A Nonlinear Control Method for Autonomous Navigation Guidance

G. Carloni¹, K. Bousson²

Abstract – This paper deals with a nonlinear feedback control method based on Lie derivatives for the navigation guidance of unmanned aerial vehicles (UAVs). The paper proposes a modelling concept for planar and three-dimensional navigation together with proven algorithms to cope with the autonomous navigation along waypoints and on loiters. The steps that were performed in order to validate the autonomous navigation method are clearly described. The method was successfully validated on various realistic navigation scenarios for unmanned aircraft.

Keywords: Navigation Guidance, UAV, Nonlinear Control, Lie Derivatives, Waypoints, Loiters

Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>Two Dimensions</td>
</tr>
<tr>
<td>3D</td>
<td>Three Dimensions</td>
</tr>
<tr>
<td>α</td>
<td>Stepsize</td>
</tr>
<tr>
<td>e</td>
<td>Error</td>
</tr>
<tr>
<td>g</td>
<td>Gravity Acceleration</td>
</tr>
<tr>
<td>γ</td>
<td>Flight Path Angle</td>
</tr>
<tr>
<td>h</td>
<td>Time-step</td>
</tr>
<tr>
<td>h_u</td>
<td>Control Gain</td>
</tr>
<tr>
<td>J</td>
<td>Performance Index</td>
</tr>
<tr>
<td>L(•)</td>
<td>Lie Derivative with respect to a variable (•)</td>
</tr>
<tr>
<td>q</td>
<td>Reference Trajectory</td>
</tr>
<tr>
<td>Q, Q₁, Q₂</td>
<td>State Weight Matrices</td>
</tr>
<tr>
<td>R, R₁, R₂</td>
<td>Control Weight Matrices</td>
</tr>
<tr>
<td>STANAG</td>
<td>NATO Standardization Agreement</td>
</tr>
<tr>
<td>σ</td>
<td>Bank Angle</td>
</tr>
<tr>
<td>u</td>
<td>System Control Vector</td>
</tr>
<tr>
<td>UAV</td>
<td>Unmanned Aerial Vehicle</td>
</tr>
<tr>
<td>v</td>
<td>Aircraft Speed</td>
</tr>
<tr>
<td>x</td>
<td>System State Vector</td>
</tr>
<tr>
<td>x</td>
<td>Cartesian Coordinate (easily understandable by the context)</td>
</tr>
<tr>
<td>y</td>
<td>Cartesian Coordinate</td>
</tr>
<tr>
<td>ψ</td>
<td>Heading Angle</td>
</tr>
<tr>
<td>z</td>
<td>Altitude (Vertical Cartesian Coordinate)</td>
</tr>
</tbody>
</table>

I. Introduction

The use of Unmanned Aerial Vehicles (UAVs) for military as well as civilian applications has increased significantly during the last decade. Recent progresses in guidance technologies based on global positioning systems are being exploited to enable UAVs to execute mission tasks without any human interface. Various methods to design nonlinear controllers have been developed up-to-date.
loiter navigation at constant flight altitude, it is possible to restrict the dynamic model to two dimensions, since loiter generally occurs at constant altitude.

Current methods dealing with navigation guidance are either based on the proportional navigation concept that is borrowed from missile guidance principles [3], [4], [5], or on path planning method [6] inspired by robot navigation. These methods are either well-suited for planar navigation, or requires adjustment of independent parameters in the case of three-dimensional navigation. Besides, they may require some amount of time for signal processing (in the case of proportional navigation) that some low cost UAVs cannot afford. Furthermore, the generated navigation trajectory may undergo overshoot in case of relatively stiff turn at waypoints.

The present paper rather resorts to a systematic approach to generate navigation guidance trajectories based on a nonlinear control method elaborated by Lu and Khan [7],[8] for a reference continuous system. That method consists in applying Lie derivatives to perform the local approximation of the system model according to its relative degree, then the control is computed as the minimizer of a quadratic cost function expressing a reference trajectory tracking error. In this approach, the time-step related to the time sampling is kept constant. The contribution of our work is twofold: the first is to extend the method described in [7] and [8] to be able to deal with adaptive time-steps so that the local error of the reference trajectory tracking be effectively minimized on-line, and the second is to validate the extended method on waypoint and loiter navigation for UAV missions.

