - \underline{\chi}_1\right)\Big] \tag{32}$$

4. MATLAB/Simulink® implementation

4.1. Aerosonde® model

The Aerosonde® is a small, inverted V-tail, single propeller, fixed-wing UAV. The Simulink® model used herein is based on the Aerosim Blockset library and includes linear aerodynamics based on component build-up and dimensionless equations, from which the terms in Eq.(22) can be readily obtained. In particular, the control derivatives used by the incremental controllers are given by:

$$\underline{M}_u = \bar{q}S\underline{C}_\delta \quad \text{and} \quad \underline{F}_u = \bar{q}S\left[\underline{C}_1 + \underline{C}_2\underline{S}_u\right] \tag{33}$$

where $\bar{q}$ is the dynamic pressure, S is the wing area and where the following definitions have been used:

$$C_\delta = \begin{bmatrix} bC_{l_{\delta_a}} & 0 & bC_{l_{\delta_r}} \\ 0 & \bar{c}C_{m_{\delta_e}} & 0 \\ bC_{n_{\delta_a}} & 0 & bC_{n_{\delta_r}} \end{bmatrix} \tag{34a}$$

$$C_1 = \begin{bmatrix} 0 & 0 & 0 \\ C_{Y_{\delta_a}} & 0 & C_{Y_{\delta_r}} \\ 0 & -C_{L_{\delta_e}} & 0 \end{bmatrix} \tag{34b}$$

$$C_2 = \begin{bmatrix} -C_{D_{\delta_a}} & -C_{D_{\delta_e}} & -C_{D_{\delta_r}} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{34c}$$

with b the wing span, $\bar{c}$ the mean aerodynamic chord and $\underline{S}_u$ is a diagonal matrix with the signs of the elements of $\underline{u}$ in its diagonal.

4.2. Preliminary results

Preliminary results are now presented to the reader in Figure 1 regarding an attitude tracking task with both INDI and IBKS controllers from the previous sections, using the simplification $\underline{T}_u = \underline{M}_u$, but without command filters. This simplification is valid, since it was verified that more than 95% of the control effectiveness is due to the $\underline{M}_u$ term alone at all times. The reference signals consist of 10 deg doublets with the duration of 3 seconds for pitch and roll and constant zero sideslip. A 100 Hz control frequency was used. The gains were designed the same for every channel, being $k_2 = 10$ for the inner loop and $k_1 = 2.5$ for the outer loop.

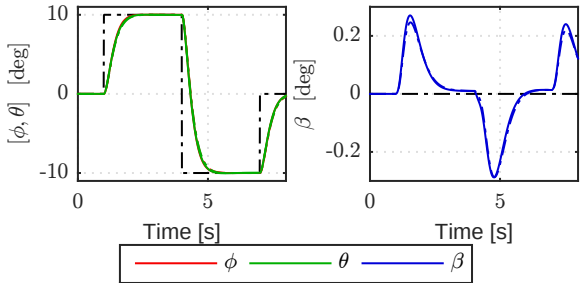


Figure 1: Attitude tracking results with INDI (solid lines) and IBKS (dash-dotted lines) controllers. The black lines represent signal references.

The results in Figure 1 are satisfactory as the references are tracked efficiently in the pitch and roll channels with minor sideslip peaks due to adverse yaw.

4.3. Introduction of actuator dynamics

After achieving successful attitude tracking in nominal conditions, the simulated system was augmented with an actuators model, incorporating a uniform time-delay of 30 ms, magnitude bounds $|\underline{u}| \in \pm[30\,,30\,,40]^\top$ deg, rate limits of 1 rad/s and a lowpass frequency response with a 13 rad/s cut-off frequency. In turn, this motivates the introduction of command filtering in the control framework. For both loops, second-order CFs were used [10], as they allow to impose both magnitude and rate limits. The design of the CF of the inner loop is driven by the the actuators modelling, whereas for the CF of the outer loop the bandwidth is imposed by the bandwidth of the controller in the inner loop. In order to illustrate the advantages of using the CFs, a magnitude limit of 0.7 deg was forced on the aileron deflection and a 5 deg step signal was used as reference for the roll angle, whilst keeping the pitch and sideslip at zero. The results are depicted in Figure 2 and one can notice that the CF prevents continued aileron saturation and exaggerated overshooting without compromising the settling time. The differences between the results for INDI and IBKS are justified by the fact that the two command filtered designs are slightly different and therefore equal gains are not expected to yield the same results.

