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

The authors acknowledge the support granted by FAPESP through process 2017/21303-2. Research was also sponsored by the Army Research Office and was accomplished under Grant Number W911NF-18-1-0012. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorised to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

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, $\mathbf{p} = [x, y, z]^T$ in the inertial frame, and the path position, which are circles with centres $\mathbf{O_i} = [x_i, y_i, z_i]^T$ 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 $\mathbf{p}$, its constant velocity intensity V and pitch and yaw angles represented by ψ and θ 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 α in the Z axis of the inertial frame as (1), where α is calculated with (2). Afterwards, this intermediate frame is rotated around its Y axis by β as (3), where β is obtained from (4).

$$\boldsymbol{R_Z}(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

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

$$\boldsymbol{R_{Y'}}(\beta) = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 1 & \cos(\beta) \end{pmatrix} \tag{3}$$

$$\beta = arctan\left(\frac{z_i^R - z^R}{\sqrt{\left(x^R - x_i^R\right)^2 + \left(y^R - y_i^R\right)^2}}\right) \tag{4}$$

From these rotations, the error vector $\mathbf{e} = [e_x, e_y, e_z]^T$ in the path frame can be obtained from the difference of the aircraft position $\mathbf{p}$ and the loiter centre $\boldsymbol{O_i}$ 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.

$$\boldsymbol{e} = \boldsymbol{R_{Y'}}(\beta)\,\boldsymbol{R_Z}(\alpha)\,(\boldsymbol{p} - \boldsymbol{O_i}) \tag{5}$$

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 λ and δ 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_ψ and K_θ 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 $Intersection\_xy$. The intersection represented by the function $Intersection\_xz$ 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.

function CARROT_CHASING($\boldsymbol{O_i} = [x_i, y_i, z_i]^T$, R_i, $\boldsymbol{p} = [x, y, z]^T$, ψ, θ, λ, δ, K_ψ, K_θ)
  $\alpha \leftarrow arctan\left(\frac{y-y_i}{x-x_i}\right) + \frac{\pi}{2}$
  ▷ $\boldsymbol{O_i^R} = \left[x_i^R, y_i^R, z_i^R\right]^T$
  $\boldsymbol{O_i^R} \leftarrow \boldsymbol{R_z}(\alpha)\boldsymbol{O_i}$
  ▷ $\boldsymbol{p^R} = \left[x^R, y^R, z^R\right]^T$
  $\boldsymbol{p^R} \leftarrow \boldsymbol{R_z}(\alpha)\boldsymbol{p}$
  $\beta \leftarrow arctan\left(\frac{z_i^R - z^R}{\sqrt{\left(x^R - x_i^R\right)^2 + \left(y^R - y_i^R\right)^2}}\right)$
  ▷ $\boldsymbol{e} = [e_x, e_y, e_z]^T$
  $\boldsymbol{e} = \boldsymbol{R_y}(\beta)\,\boldsymbol{R_z}(\alpha)\,(\boldsymbol{p} - \boldsymbol{O_i})$
  ▷ Calculates the virtual target
  $x_t \leftarrow R_i cos(\alpha + \lambda) + x$
  $y_t \leftarrow R_i sin(\alpha + \lambda) + y$
  $\psi_d \leftarrow arctan\left(\frac{y_t - y}{x_t - x}\right)$
  $\theta_d \leftarrow \beta + arctan\left(\frac{e_z}{\delta}\right)$
  $r_d \leftarrow K_\psi\,(\psi_d - \psi)$
  $q_d \leftarrow K_\theta\,(\theta_d - \theta)$
  return (r_d, q_d)
end function

Algorithm 2 NLGL algorithm for loiter paths.

