Comparison of path-following algorithms for loiter paths of Unmanned Aerial Vehicles

Daniel M. Xavier*, Natassya B. F. Silva† and Kalinka R. L. J. C. Branco†

*Escola de Engenharia de São Carlos
Universidade de São Paulo
São Carlos, Brazil
danielmartinxavier@gmail.com

†Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
São Carlos, Brazil
naty.ays@gmail.com, kalinka@icmc.usp.br

Abstract—Unmanned Aerial Vehicles are an example of critical embedded system, since it uses specific hardware and software to control the vehicle through its operation. One of the main parts of the system is the autopilot, which is responsible for stabilising the aircraft during the flight, executing navigation tasks and sensing the environment. The path-following is an important capability of a UAV, allowing it to follow a desired path defined by waypoints. Several solutions of the path-following for loiter paths are described in the literature, but most of them only deals with the 2D scenario. Therefore, this paper presents an extension process to determine path-following algorithms based on Carrot-Chasing, Non-Linear Guidance Law (NLGL), Pure Pursuit and Line-of-Sight (PLOS) and Vector Field. It also demonstrates a comparison between these new algorithms using a simulation with wind disturbances, which shows that NLGL produces smaller errors and Carrot-Chasing and PLOS requires less effort.

Index Terms—Unmanned aerial vehicles, navigation, path-following algorithms

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are aircraft that can fly without an human aboard and are a clear example of a critical embedded system. They are composed by a combination of hardware and software developed to fulfil a specific task and a failure of their system can result in human risk or loss of high-value assets [1].

These vehicles can operate autonomously by following a predefined trajectory or can be remotely controlled. Their autonomous functionality are guaranteed by the autopilot, which is the system responsible for stabilising the aircraft during the flight, executing navigation tasks and sensing the environment [2].

One of the main important navigation tasks is the path-following, when the aircraft follows a path that is described by a list of waypoints [3]. The most usual paths are straight lines and loiters, which are circular flights around a specific point. This task is accomplished by path-following algorithms that rely in control laws and geometric methods, like Line-of-Sight (LOS) [4], non-linear control [5] and vector fields [6].

Park et al. [7] and Curry et al. [8] evaluated the Nonlinear Guidance Law (NLGL) path-following algorithm for curved paths and loiters. The former demonstrated the stability of the method with Asymptotic Lyapunov stability and presented flight tests of the algorithm for circular paths. The latter improved the algorithm and performed simulations with wind disturbances.

Kukreti et al. [9] proposed a path-following algorithm based on Linear Quadratic Regulator (LQR) for straight lines and loiter paths. The regulator gains were adjusted with a genetic algorithm and simulations results were described with wind disturbances.

Ali et al. [10] defined a guidance law based on nonlinear sliding mode to estimate and incorporate wind disturbances in a path-following algorithm with sliding surfaces. Simulation results showed the correct estimation of the wind vector and the improvement in the path-following algorithm.

Abozied abd Qin [11] developed a path-following algorithm based on Vector Field guidance by incorporating sliding mode controller and disturbance observer. Using these strategies, it was defined lateral and longitudinal control laws. The resulting algorithm was evaluated in a simulation environment for straight lines and helical paths with different parameters.

Sujit et al. [12], [13] compared a subset of path-following algorithms for straight lines and loiter paths. The authors analysed different parameters for each algorithm and executed simulation experiments with wind disturbances, concluding that Vector Field path-following was more accurate and required less effort than the others.

Only Abozied abd Qin [11] presented results with variation of the altitude, while most of the works with path-following algorithms dealt with loiter paths in only one plane, resulting in a 2D scenario. However, it is important to consider the altitude
during the UAV navigation tasks due to its inherent movement. Our preliminary work have focused in the comparison already considering the 3D scenario, however, the analysis was only performed for straight lines paths [14], [15].

Therefore, this paper presents new extended path-following algorithms for loiter paths based on Carrot-Chasing [13], [16], Non-Linear Guidance Law (NLGL) [7], Pure Pursuit and Line-of-Sight (PLOS) [17] and Vector Field [18]. Moreover, these new algorithms are compared in simulations with different wind intensities and directions according to the path error and effort.

The remaining part of the paper proceeds as follows: Section II describes the extended path-following algorithms, Section III presents the results of the simulations and the comparison metrics for experiments without and with different wind disturbances and Section IV summarises the conclusions.