The paper is organized as follows: section 2 describes the nonlinear control method that will be used in further sections, it first summarizes the method developed by Lu and Khan [7],[8], then proposes an extension to it; section 3 deals with the planar navigation; section 4 copes with the three-dimensional navigation; section 5 discusses about the results of the simulations, and section 6 concludes the paper.

II. Nonlinear Control

We first summarize in this section the aforementioned nonlinear predictive control method devised by Lu and Khan [7],[8], then we propose an extension to it so that it may account for adaptive time-step.

The method assumes that the model of the system is affine in the control, that is as in (1):

\[
\dot{x} = f(x) + G(x)u \tag{1}
\]

However, this equation is assumed to be on the following form where the state vector is split into two subvectors \(x_1\) and \(x_2\):

\[
\begin{align*}
\dot{x}_1 &= f_1(x) \\
\dot{x}_2 &= f_2(x) + B_2(x)u
\end{align*}
\tag{2}
\]

with:

\[
x = [x_1 \ x_2]^T
\]

and consequently:

\[
G = \begin{bmatrix}
0 \\
B_2
\end{bmatrix}
\]

\(B_2\) and \(G\) being matrices of appropriate dimensions.

The first state vector \(x_1\) in equation (2) is not directly dependent on the control vector. The second state vector \(x_2\), instead, is composed of that set of differential equations that involves the control vector.

II.1. Lu and Khan’s Nonlinear Control Method

Consider the equation (2) above. Then, the relative degree \(r_i\) of state vector \(x_i(t), (i = 1 \text{ or } 2)\), is defined in [2] as the minimum number of times one has to differentiate \(x_i(t)\) in order to have the control vector \(u\) appearing explicitly in the expression of \(x_i(t)\). Let \(r_i\) be the relative degree of state vector \(x_i(t), \text{ for } i = 1 \text{ or } 2\). If \(h\) is the time-step, then, it is possible to approximate the state at time \(t + h\) as:

\[
x_1(t + h) \approx x_1(t) + h h f_1(x) + \frac{h^2}{2!} \frac{\partial f_1}{\partial x} (x) f(x) + \ldots + \frac{h^{r_1-1}}{r_1!} \frac{\partial^{r_1-1} f_1}{\partial x^{r_1-1}} (x) f(x) \tag{3}
\]

\[
x_2(t + h) \approx x_2(t) + h h f_2(x) + \frac{h^2}{2!} \frac{\partial f_2}{\partial x} (x) f(x) + \ldots + \frac{h^{r_2-1}}{r_2!} \frac{\partial^{r_2-1} f_2}{\partial x^{r_2-1}} (x) f(x) + \Lambda(h) W(x(t)) u(t) \tag{4}
\]

where \(\Lambda \in \mathbb{R}^{n \times n}\) is a diagonal matrix (\(n\) being the dimension of the state space) with elements \(\lambda_{ii}\) on the main diagonal given by:

\[
\lambda_{ii}(h) = \frac{h^{r_i}}{r_i!}, \quad i = 1, \ldots, n \tag{5}
\]

and \(W \in \mathbb{R}^{n \times m}\) is a matrix whose row \(i\) is defined as:

\[
w_i = \left[ L_{i,1} \left[ L_{1,1}^{-1}(x_1) \right] \ldots L_{i,n} \left[ L_{n,1}^{-1}(x_1) \right] \right], i = 1, \ldots, n \tag{6}
\]
where vector functions \( g_1, \ldots, g_m \) are the columns of matrix \( G \), and the Lie derivative with respect to each \( g_j \) is defined as:

\[
L_{g_j} \left[ L_{g_j}^{-1} (x_j) \right] = \frac{\partial L_{g_j}^{-1} (x_j)}{\partial x} g_j
\]

Assuming the reference trajectory as a continuous time-function \( q(t) = [q_1(t) \; q_2(t)]^T \), it is possible to define the related dynamic system as

\[
\begin{align*}
q_1 &= f_1(q) \\
q_2 &= f_2(q) + B_2(q) u_{\text{ref}}
\end{align*}
\]

where \( u_{\text{ref}} \) is the control reference.