function NLGL($\boldsymbol{O_i} = [x_i, y_i, z_i]^T$, R_i, $\boldsymbol{p} = [x, y, z]^T$, ψ, θ, L_y, L_z, V)
  $\alpha \leftarrow arctan\left(\frac{y-y_i}{x-x_i}\right) + \frac{\pi}{2}$
  ▷ $\boldsymbol{O_i^R} = \left[x_i^R, y_i^R, z_i^R\right]^T$
  $\boldsymbol{O_i^R} \leftarrow \boldsymbol{R_z}(\alpha)\boldsymbol{O_i}$
  ▷ $\boldsymbol{p^R} = \left[x^R, y^R, z^R\right]^T$
  $\boldsymbol{p^R} \leftarrow \boldsymbol{R_z}(\alpha)\boldsymbol{p}$
  $\beta \leftarrow arctan\left(\frac{z_i^R - z^R}{\sqrt{\left(x^R - x_i^R\right)^2 + \left(y^R - y_i^R\right)^2}}\right)$
  ▷ Calculates intersection in xy plane between circle of radius L_y with centre at $[x, y]$ and path circle with centre $[x_i, y_i]$
  ▷ $s_{xy} = [x_{xy}, y_{xy}, z_{xy}]^T$
  $s_{xy} \leftarrow Intersection\_xy(L_y, [x, y], r, [x_i, y_i])$
  $\nu_{yaw} \leftarrow arctan\left(\frac{y_{xy} - y}{x_{xz} - x}\right)$
  $\eta_{yaw} \leftarrow \nu_{yaw} - \psi$
  ▷ Calculates intersection in xz plane between circle of radius L_z with centre at $[x, y]$ and path circle with centre $[x_i, y_i]$
  ▷ $s_{xz} = [x_{xz}, y_{xz}, z_{xz}]^T$
  $s_{xz} \leftarrow Intersection\_xz(L_z, [x, z], r, [x_i, z_i])$
  $\nu_{pitch} \leftarrow arctan\left(\frac{-(z_{xz} - z)}{x_{xz} - x}\right)$
  $\eta_{pitch} \leftarrow \nu_{pitch} - \theta$
  $r_d \leftarrow \frac{2V^2}{L_y} sin(\eta_{yaw})$
  $q_d \leftarrow \frac{2V^2}{L_z} sin(\eta_{pitch})$
  return (r_d, q_d)
end function

Algorithm 3 PLOS algorithm for loiter paths.

function PLOS($\boldsymbol{O_i} = [x_i, y_i, z_i]^T$, R_i, $\boldsymbol{p} = [x, y, z]^T$, ψ, θ, $K_{PP,\psi}$, $K_{LOS,\psi}$, $K_{PP,\theta}$, $K_{LOS,\theta}$)
  $\alpha \leftarrow arctan\left(\frac{y-y_i}{x-x_i}\right) + \frac{\pi}{2}$
  ▷ $\boldsymbol{O_i^R} = \left[x_i^R, y_i^R, z_i^R\right]^T$
  $\boldsymbol{O_i^R} \leftarrow \boldsymbol{R_z}(\alpha)\boldsymbol{O_i}$
  ▷ $\boldsymbol{p^R} = \left[x^R, y^R, z^R\right]^T$
  $\boldsymbol{p^R} \leftarrow \boldsymbol{R_z}(\alpha)\boldsymbol{p}$
  $\beta \leftarrow arctan\left(\frac{z_i^R - z^R}{\sqrt{\left(x^R - x_i^R\right)^2 + \left(y^R - y_i^R\right)^2}}\right)$
  ▷ $\boldsymbol{e} = [e_x, e_y, e_z]^T$
  $\boldsymbol{e} = \boldsymbol{R_y}(\beta)\,\boldsymbol{R_z}(\alpha)\,(\boldsymbol{p} - \boldsymbol{O_i})$
  $\beta \leftarrow arctan\left(\frac{z_i^R - z_{i+1}^R}{x_{i+1}^R - x_i^R}\right)$
  $\theta_d \leftarrow \beta$
  ▷ $\boldsymbol{e} = [e_x, e_y, e_z]^T$
  $\boldsymbol{e} = \boldsymbol{R_y}(\beta)\,\boldsymbol{R_z}(\alpha)\,(\boldsymbol{p} - \boldsymbol{W_i})$
  ▷ Calculates control laws for Pure Pursuit (PP) and LOS
  $r_d \leftarrow K_{PP,y}(\alpha - \psi)$ + $K_{LOS,y}(-e_y - R_i))$
  $q_d \leftarrow K_{PP,z}(\beta - \theta)$ + $K_{LOS,z}(e_z))$
  return (r_d, q_d)
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 ψ_w can be represented by (6)-(10) [13].

$$\dot{x} = V\cos(\psi)\cos(\theta) + V_w\cos(\psi_w) \tag{6}$$
$$\dot{y} = V\sin(\psi)\cos(\theta) + V_w\sin(\psi_w) \tag{7}$$
$$\dot{z} = -V\sin(\theta) \tag{8}$$
$$q = \dot{\theta} \tag{9}$$
$$r = \dot{\psi} \tag{10}$$

This kinematic model with wind disturbances was implemented in MATLAB to represent the aircraft movement and

Algorithm 4 Vector Field algorithm for loiter paths.