II. 3D PATH-FOLLOWING ALGORITHMS

The path-following problem for loiter paths can be described as the minimisation of the distance between the aircraft position, \( p = [x, y, z]^T \) in the inertial frame, and the path position, which are circles with centres \( O_i = [x_i, y_i, z_i]^T \) and radius \( R_i \) (where \( i \) determines the order of the circles that specifies the entire path). When considering the kinematic model of the aircraft, its velocity vector direction can be controlled by changes in its attitude, thus, the output of the path-following algorithms are defined as the rates of pitch and yaw angles, \( q_d \) and \( r_d \) respectively. It is also considered that the position of the aircraft \( p \), its constant velocity intensity \( V \) and pitch and yaw angles represented by \( \psi \) and \( \theta \) are all previously estimated.

The known path-following algorithms Carrot-chasing, Non-Linear Guidance Law (NLGL), Pure Pursuit and Line-of-Sight (PLOS) and Vector Field were extended to consider the altitude for loiter paths. The extension process consists in two rotations to transform the error vector from the inertial frame to the path frame. This process was already applied successfully for straight lines paths [15] and a similar approach was implemented for loiter paths.

In the case of loiter paths, the path frame axis \( X \) is tangent to the loiter, resulting in a rotation of \( \alpha \) in the \( Z \) axis of the inertial frame as (1), where \( \alpha \) is calculated with (2). Afterwards, this intermediate frame is rotated around its \( Y \) axis by \( \beta \) as (3), where \( \beta \) is obtained from (4).

\[
R_Z (\alpha) = \begin{pmatrix}
\cos(\alpha) & -\sin(\alpha) & 0 \\
\sin(\alpha) & \cos(\alpha) & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

\[
\alpha = \arctan\left( \frac{y - y_i}{x - x_i} \right) + \frac{\pi}{2}
\]

\[
R_Y (\beta) = \begin{pmatrix}
\cos(\beta) & 0 & \sin(\beta) \\
0 & 1 & 0 \\
-\sin(\beta) & 1 & \cos(\beta)
\end{pmatrix}
\]

\[
\beta = \arctan\left( \frac{z_i^R - z^R}{\sqrt{(x^R - x_i^R)^2 + (y^R - y_i^R)^2}} \right)
\]

From these rotations, the error vector \( e = [e_x, e_y, e_z]^T \) in the path frame can be obtained from the difference of the aircraft position \( p \) and the loiter centre \( O_i \) as (5). However, \( e_y \) is composed by the real error in the \( Y \) axis, called cross-track error, plus the loiter radius. The other elements of the error vector are the along-track error \( e_x \) and the vertical-track error \( e_z \).

\[
e = R_Y (\beta) R_Z (\alpha) (p - O_i)
\]

The cross-track error and the vertical-track error are directly related to the changes necessary to direct the aircraft to the loiter path. All equations for the 3D scenario were derived from the techniques already presented in the literature for the 2D scenario. Therefore, the control laws presented for each extended path-following algorithm in the next subsections, which uses the errors and the extensions process, were adapted to guarantee the convergence for the 3D desired trajectories.

A. Carrot-chasing

The extended Carrot-chasing algorithm uses virtual targets determined by the parameters \( \lambda \) and \( \delta \) to direct the aircraft to the loiter paths. In the extended algorithm, the preliminary control law of the Carrot-chasing is applied to obtain the yaw rate [13] and a control law based on the Lookahead algorithm [16] is used to obtain the pitch rate. This combination, associated with the error vector in the path frame resulted from the extension process and the proportional gains \( K_\psi \) and \( K_\theta \) to obtain the angular rates, guarantees the convergence to the desired path, as can be seen in Algorithm 1.

B. Non-Linear Guidance Law

The extended NLGL algorithm uses virtual targets as intersections between the desired path and circles with radius \( L_y \) and \( L_z \) with nonlinear control laws. Because the intersection can result in two points, the virtual target is chosen by comparing the signal of the cross-track error in the function \( \text{Intersection}_xy \). The intersection represented by the function \( \text{Intersection}_{xz} \) is a intersection between the line of the circle plane and the circle centred at the aircraft position, thus, the first point is always chosen as the virtual target. The nonlinear control laws are described in Algorithm 2.

C. Pure Pursuit and Line-of-Sight

The extended PLOS algorithm for loiter paths uses two strategies, the Pure Pursuit with gains \( K_{PP,y} \) and \( K_{PP,z} \), and the LOS guidance with gains \( K_{LOS,y} \) and \( K_{LOS,z} \). The Pure Pursuit part of the algorithm directs the aircraft by considering the reference angles from the path frame, while the LOS part considers the cross-track error and vertical-track error, both detailed in Algorithm 3.
Algorithm 1 Carrot-chasing algorithm for loiter paths.