The reference function (representing, for instance, a nominal trajectory) can be recursively approximated as:

\[
q(t + h) \approx q(t) + \frac{h \dot{q}(t) + \frac{h^2}{2!} \ddot{q}(t) + \ldots + \frac{h^n}{n!} q^{(n)}(t)}{(9)}
\]

The error between the actual state of the system and the reference is:

\[
e(t) = x(t) - q(t)
\]

The performance index to be considered is defined as:

\[
J(u(t)) = \frac{1}{2} e^T(t) Q e(t) + \frac{1}{2} u^T(t) R u(t)
\]

where \( Q \in \mathbb{R}^{m \times m} \) and \( R \in \mathbb{R}^{n \times n} \) are at least positive semi-definite and defined by the designer. This performance index expresses the tracking error and the control effort.

In particular \( Q \) and \( R \) will be defined as:

\[
Q = \begin{bmatrix} Q_1 & 0 \\ 0 & hQ_2 \end{bmatrix}, \quad R = \begin{bmatrix} 0 & 0 \\ 0 & R_2 \end{bmatrix}
\]

The optimal predictive control for the minimized performance index is found by solving the equation \( \partial J / \partial u = 0 \) for the control \( u \). This gives:

\[
u(t) = - \left[ \left[ \Lambda(h) W(x) \right]^T Q \Lambda(h) W(x) + R \right]^{-1} \left[ \left[ \Lambda(h) W(x) \right]^T Q e(t) + e(t) \right]
\]

Hence, for the system previously described, the expression of the optimizing control is given as:

\[
u(t) = P^{-1} \left\{ \frac{1}{2h^2} \left( \frac{\partial f_1(x)}{\partial x_1} B_2 \right)^T Q_1 \left( e_1 + h \dot{e}_1 + \frac{h^2}{2!} \ddot{e}_1 \right) + \frac{1}{h} B_2^T Q_2 (e_2 + h \dot{e}_2) \right\}
\]

where \( e_1 = x_1 - q_1, e_2 = x_2 - q_2 \), and:

\[
P = \frac{1}{4} \left( \frac{\partial f_1(x)}{\partial x_2} B_2 \right)^T Q_2 \left( \frac{\partial f_1(x)}{\partial x_2} B_2 \right) + B_2^T Q_2 B_2 + h^{-2} R_2
\]

II.2. Dealing with Adaptive Time-Step

The model exposed above may either be used with a constant value of the time-step \( h \) (which, as proved in [7], may be seen as the controller gain), or with an optimization procedure that computes a value of the time-step that minimizes the cost function in (11) assuming the control to be given by equation (14). On one hand the performance of the controller when a constant time-step is used may be very low, on the other hand, search for a minimizing time-step through online optimization procedure is cumbersome in the framework of actual control applications. In order to increase the performance of the control without jeopardizing the computing time and resource of the controller, we propose an adaptive method to update the time-step. Let call \( h_u \) the time-step computed assuming the current control to be \( u \). Instead of searching for a minimizing time-step, we search for a time-step that may decrease the current value of the performance index using a gradient descent rule:

\[
h_u (t + h) = h_u (t) - \alpha \left( \frac{\partial J}{\partial h} \right)_{h=h_u}
\]

where \( h \) is the generic variable representing the time-step of the system and \( h_u \) is the actual time-step associated to the current control. The step-size \( \alpha \) is defined by the designer and should be small enough to allow a reduction in the value of the performance index in the opposite direction of the gradient.

In the analyzed model, it gives rise to a system based on the following equations:

\[
\begin{align*}
x_1(t + h) &\approx x_1(t) + h f_1(x) + \frac{h^2}{2!} \frac{\partial f_1(x)}{\partial x} f(x) + \ldots + \frac{h^{n-1}}{n!} \frac{\partial^{n-1} f_1(x)}{\partial x^{n-1}} f(x) \\
&\quad + h e_1 + \frac{h^2}{2!} \ddot{e}_1 f(x) + \ldots + \frac{h^{n-1}}{n!} \frac{\partial^{n-1} e_1}{\partial x^{n-1}} f(x)
\end{align*}
\]
where:

\[ u(t) = \begin{cases} \frac{1}{2h_u^2} \left( \frac{\partial f_1(x)}{\partial x} B_2 \right)^T Q_1 \begin{pmatrix} e_1 + h_u \dot{e}_1 + \frac{h_u^2}{2} \dot{e}_1 \\ e_2 + h_u \dot{e}_2 \end{pmatrix} + P = \frac{1}{4} \left( \frac{\partial f_1(x)}{\partial x} B_2 \right)^T Q_1 \left( \frac{\partial f_1(x)}{\partial x} B_2 \right) + B_t^T Q_2 B_t + h_u^{-1} R_2 \end{cases} \]