function VECTOR_FIELD($\boldsymbol{O_i} = [x_i, y_i, z_i]^T$, R_i, $\boldsymbol{p} = [x, y, z]^T$, ψ, θ, χ_{ap}, τ, γ, k_y, k_z, ϵ, λ, V, κ_y, κ_z)
  $\alpha \leftarrow arctan\left(\frac{y-y_i}{x-x_i}\right) + \frac{\pi}{2}$
  ▷ $\boldsymbol{O_i^R} = \left[x_i^R, y_i^R, z_i^R\right]^T$
  $\boldsymbol{O_i^R} \leftarrow \boldsymbol{R_z}(\alpha)\boldsymbol{O_i}$
  ▷ $\boldsymbol{p^R} = \left[x^R, y^R, z^R\right]^T$
  $\boldsymbol{p^R} \leftarrow \boldsymbol{R_z}(\alpha)\boldsymbol{p}$
  $\beta \leftarrow arctan\left(\frac{z_i^R - z^R}{\sqrt{\left(x^R - x_i^R\right)^2 + \left(y^R - y_i^R\right)^2}}\right)$
  ▷ Check if the aircraft is too far away from the circle, requiring a higher effort
  if $|e_y| > 2R_i$ **then**
    ▷ Control laws for the desired yaw
    $\psi_d \leftarrow \alpha + \pi + arcsin\left(\frac{R_i}{-e_y}\right)$
    $\psi_c \leftarrow \psi_d + \frac{\pi}{2} + \frac{\pi}{3}\frac{V}{-e_y\epsilon} sin\left(\psi - \alpha\right)$
  else
    ▷ Control laws for the desired yaw
    $\psi_d \leftarrow \alpha - \frac{\pi}{2} - \frac{\pi}{3}(\frac{-e_y - R_i}{R_i})^{k_y}$
    $\psi_c \leftarrow \psi_d - \frac{V}{-e_y\epsilon}\sin\left(\psi - \theta\right) - \frac{k_y V \pi}{3R_i^{k_y}\epsilon}\left(e_y - R_i\right)^{k_y - 1}\cos\left(\psi - \alpha\right)$
  end if
  $e_z \leftarrow -e_z$
  $\theta_R \leftarrow \theta - \beta$
  $\chi_d \leftarrow -\frac{2}{\pi}\chi_{ap} arctan(k_z e_z)$
  $\tilde{\chi} \leftarrow \theta_R - \chi_d$
  ▷ Control laws for the desired pitch
  $\chi_{cont} \leftarrow (\theta_R - \frac{\chi_{ap}}{\gamma}\frac{2}{\pi}\frac{k_z V sin(\theta_R)}{1+(k_z e_z^2)} - (\frac{\lambda}{\gamma}) sat\frac{\tilde{\chi}}{\tau})$
  $\chi_{cont} \leftarrow \chi_{cont} + \beta$
  $\theta_c \leftarrow \gamma(\chi_{cont} - \theta)$
  $r_d \leftarrow \kappa_y(\psi_c - \psi)$
  $q_d \leftarrow \kappa_z(\theta_c - \theta)$
  return (r_d, q_d)
end function

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 $\|q\| < 0.19$ rad/s and $\|r\| < 0.33$ 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.

B. Simulations experiments

The parameters of each extended path-following algorithm were determined previously by evaluating the response from a range of values and choosing the ones that resulted in the smallest error visually. The resulting parameters that are used in the comparison are presented in Table I.

The comparison takes into account the wind disturbances, which can affect the algorithms performance. However, the 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.

TABLE I
PARAMETERS OF ALL EXTENDED PATH-FOLLOWING ALGORITHMS USED IN THE COMPARISON.

Algorithm	Parameters
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	$\chi_{ap} = \frac{\pi}{2}$, $\tau = 0.5$, $\gamma = 1.65$, $k_y = 1$, $k_z = 1.3$, $\epsilon = 100$, $\lambda = \pi$, $\kappa_y = 1$, $\kappa_z = 0.5$

$$E_N = \sqrt{(e_y - R)^2 + e_z^2} \tag{11}$$

For all simulations, the desired path is a loiter with centre at $\mathbf{O} = [100, 100, 10]^T$ and radius $R = 90$, and the aircraft initial position is $\mathbf{p_{ini}} = [20, 20, 0]^T$. 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 II
MEANS OF THE NORM OF THE CROSS-TRACK AND VERTICAL-TRACK ERRORS CALCULATED AS E_M, OF THE COMMANDED PITCH RATE q AND OF THE COMMANDED YAW RATE r.