4.4. Angular acceleration feedback

In reality, angular acceleration sensors, often costly, are not commonly integrated in the sensing systems. The majority of the literature on incremental attitude control has therefore tackled the issue

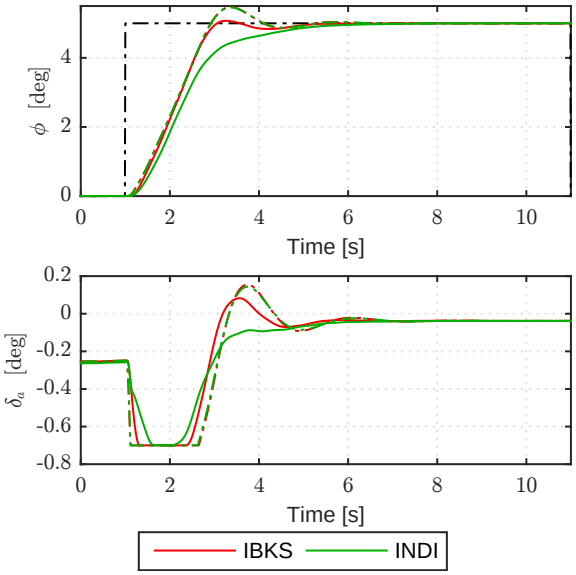


Figure 2: Effect of aileron saturation on roll tracking. The black lines represents the roll reference. The solid lines correspond to the case with the CF included in the simulation and the dashed lines without.

of obtaining $\underline{\dot{\omega}}_0$ by using estimation/filtering methods on IMU data. An intuitive solution is to use backward finite differences on the measured angular rates. This approach is relevant for this work since the end goal is practical implementation. In [7], the authors implemented a predictive filter that is designed by offline parameter estimation based on ordinary least squares optimization. Since the controllers developed in this chapter are intended for experimental implementation, this approach is not adequate, as the filter design should not depend on the availability of a good simulation model. Another option is the use of a first-order highpass (HP) filter. By inspection of its generic transfer function:

$$H_{HP}(s) = \frac{a_1 s}{1 + \tau s} \tag{35}$$

with $a_1 \in \mathbb{R}$ and $\tau \in \mathbb{R}$, one may notice that for low frequencies, $s \ll 1/\tau$, the denominator is dominated by the constant term and the highpass filter becomes a low-frequency derivator. Furthermore, the validity of the approximation increases with a decreasing value for the time constant of the filter, τ. However, decreasing τ results in an increase of the high frequency gain, given by a_1/τ. This fact might be undesirable if the input signal presents high-frequency noise. For that reason, a second order bandpass (BP) filter is also proposed. The transfer function of such a filter can be represented by:

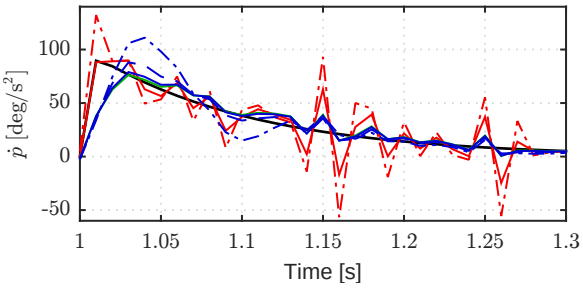
$$H_{BP}(s) = \frac{a_1 s}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{36}$$

where ω_n is the natural frequency and ζ is the damping factor. At low frequencies, the filter approximates a derivator:

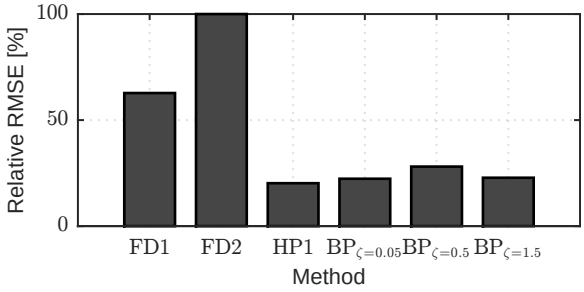
$$H_{BP} \approx \frac{a_1}{\omega_n^2} s \tag{37}$$

if a_1 is designed as $a_1 = \omega_n^2$.