The effectiveness of this method will be shown in the following chapters.

**III. Development for Planar Navigation**

**III.1. Dynamic Model**

The dynamic model used in this chapter is a 2D model suitable designing the controller. To be coherent with the development performed by Lu, the simulation describes a subsonic aircraft at constant speed \( V = 150 \text{ m/s} \). The equations of motion are:

\[ \dot{x} = V \cos \psi \]
\[ \dot{y} = V \sin \psi \]
\[ \dot{\psi} = \frac{g}{V} \tan \sigma \]

where \( x \) and \( y \) are position coordinates, \( \psi \) is the heading angle and \( \sigma \) is the bank angle. This bank angle is subject to the constraint:

\[ |\sigma| \leq 80^\circ \]

**III.2. Loiter Navigation**

The first application of the controller is applied to a continuous system. The following application is similar to the one presented by Lu [7]. It was performed with the purpose of validating the implementation of the mathematical model reported above, and with a constant time-step and with a dynamic time-step.

In this section the dynamic time-step controller is named Controller I and the constant time-step controller is named Controller II.

The reference trajectory is a circle of radius \( R \) defined by the equations:

\[ \dot{x}(t) = R \cos \omega t \]
\[ \dot{y}(t) = R \sin \omega t \]
\[ \dot{\psi}(t) = \frac{\pi}{2} + \omega t \]

where \( R = 1000 \text{ m} \) and \( \omega = \frac{V}{R} = 0.15 \text{ rad/s} \).

The initial state of the aircraft is:

\[ x(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \]

and the control defined as:

\[ u = \tan \sigma \]

The weight matrices are set as:

\[ Q_1 = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}, \quad Q_2 = 0.01h^2, \quad R_2 = 1 \]

Fig. 1 shows the tracking response of the system for \( h=0.5s \). It demonstrates how the control dynamic time-step increases considerably performances of the controller in a dynamic tracking problem. Moreover, the following picture (Fig. 2) illustrates the tracking error of the system. Despite its large initial error, the trajectory reaches the reference easily.

As previously it is possible to see the different response of the two controllers. The “dynamic” controller reaches the reference faster and without any oscillation. Instead, the error in case of constant time-
step is characterized by a damping that yields a negligible value only after two oscillations.

Fig. 2. 2D Loiter – Tracking Error

The control histories of the two controllers are compared in Fig. 3. Whereas the control variable $u_{II}$ presents a smooth trend, $u_I$ described two valleys that represent the correction necessary to ensure the desired status.

Fig. 3. 2D Loiter – Control History

Finally, the history of $h_u$ is reported in Fig. 4. It smoothly approaches a value that will increase performances of the controller. Note that the step-size $\alpha$ is defined as:

$$\alpha(i) = \frac{0.01}{i}, \quad \text{for } i = 1, \ldots, n$$  \hspace{1cm} (31)

$$\alpha = 0.0001, \quad \text{for } i > 100$$  \hspace{1cm} (32)

where $n$ is the number of the steps required in the first 20 seconds, the value, which $h_u$ will approach too, will not be the optimum control gain but it will simply give rise to an improvement of the controller performances.

Fig. 4. 2D Loiter – Dynamic Control Gain Trend


The same model previously exposed has been implemented for a trajectory defined by waypoints. In this case the reference trajectory cannot be considered time-varying and, hence, all the considerations about the dynamic time-step will fall. Even if the implementation will remain the same as the previous one, the resulting control gain will be constant.