1: function CARROT_CHASING($O_i = [x_i, y_i, z_i]^T$, $R_i$, $p = [x, y, z]^T$, $\psi$, $\theta$, $\lambda$, $\delta$, $K_\psi$, $K_\theta$)
2: $\alpha \leftarrow \arctan \left( \frac{y - y_i}{x - x_i} \right) + \frac{\pi}{2}$
3: $O_i^R \leftarrow [x_i^R, y_i, z_i]^T$
4: $O_i^R \leftarrow R_z(\alpha)O_i$
5: $p_i^R \leftarrow [x_i^R, y_i, z_i]^T$
6: $q_i \leftarrow K_\psi(\psi_i - \psi)$
7: $r_d \leftarrow K_\theta(\theta_i - \theta)$
8: return $(r_d, q_d)$
9: end function

Algorithm 2 NLGL algorithm for loiter paths.

1: function NLGL($O_i = [x_i, y_i, z_i]^T$, $R_i$, $p = [x, y, z]^T$, $\psi$, $\theta$, $L_y$, $L_z$, $V$
2: $\alpha \leftarrow \arctan \left( \frac{y - y_i}{x - x_i} \right) + \frac{\pi}{2}$
3: $O_i^R \leftarrow [x_i^R, y_i, z_i]^T$
4: $O_i^R \leftarrow R_z(\alpha)O_i$
5: $p_i^R \leftarrow [x_i^R, y_i, z_i]^T$
6: $q_i \leftarrow K_\psi(\psi_i - \psi)$
7: $r_d \leftarrow K_\theta(\theta_i - \theta)$
8: return $(r_d, q_d)$
9: end function

Algorithm 3 PLOS algorithm for loiter paths.

1: function PLOS($O_i = [x_i, y_i, z_i]^T$, $R_i$, $p = [x, y, z]^T$, $\psi$, $\theta$, $K_{PP,\psi}$, $K_{LOS,\psi}$, $K_{PP,\theta}$, $K_{LOS,\theta}$)
2: $\alpha \leftarrow \arctan \left( \frac{y - y_i}{x - x_i} \right) + \frac{\pi}{2}$
3: $O_i^R \leftarrow [x_i^R, y_i, z_i]^T$
4: $O_i^R \leftarrow R_z(\alpha)O_i$
5: $p_i^R \leftarrow [x_i^R, y_i, z_i]^T$
6: $q_i \leftarrow K_\psi(\psi_i - \psi)$
7: $r_d \leftarrow K_\theta(\theta_i - \theta)$
8: return $(r_d, q_d)$
9: end function

D. Vector Field

The extended Vector Field algorithm directs the aircraft to the desired loiter path with a flow constructed from the cross-track error and vertical-track error. The algorithm follows a sliding mode approach that result in an aggressive change in the direction when the aircraft is far away from the path and a smoother change when it is closer. The control laws resulting from the expansion process are represented in Algorithm 4.

III. ALGORITHMS COMPARISON

All extended path-following algorithms were implemented in MATLAB and are compared using simulations of the kinematic model for a fixed-wing UAV. The kinematic model, simulation results and the comparison are presented in the following subsections.

A. Kinematic Model

The kinematic model of a stabilised flight with constant velocity intensity $V$ and disturbed by a wind with intensity $V_w$ and direction $\psi_w$ can be represented by (6)-(10) [13].

$$\dot{x} = V \cos(\psi) \cos(\theta) + V_w \cos(\psi_w)$$  \hspace{1cm} (6)

$$\dot{y} = V \sin(\psi) \cos(\theta) + V_w \sin(\psi_w)$$  \hspace{1cm} (7)

$$\dot{z} = -V \sin(\theta)$$  \hspace{1cm} (8)

$$\dot{\psi} = \dot{\theta}$$  \hspace{1cm} (9)

$$r = \frac{\psi}{\dot{\psi}}$$  \hspace{1cm} (10)

This kinematic model with wind disturbances was implemented in MATLAB to represent the aircraft movement and
which can affect the algorithms performance. However, the
in the comparison are presented in Table I.

smallest error visually. The resulting parameters that are used
a range of values and choosing the ones that resulted in the
were determined previously by evaluating the response from
physical constraints, the pitch and yaw rates were limited as
Also, to represent a more realistic scenario for an aircraft with
is applied in the comparison of the path-following algorithms.
Also, to represent a more realistic scenario for an aircraft with
metric of the comparison, it is
considered the norm of the cross-track error and vertical-track error as (11) and the commanded values of the pitch and yaw rates to represent the aircraft effort.