Results were obtained for the angular acceleration vector estimate through numerical differentiation of a noisy angular rate signal. The sensor model was obtained from the datasheet of a commercial MEMS-based sensor with an integrated IMU and GNSS receiver. First and second order finite differences, FD1 and FD2, respectively, were implemented with a uniform sampling time of 0.01 seconds. The first-order highpass filter, HP1, and the second-order bandpass filter, BP2, were discretized at 100 Hz using the same finite differences, with the BP filter requiring, additionally, the second-order FD for the second-order time-derivative. A 10 Hz cut-off frequency was used for both filters. The results for the roll channel for a short transient of time are depicted in Figure 3. The results for the BP2 filter include three different values of ζ to better illustrate its effect on performance. In Figure 3(b), a bar plot of the root-mean-square error (RMSE) of each method is presented to aid in comparing performance.



(a) Roll acceleration. The solid and dash-dotted red lines concern the FD1 and FD2 approaches. The blue lines regard the BP2 filter - the solid line was used for $\zeta = 1.5$, the dashed line for $\zeta = 0.5$ and the dash-dotted line for $\zeta = 0.05$. The HP1 filter is represented by the green line.



(b) Relative RMSEs for the different methods with respect to the maximum of the set

Figure 3: Roll acceleration results using finite differences and linear filters.

From Figure 3(a) it is possible to conclude that the second order FD is more sensitive to noise than first order, which is confirmed by a 40% performance decrease in terms of RMSE between the two FDs. The filters clearly produced better results. Regarding the BP2 filter in specific, inspection of Figure 3 shows how decreasing ζ implies more overshoot (higher passband peak gain). The HP1 filter presents very similar behaviour when compared to the BP2 filter for $\zeta = 1.5$. This could be expected since the width of the passband grows with ζ. However, if very high-frequency noise plays a more relevant role, the benefits of the BP2 filter become more clear. In sum, the fact that finite differences are so sensitive to sensor noise renders them unfit for implementation. The linear filters present roughly the same performance and, although the HP1 filter is a simpler design, the BP2 filter is expected to offer additional high-frequency advantages in a practical implementation.

Recall that the core principle behind incremental control is that the desired angular acceleration is obtained one time step ahead. If delay exists in the measurement, the incremental step assumption is violated and degradation deteriorates. Since the filter does indeed induce a time delay, this problem must be taken into consideration. One intuitive solution is to achieve synchronization between $\dot{\underline{\omega}}_0$ and $\underline{\omega}$, as seen from the controller, by using a lowpass filter on the angular rates measurement with the same characteristic polynomial as the BP2 filter. This way, the virtual control signal and the angular acceleration vector that are available at any given time for the computations of the inner loop correspond to the same (delayed) time step and therefore the controller does not overshoot/undershoot its output signal.

The sampling frequency was verified to have a minimum acceptable value of 15 Hz. The design for the bandpass and the lowpass filter was then set to a damping factor $\zeta = 2$, a cut-off frequency of 10 Hz and a sampling frequency of 20 Hz.

4.5. Calibration

In this section, sensor noise is added to the simulation. In order to justify the choice of the controller architecture to promote for practical implementation, four different command filtered, incremental controllers are briefly compared:

- The PI-INDI controller that consists of standard NDI in the outer loop and INDI in the inner loop, both robustified by integral action;
- The R-INDI controller that consists of standard NDI in the outer loop and INDI in the inner loop, both robustified by discontinuous control, as in [12];
- The PI-IBKS controller that consists of the IBKS controller with integral action in both loops, as shown in [10];
- The R-IBKS controller that consists of the IBKS controller with discontinuous terms in both loops.