Waypoints are sets of coordinates that identify a point in physical space. For the purpose of navigation, these coordinates usually include longitude and latitude, and sometimes altitude (particularly for air navigation). Waypoints have only become widespread for navigation since the development of advanced navigational systems, such as the Global Positioning System (GPS) and certain other types of radio navigation. Waypoints located on the surface of the Earth are usually defined in two dimensions (e.g., longitude and latitude); those used in the Earth's atmosphere or in outer space are defined in at least three dimensions (four if time is one of the coordinate axis). These waypoints are used to help define invisible routing paths for navigation. A waypoint can be a destination, a fix along a planned course used to make a journey, or simply a point of reference useful for navigation.

In the following example the waypoint has been set as:

$$x_{ref} = 500\text{m}$$

$$y_{ref} = 1000\text{m}$$

and the simulation will stop when

$$e \leq 50\text{m}$$

where $e$ is the line-of-sight distance between the current position of the aircraft and the target waypoint, that is, $e$ is computed as:

$$e = \sqrt{(x_{ref} - x)^2 + (y_{ref} - y)^2}$$

Once the waypoint is defined, the system will respond as illustrated in Fig. 5.

Fig. 5. Waypoint Tracking
It is possible to see how the choice of an appropriate time-step $h$ is extremely important for the minimization of the final error and for a smooth response. The control history is illustrated in Fig. 6. It is easy to see how the system is better controlled for a time-step $h=0.1s$.

![Control History](image)

*Fig. 6. Waypoint Tracking – Control History*

The following figure shows an example of navigation by a three-waypoint set with excellent results.

![Navigation by Waypoint](image)

*Fig. 7. Planar Navigation by Waypoint*

The following picture shows how the navigation is perfectly controlled.

![Control History](image)

*Fig. 8. Planar Navigation by Waypoint – Control History*

### III.4. Waypoint Navigation with Variable Range

In order to lead the aircraft to the next waypoint, the model previously exposed has been provided with a variable waypoint range. To prevent overshoot when the aircraft has to switch from a waypoint to another, it is necessary that the switching operation occurs from a certain distance to the current waypoint. The determination of that distance is based on the bearing of the current waypoint, as will be described hereafter.

First of all the minimum radius of bank $R_{\text{min}}$ is calculated in accordance with the constraint of $|\sigma| \leq \bar{\sigma} = 80^\circ$ (equation (24)), that is:

$$R_{\text{min}} = \frac{V^2}{g \tan \bar{\sigma}}$$

(36)

Then, based on the waypoint set, we can determine the course from the current waypoint $A$ to the next $B$ from their coordinates as:

$$\psi_{AB} = \tan^{-1}\left(\frac{y_B - y_A}{x_B - x_A}\right)$$

(37)

Hence, the variation of the course (and therefore, of heading) will be

$$\Delta \psi = |\psi_{BC} - \psi_{AB}|$$

(38)

Finally the distance from which the switching to the next waypoint should occur is computed as:

$$R_{\text{din}} = R_{\text{min}} \tan(\Delta \psi)$$

(39)

Therefore, the aircraft will start to bank as soon as the distance to the current waypoint is less than $R_{\text{din}}$, that is, when:

$$\sqrt{(x_{\text{wpt}} - x_{\text{aircraf}})^2 + (y_{\text{wpt}} - y_{\text{aircraf}})^2} \leq R_{\text{din}}$$

(40)

where $(x_{\text{wpt}}, y_{\text{wpt}})$ represents the coordinate position of the current waypoint, and $(x_{\text{aircraf}}, y_{\text{aircraf}})$ that of the aircraft.

The navigation strategy that we have just described allows the aircraft to initiate turn at the right distance from the next waypoint, as shown in the Fig. 9.

![Waypoint Navigation with Dynamic Range](image)

*Fig. 9. Navigation by Waypoint with Variable Range*
IV. Development for Three-Dimensional Navigation

IV.1. Dynamic Model

The dynamic model, used for the development of the 3D navigation, is a quasi-linearized model for small path angle $\gamma$:

\begin{align}
\dot{x} &= V \cos \psi \\
\dot{y} &= V \sin \psi \\
\dot{z} &= V \gamma \\
\dot{\psi} &= \frac{g}{V} \tan \sigma \\
\dot{\gamma} &= f(\delta)
\end{align}

being $\delta$ the deflection of the vertical control surfaces, $V$ the speed, $\psi$ the heading, $g$ the gravity acceleration, $\sigma$ the bank angle.