\[ E_N = \sqrt{(e_y - R)^2 + e_z^2} \]  

For all simulations, the desired path is a loiter with centre at \( O = [100, 100, 10]^T \) and radius \( R = 90 \), and the aircraft initial position is \( \text{Pini} = [20, 20, 0]^T \). Figure 1 shows the resulting trajectories for the extended path-following algorithms and Table II presents the metrics means (represented as a bar in the symbol) for the experiment without wind. The means are calculated for the absolute values obtained at each step of the simulation, which was executed with a time step of 0.01 seconds.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| Carrot-Chasing   | \( \lambda = 0.2, \delta = 10, \)  
|                  | \( K_\psi = 0.2, K_\theta = 0.4 \) |
| NLGL             | \( L_y = 6, L_z = 10 \) |
| PLOS             | \( K_{PP,\psi} = 12, K_{LOS,\psi} = 0.9, \)  
|                  | \( K_{PP,\theta} = 2.75, K_{LOS,\theta} = 0.8 \) |
| Vector Field     | \( \chi_{ap} = 5.5, \tau = 0.5, \gamma = 1.65, \)  
|                  | \( k_y = 1, k_z = 1.3, \epsilon = 100, \)  
|                  | \( \lambda = \pi, \kappa_y = 1, \kappa_z = 0.5 \) |

correctness of the algorithms was firstly verified in an environment without wind. As metrics of the comparison, it is considered the norm of the cross-track error and vertical-track error as (11) and the commanded values of the pitch and yaw rates to represent the aircraft effort.

\[ E_N = \sqrt{(e_y - R)^2 + e_z^2} \]  

\[ \text{TABLE I} \]

\textbf{PARAMETERS OF ALL EXTENDED PATH-FOLLOWING ALGORITHMS USED IN THE COMPARISON.}

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| Carrot-Chasing | \( \lambda = 0.2, \delta = 10, \)  
|              | \( K_\psi = 0.2, K_\theta = 0.4 \) |
| NLGL         | \( L_y = 6, L_z = 10 \) |
| PLOS         | \( K_{PP,\psi} = 12, K_{LOS,\psi} = 0.9, \)  
|              | \( K_{PP,\theta} = 2.75, K_{LOS,\theta} = 0.8 \) |
| Vector Field | \( \chi_{ap} = 5.5, \tau = 0.5, \gamma = 1.65, \)  
|              | \( k_y = 1, k_z = 1.3, \epsilon = 100, \)  
|              | \( \lambda = \pi, \kappa_y = 1, \kappa_z = 0.5 \) |

is applied in the comparison of the path-following algorithms. Also, to represent a more realistic scenario for an aircraft with physical constraints, the pitch and yaw rates were limited as \( ||q|| < 0.19 \text{ rad/s} \) and \( ||r|| < 0.33 \text{ rad/s} \) [13]. It is also assumed that the commanded pitch and yaw rates are immediately performed by the aircraft when smaller than their limited values, which is appropriate because an inner loop with higher frequency in the autopilot could guarantee its execution.

\[ E_N = \sqrt{(e_y - R)^2 + e_z^2} \]  

\[ \text{TABLE II} \]

\textbf{MEANS OF THE NORM OF THE CROSS-TRACK AND VERTICAL-TRACK ERRORS CALCULATED AS } E_{\Delta} \text{ OF THE COMMANDED PITCH RATE } q \text{ AND OF THE COMMANDED YAW RATE } p. \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>( \bar{E}_N )</th>
<th>( \bar{q} )</th>
<th>( \bar{p} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrot-Chasing</td>
<td>0.30</td>
<td>0.01</td>
<td>0.03</td>
</tr>
<tr>
<td>NLGL</td>
<td>0.16</td>
<td>0.56</td>
<td>0.96</td>
</tr>
<tr>
<td>PLOS</td>
<td>0.16</td>
<td>0.02</td>
<td>0.04</td>
</tr>
<tr>
<td>Vector Field</td>
<td>0.85</td>
<td>0.55</td>
<td>0.03</td>
</tr>
</tbody>
</table>

It is possible to observe that NLGL and PLOS presented the smallest errors, while Vector Field resulted in the largest one. However, when considering the effort, Carrot-Chasing had the smallest values, followed by PLOS. Individually, the commanded pitch rate for NLGL and Vector Field were considerably bigger than the others, but Vector Field’s commanded yaw rate was one of the smallest.