Since the different controllers will compete for performance, it is adequate for the tuning of their parameters to be performed in the same systematic way. In short, the method employed consists in setting intervals for the gains of each controller and then recording some performance variables for each simulation run with a different combination of parameter values inside those intervals. In order to simplify the method, each controller was assumed to have four parameters only - one proportional gain and one robust control gain for each loop. For simplicity, the gains of the inner and outer loops were tested separately. For each controller, after obtaining data from every simulation, an n_v-by-n_s performance matrix D is built, with n_v being the number of performance variables and n_s the number of simulations. Then, a cost function is quantified, for each channel ($c \in [\phi, \theta, \beta]$), and for each pair s of gains, as:

$$J_s^c = \frac{1}{\|\underline{w}\|_1} \sum_{i=1}^{n_v} \underline{w}_i \frac{D_{is}}{\overline{D}_i} \qquad (38)$$

$$\overline{D}_i = \max_{s=1,\cdots,n_s} D_{is}$$

with D_{is} denoting the element (i, s) of D, $\overline{D}_i$ being the maximum element in each row of D and $\underline{w}$ being a vector weighing each performance variable. The optimal set of gains is found, for each loop, by finding the minimum of the unified performance index:

$$J^* = \min_{s=1,\cdots,n_s} J_s = \min_{s=1,\cdots,n_s} J_s^\phi + J_s^\theta + J_s^\beta \qquad (39)$$

It can be argued that the actual optimal gains can be different for each channel. For that reason, the performance index is also minimized separately for each control channel and two additional simulations are performed to verify if using the different gains in each channel results in performance increase.

After defining the optimal gains for each loop, the search intervals are refined and so are the fixed values that are used for one loop as the other is being tested. The procedure is repeated until the performance obtained with the extreme values of each interval becomes nearly identical.

The performance variables used in all channels include settling time, overshoot, tracking error RMS, control vector RMS (control effort) and control moving standard deviation (MSTD).

4.6. Position Control

Since the selected controller will be integrated in the open-source autopilot software Ardupilot© , it is adequate to choose the controller that seems to

respond better to the position controllers of that framework. Therefore, the L_1 and TECS controllers were implemented in MATLAB/Simulink® and introduced in the simulation, as shown in Figure 4. The former is responsible for lateral navigation and the latter by coupled height and speed control. The manoeuvre selected for this comparison consists of a lateral/directional "zig-zag" with two alternating reference altitudes at a constant speed of 20 m/s. It was designed to excite the couplings between lateral and longitudinal motions. The results obtained with the four incremental controllers are depicted in Figure 5. Results obtained with the baseline linear controller from Ardupilot© are also included to provide a comparison point. It is noticeable in Figure 5 that the altitude tracking of the TECS controller does not present satisfactory results, and the lateral position controller presents steady-state cross-track error. This can be explained by the fact that the two position controllers are decoupled and therefore they are both inefficient when aggressive manoeuvres are being simultaneously executed in the longitudinal and lateral planes. Nevertheless, it can be concluded that the four incremental controllers appear to have the same performance. Moreover, it can be observed that the nonlinear control solutions are more efficient in dealing with the motion couplings - the linear controller clearly loses more altitude when banking after completion of waypoints 3 and 5. In order to perform also a brief quantitative analysis, the attitude tracking performance of each controller is evaluated based on the tracking error RMS in Table 1, where it is evident that the nonlinear, incremental controllers yield roughly the same results. It is arguable that the small performance differences in attitude tracking error RMS could be mitigated with a more optimal scheme to tune the controllers. It is possible to conclude that the main advantage of the nonlinear control techniques over the baseline linear control scheme proved to be a combination of both performance increase and also the ease of performing fine tuning. In sum, only at the expense of requiring a model for the control dependent dynamics, the incremental controllers showed to be a simpler solution for attitude control with better closed-loop performance.

Regarding the choice of the nonlinear controller to be promoted for practical implementation, the PI-INDI controller was selected. The reason behind this choice is related to the fact that NDI is slightly conceptually simpler than Backstepping and provides equally satisfying results. Furthermore, it is the nonlinear approach which is most inline with the architecture of the baseline linear controller, which suggests that its implementation in the Ardupilot© software should be taken as a first step before engaging on more elaborate control approaches.