Let the controls be defined as:

\begin{align}
u_1(t) &= \tan \sigma \\
u_2(t) &= f(\delta)
\end{align}

The model is subject to the following constraints:

- One constraint applied to the state, in particular on the flight path angle
  \[ |\gamma| < 5^\circ \]

- Two other constraints applied to the controls
  \[ |\sigma| \leq 80^\circ \text{ and } |\dot{\gamma}| = |\dot{\nu}_2| \leq 30^\circ/s \]

Therefore the model will be split into a state $x_1$ composed of three elements and a state $x_2$ composed of two elements, that is:

\[ x_1 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad x_2 = \begin{bmatrix} \psi \\ \gamma \end{bmatrix} \]

IV.2. Waypoint Navigation

First of all, the controller has been tested for a 3D navigation along fixed waypoints.

Let:

\[ Q_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} h^2 & 0 \\ 0 & h^2 \end{bmatrix}, \quad R_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \]

where $\epsilon=10^{-6}$ is a parameter to prevent matrix $Q_1$ from being singular, $h$ is the time-step, and

\[ e_z = x_z - q_z \]

The error is defined as:

\[ e = \sqrt{(x_{ref} - x)^2 + (y_{ref} - y)^2 + (z_{ref} - z)^2} \]

which is actually the distance between the aircraft and the next waypoint.

The termination condition for the simulation is when:

\[ e \leq 50m \]

Let consider the following waypoints:

\[ q_1 = \begin{bmatrix} 1000 \\ 0 \\ 1000 \end{bmatrix} \text{ and } q_2 = \begin{bmatrix} 1000 \\ 0 \\ 1000 \end{bmatrix} \]
The aircraft reaches the second waypoint after a series of manoeuvres. These are due to the saturation of the flight path angle since the system is considered for small angles.

Below is reported the history of the two controls during the upward flight.

Finally all the tests performed are joined in a simplified mission. Splitting the mission in three phases (climb – loiter – approach), the control is obtained by three different controllers specifically designed. The simulation is run placing the three controllers in cascade. Each controller inputs the last state and control vectors of the previous one. Furthermore, the same data passing applies to the derivatives output in order to ensure the continuity of the derivatives.

Climb phase.

In this section the result for a 3D navigation along waypoints is reported. The original program has been implemented with the waypoint variable range as described in section III.4.
Fig. 19. Mission Climb – Control History $u_2$

The behaviour of the control is illustrated in Figs. 18 and 19.

Loiter phase.
A loiter consists of a cruise flight for a certain amount of time over a specified area. The loiter phase occurs, for general aviation, generally at the end of the flight plan, normally when the plane is waiting for clearance to land. However, some unmanned aircraft used for special purposes, like reconnaissance, monitoring or surveillance, may have a loiter phase in mid-flight.

STANAG 4586 [9] defines a set of loiter patterns along which a UAV should navigate in case of area surveillance and monitoring. These patterns are: circle, race-track, 8-shape (also called “figure-8”). In general the flight altitude and the speed on a given loiter are kept constant during the mission.

The results of a circular loiter are described in this paragraph. That loiter is almost similar to the one reported in section III. The difference is that in the latter case the controller has to maintain the aircraft at a constant altitude. Therefore the trajectory will be described by the following continuous system:

$$\bar{x}(t) = R \cos \omega t$$  \hspace{1cm} (55)

$$\bar{y}(t) = R \sin \omega t$$  \hspace{1cm} (56)

$$\bar{z}(t) = \text{constant}$$  \hspace{1cm} (57)

$$\varphi(t) = \frac{\pi}{2} + \omega t$$  \hspace{1cm} (58)

where the variables have the same meaning as in section III.2, $z$ being a constant altitude.

Let the controller parameters be:

$$Q_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} h^2 & 0 \\ 0 & h^2 \end{bmatrix},$$  \hspace{1cm} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$  \hspace{1cm} (59)

Then, the trajectory of the aircraft is displayed in Figs. 20 and 21.