The comparison considering the wind depends on the wind intensity and direction, so 30 replications were executed with different values for them. These values were calculated from a Gaussian distribution with covariance of 0.01 for the intensity and 0.2 for the direction. The number of replications was

\[ E_N = \sqrt{(e_y - R)^2 + e_z^2} \]  

\[ \text{TABLE II} \]

\textbf{MEANS OF THE NORM OF THE CROSS-TRACK AND VERTICAL-TRACK ERRORS CALCULATED AS } E_{\Delta} \text{ OF THE COMMANDED PITCH RATE } q \text{ AND OF THE COMMANDED YAW RATE } p. \]

\[ \text{TABLE II} \]

\textbf{MEANS OF THE NORM OF THE CROSS-TRACK AND VERTICAL-TRACK ERRORS CALCULATED AS } E_{\Delta} \text{ OF THE COMMANDED PITCH RATE } q \text{ AND OF THE COMMANDED YAW RATE } p. \]

\[ \text{TABLE II} \]

\textbf{MEANS OF THE NORM OF THE CROSS-TRACK AND VERTICAL-TRACK ERRORS CALCULATED AS } E_{\Delta} \text{ OF THE COMMANDED PITCH RATE } q \text{ AND OF THE COMMANDED YAW RATE } p. \]
chosen as the one that gives no intersection between the confidence intervals, resulting in a statistically viable analysis [19].

Figure 2 provides the trajectories from all extended path-following algorithms for the first replication with a wind intensity of 0.11 m/s and 20° yaw.

The means of the comparison metrics were calculated for each replication and, afterwards, the mean and standard deviation for all replications were obtained from all the means. Table III presents the means (represented as a $M$ with the symbol subscripted) and standard deviations (represented as a $\sigma$ with the symbol subscripted) of the metrics for the means of all replications with the different wind intensities and directions.

**TABLE III**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$M_{E_N}$</th>
<th>$\sigma_{E_N}$</th>
<th>$M_q$</th>
<th>$\sigma_q$</th>
<th>$M_r$</th>
<th>$\sigma_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrot-Chasing</td>
<td>4.25</td>
<td>3.59</td>
<td>0.01</td>
<td>0</td>
<td>0.08</td>
<td>0.06</td>
</tr>
<tr>
<td>NLGL</td>
<td>1.31</td>
<td>0.93</td>
<td>1.85</td>
<td>1.05</td>
<td>3.19</td>
<td>1.80</td>
</tr>
<tr>
<td>PLOS</td>
<td>2.85</td>
<td>2.14</td>
<td>0.02</td>
<td>0</td>
<td>0.09</td>
<td>0.06</td>
</tr>
<tr>
<td>Vector Field</td>
<td>14.25</td>
<td>11.95</td>
<td>1.84</td>
<td>1.05</td>
<td>0.08</td>
<td>0.06</td>
</tr>
</tbody>
</table>

Similarly to the results obtained from the experiment without wind, NLGL presented the smallest error, followed by PLOS and then Carrot-Chasing. The error resulted from Vector Field is significantly larger than the others. Related to the efforts, Carrot-Chasing and PLOS produced the smallest angular rates, requiring less effort to follow the loiter path.

In general, the standard deviations from the error and efforts are very close to the means, showing that the algorithms performance varies with the different wind intensities and directions. The exceptions are the pitch rate for Carrot-Chasing and PLOS, which resulted in a value very close to zero, demonstrating the consistence of their control laws for the attitude.

**IV. CONCLUSIONS**

Extended path-following algorithms for loiter paths in the 3D scenario were defined for Carrot-Chasing, NLGL, PLOS and Vector Field. Previous results in the literature showed that Vector Field was more accurate than the other path-following algorithms for the 2D scenario. However, the simulation results for the 3D scenario with and without wind disturbances demonstrated that Vector Field has largest errors than Carrot-Chasing, NLGL and PLOS.

All the simulation results with and without wind disturbances validated the new algorithms and a comparison in respect to the error path and the effort was analysed in this paper. In both flight conditions, NLGL presented the smallest error, which was close to the PLOS’s error in the experiment without wind. However, when considering the efforts, Carrot-Chasing and PLOS produced the smallest angular rates, requiring less effort to follow the loiter path.

As future work, it will be investigated automatic adjustment of the parameters for the extended path-following algorithms to improve its performance and the rejection of disturbances. It is also possible to explore adaptive control laws with the estimation of the wind during the execution of the path-following to enhance the algorithms performance. Finally, low-level controllers will be used to execute real flight experiments to compare all extended path-following algorithms.
References