Table 1: Waypoint navigation performance metrics.

Controller	**Tracking RMSE [deg]**		
	ϕ	θ	β
PI-INDI	0.27	0.23	0.37×10^{-4}
R-INDI	0.10	0.14	3.38×10^{-4}
PI-IBKS	0.28	0.20	1.02×10^{-4}
R-IBKS	0.28	0.22	9.38×10^{-4}
Linear	0.37	0.48	21.48×10^{-4}

4.7. Robustness

Some robustness tests were performed on the PI-INDI controller before engaging on its practical implementation. These allowed concluding that the controller is relatively insensitive to uncertainties in the control derivatives up to 100%, with the pitch response being more susceptible to degradation; it also showed to be robust against inertia tensor mismatches as performance was not seriously compromised for deviations of ±50%. Given the design assumptions of the incremental controller, its sensitivity to actuator time-delay was studied and it was verified to achieve satisfactory results with delays of up to 150 ms, which is admissible for practical integration. It was also confirmed that the incremental controller is dependant on high sampling frequency, with similar results for 50 Hz and 100 Hz. The directional channel showed to be the most sensitive to low control frequencies (< 40 Hz). Finally, robustness against external disturbances was analysed with the introduction of simulated wind, leading to the conclusion that it has a serious effect on performance with wind speeds of up to 10 m/s.

5. Ardupilot© integration

This final section concerns the pre-experimental implementation and testing of the nonlinear control algorithm in the Ardupilot© framework. Fortunately, the software includes a native software-in-the-loop (SITL) simulator that allows testing the code under development without the need for any additional hardware and it can interface with a number of different vehicle simulators to obtain sensor data, either the ones built in the Ardupilot© code or several external alternatives. In this project, the JSBSIM simulator was used. As for the fixed-wing model, the already available model for the Rascal 110 fixed-wing aircraft was used, in order to take advantage of the fact that the control derivatives used by JSBSIM can be easily extracted from the underlying code, which in turn allows designing the INDI controller for nominal conditions.

The discretization of the bandpass filter (from which the angular accelerations are obtained), the lowpass filter (used to delay the angular rates sampling) and the command filters must take into account the fact that, in a real-time system, the sam-

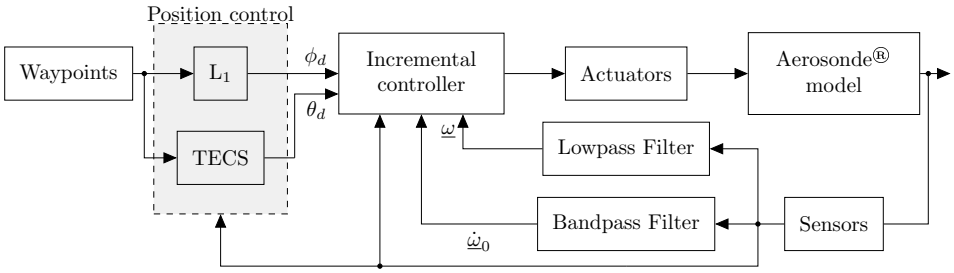


Figure 4: Block diagram of the final Simulink® control architecture.