Fig. 20. Mission Loiter – Isometric View

Fig. 21. Mission Loiter – Altitude

The control history of the transient is reported in Figs. 22 and 23.

Fig. 22. Mission Loiter – Control History $u_1$

Fig. 23. Mission Loiter – Control History $u_2$

Approach phase.
The approach phase is performed using the same principle as the one used for the climbing. The only difference is the waypoint set which is constant and set on:
\[ q_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \]  \hspace{1cm} (60)

V. Discussion of the Results

As previously shown, the simulation results are clearly affected by the time-step parameter \( h \). For this reason a preliminary study about \( h \) and \( Q \) influence is presented below.

All the tests were performed changing only one parameter and keeping the other constant. The first parameter which is modified is the time-step \( h \).

The way the time-step parameter affects the performance of the controller is depicted in a series of figures, all designated by Fig. 24, that show the loiter tracking error expressed as the deviation of the aircraft trajectory across-time with respect to the actual curve of the loiter.

Let the following parameters:

\[ Q_1 = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}, \quad Q_2 = 0.01h^2, \quad R_2 = 1 \]  \hspace{1cm} (61)

The value of \( h \) (time-step) that yields the minimum error after 2 minutes is \( h=0.1s \). In spite of this, it doesn’t reach the target trajectory as smoothly as when higher time-step values (\( h>0.1s \)) are used. Furthermore, its transient state is longer than in most of the other cases. A better response is, instead, obtained setting the value \( h=0.7s \). It reaches an error smaller than 20m in about 14s, afterwards it remains stable around the value of 13m. From the value of \( h=0.9s \) it is possible to see how the response presents some oscillations during the transient phase. In particular, observing the response for \( h=1.1 \) and \( h=1.3 \), despite bigger oscillations in the beginning, the steady-state errors are larger for higher values of \( h \) than for lower ones.
Matrix $Q$ is another parameter that affects notably the response. Let:

$$Q = c \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & h^2 \end{bmatrix}$$ (62)

where $c$ is a positive parameter. The loiter tracking error dynamics are depicted in a series of figures for different values of parameter $c$, these are designated by Fig 25.

First of all, a value suitable for the model had been found ($c=0.01$). Afterwards the parameter had been changed to analyze response variations. It is easy to observe that the error significantly decreases when increasing the value of $c$. With the highest value of $c$, it is possible to notice an oscillation during the first transient phase, similar to the one shown in 24. Another interesting similarity is that, although the value of the test parameter ($c=0.024$) is the highest, it causes a smaller oscillation during the transient phase but yields to a larger final error than in some other cases; for instance, when $c = 0.022$.

### VI. Conclusion

The presented method dealt with a nonlinear feedback control method based on Lie derivatives for UAV navigation guidance. This derives from a nonlinear predictive control concepts previously proposed by Lu and Khan [7],[8], but with an extension that we described that accounts for the adaptability of the time-step for the sake of improving the performance of the controller. Beyond the time-step adaptive update method, the paper described the various steps that were performed, in order to validate the navigation model along waypoints and on loiters. The model was first tested for planar navigation on a circular loiter. The first sample dealt with a reference trajectory described by a continuous system. The second test concerned a planar
navigation along waypoints and the third was a variable waypoint range model for a 3D navigation. The method was validated successfully on all the scenarios related to the planar and three-dimensional navigation patterns including the circular loiter.

The design of the controller is not elementary. Controller parameters such as \( h \), \( Q \), \( R \) and \( \alpha \) used for the controller design actually affect the performance of the controller and a further study should be conducted as an extension to the present work.

We emphasize that this is the first work, to our knowledge, that uses Lie derivative based nonlinear control theory to solve systematically waypoint and loiter navigation problems.

The limitations of the presented method are that the adaptive time-steps were used for loiter navigation but could not be fully explored for waypoint navigation due to the discontinuity of the discrete reference trajectory. Future work will aim at fully exploring the adaptive time-steps for waypoint navigation as well, so that adaptive control gains can be used not only for loiters but also for a waypoint sequence.

Acknowledgements

This research was conducted in the Aeronautics and Astronautics Research Center (AeroG) at the University of Beira Interior in Covilhã, Portugal, and supported by the Portuguese Foundation for Sciences and Technology.

References