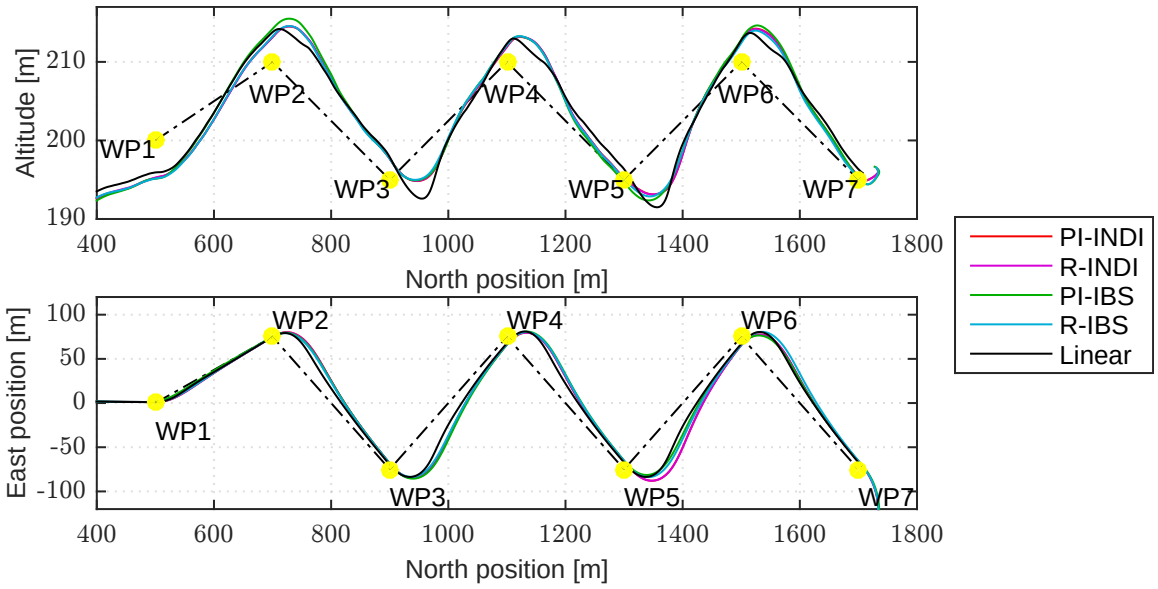


Figure 5: Position control results for controller comparison.

pling time may vary with time. Therefore, its implementation in Ardupilot© was changed from the one in MATLAB/Simulink® by considering finite differences with non-uniform grid spacing.

The Ardupilot© code includes sideslip compensation by means of lateral acceleration feedback. However, for many small UAVs, this setup may be inefficient if the aircraft does not have sufficient fuselage side area to produce a measurable lateral acceleration. For that reason, active sideslip control is optional in Ardupilot© control and the rudder can then be used to simply maintain coordinated flight which, in theory, occurs at zero sideslip as well. To retain this design (and operational) flexibility, the INDI controller was adapted in a way such that when active control of the sideslip angle is not desired, the outer loop performs dynamics inversion for the roll and pitch rates only, whilst the yaw rate is fed directly to the inner loop to keep the flight coordinated. The main results obtained with JSBSIM are depicted in Figure 6 and correspond to the position control of the fixed-wing aircraft using both the PI-INDI and baseline linear attitude controllers, without active sideslip regulation. The 8-shaped manoeuvre was selected because it allows studying the response of the aircraft to aggressive manoeuvring, its ability to keep a coordinated flight and also its response to a roll reference doublet. Inspection of Figure 6 suggests that the controllers have very similar performance and the lateral position tracking appears to be achieved efficiently. A deeper look into the quantitative results of Table 2 allows concluding that the attitude tracking presents roughly the same quality in both cases, with the most significant difference regarding the sideslip regulation task. This suggests, as expected, that the nonlinear controller is more efficient at keeping the flight coordinated. Additionally, the altitude RMS is marginally smaller with INDI, as the coupled nonlinear controller is more efficient at maintaining pitch when banking, as seen also in the MATLAB/Simulink® simulations.

6. Conclusions

This work addressed the practical applicability of incremental control to the attitude control problem of fixed-wing aircraft. The first results were obtained from a MATLAB/Simulink® implementation with a nonlinear model for the Aerosonde® UAV, for which both IBKS and INDI approaches

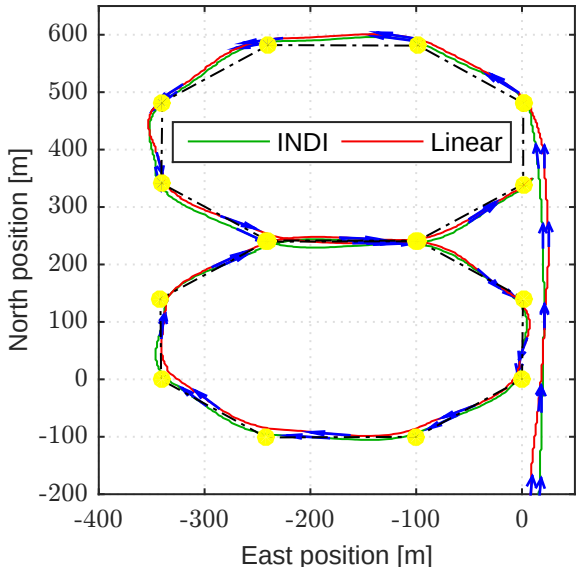


Figure 6: JSBSIM simulation results without active sideslip control.

Table 2: Performance metrics for INDI and linear controllers in the JSBSIM simulation.

Controller	**Altitude tracking RMS [m]**	**Tracking RMS [deg]**		
		ϕ	θ	β
INDI	1.461	4.98	0.84	7.94
Linear	1.726	4.85	1.11	11.9

yielded satisfactory results when command filters were integrated in each framework and the actuators dynamics were included in the simulation. The suitability of the bandpass filter to obtain the angular acceleration measurement was verified. Then, two INDI and two IBKS controllers, with either integral or discontinuous control in both loops, were considered for practical implementation. A simple optimization method was developed to tune the controllers under a common criterion. The position and attitude controllers from Ardupilot© were integrated in the simulation environment in order to assess which nonlinear technique maximizes the performance of the integrated design and the results endorsed the expected conclusion that performance can be increased with the adoption of nonlinear control, which presents more intuitive tuning, at the expense of knowledge on the control derivatives. The PI-INDI controller was then selected for practical implementation. Robustness tests were performed to appraise some limitations of the chosen controller, validating important robustness properties for practical implementation, including the requirements imposed by incremental control design. Simulation results using the JSBSIM simulator and the Ardupilot© implementation of the PI-INDI controller, without active sideslip control, were very satisfactory as the controller showed to achieve better performance than its linear counterpart at keeping the flight coordinated and maintaining altitude. Unfortunately, due to calendar incompatibilities, the experimental flight tests could not be performed before the submission deadline of this project and are left out as future work.

References

[1] P. Acquatella, W. Falkena, E. van Kampen, and Q. P. Chu. Robust nonlinear spacecraft attitude control using incremental nonlinear dynamic inversion. In *AIAA Guidance, Navigation, and Control Conference*, 2012.

[2] P. J. Acquatella. Robust nonlinear spacecraft attitude control. Master's thesis, Technische Universiteit Delft, November 2011.

[3] P. J. Acquatella, E. van Kampen, and Q. P. Chu. Incremental backstepping for robust nonlinear flight control. In *EuroGNC 2013, 2nd CEAS Specialist Conference on Guidance, Navigation & Control*, April 2013.

[4] B. Etkin and L. D. Reid. *Dynamics of flight: stability and control*. Wiley New York, 1996.

[5] J. A. Farrell, M. Polycarpou, M. Sharma, and W. Dong. Command filtered backstepping. *IEEE Transactions on Automatic Control*, 2009.

[6] H. K. Khalil. *Nonlinear Control*. Pearson, global edition, 2015.

[7] S. Sieberling, Q. P. Chu, and J. A. Mulder. Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction. *Journal of Guidance, Control, and Dynamics*, 33(6), November 2010.

[8] P. Simplício. Helicopter nonlinear flight control. Master's thesis, Instituto Superior Técnico, October 2011.

[9] P. Smith. A simplified approach to nonlinear dynamic inversion based flight control. In *23rd Atmospheric Flight Mechanics Conference*, 1998.

[10] L. Sonneveldt. *Adaptive Backstepping Flight Control for Modern Fighter Aircraft*. PhD thesis, Technische Universiteit Delft, July 2010.

[11] P. van Gils, E. van Kampen, C. C. de Visser, and Q. P. Chu. Adaptive incremental backstepping flight control for a high-performance aircraft with uncertainties. In *AIAA Guidance, Navigation, and Control Conference*, 2016.

[12] I. Yang, D. Lee, and D. S. Han. Designing a robust nonlinear dynamic inversion controller for spacecraft formation flying. *Mathematical Problems in Engineering*, 2014.