Algorithm	$\bar{\mathbf{E}}_\mathbf{N}$	$\bar{q}$	$\bar{r}$
Carrot-Chasing	0.30	0.01	0.03
NLGL	0.16	0.56	0.96
PLOS	0.16	0.02	0.04
Vector Field	0.85	0.55	0.03

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 considerable 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

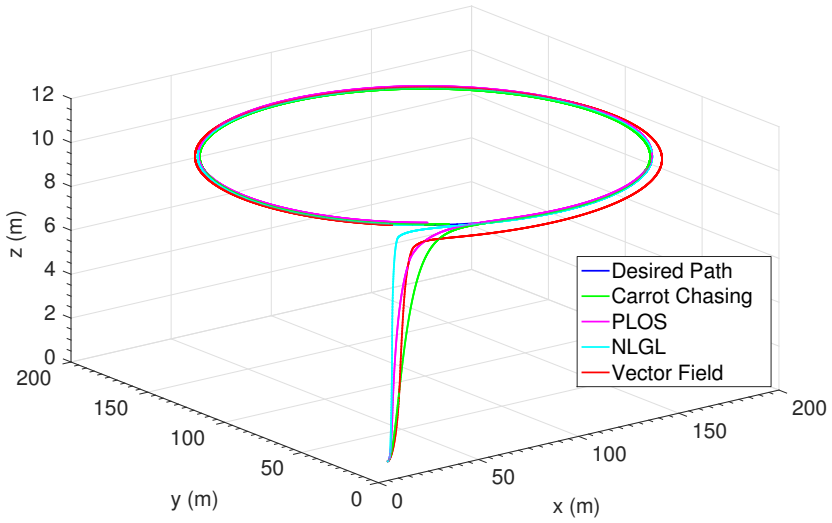

Fig. 1. Trajectories resulting from simulations of the extended path-following algorithms Carrot-Chasing, NLGL, PLOS and Vector Field for the experiment without wind.

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 σ with the symbol subscripted) of the metrics for the means of all replications with the different wind intensities and directions.

TABLE III
MEAN AND STANDARD DEVIATION OF THE MEANS OF THE NORM OF THE CROSS-TRACK AND VERTICAL-TRACK ERRORS CALCULATED AS E_M, OF THE COMMANDED PITCH RATE q AND OF THE COMMANDED YAW RATE r FOR 30 REPLICATIONS WITH DIFFERENT WIND INTENSITIES AND DIRECTIONS.

Algorithm	$\mathbf{M_{\bar{E}_N}}$	$\boldsymbol{\sigma_{\bar{E}_N}}$	$\boldsymbol{M_{\bar{q}}}$	$\boldsymbol{\sigma_q}$	$\boldsymbol{M_{\bar{r}}}$	$\boldsymbol{\sigma_r}$
Carrot-Chasing	4.25	3.59	0.01	0	0.08	0.06
NLGL	1.31	0.93	1.85	1.05	3.19	1.80
PLOS	2.85	2.14	0.02	0	0.09	0.06
Vector Field	14.25	11.95	1.84	1.05	0.08	0.06

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 presented the smallest values, while NLGL presented the highest efforts for both pitch and yaw rates.

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 altitude.

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.

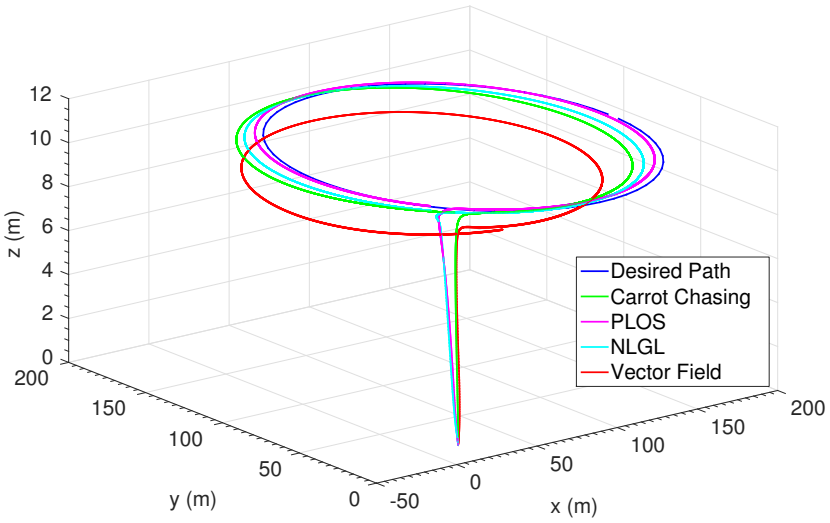

Fig. 2. Trajectories resulting from simulations of the extended path-following algorithms Carrot-Chasing, NLGL, PLOS and Vector Field for the first replication with a wind intensity of 0.11 m/s and 20° yaw.

REFERENCES

[1] V. Januzaj *et al.*, "New challenges in the development of critical embedded systemsan aeromotive perspective," in *Leveraging Applications of Formal Methods, Verification, and Validation*, ser. Lecture Notes in Computer Science, T. Margaria and B. Steffen, Eds. Springer Berlin Heidelberg, 2010, vol. 6415, pp. 1–2.

[2] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of Unmanned Aerial Vehicles*. USA: Springer Publishing Company, Incorporated, 2015.

[3] G. Elkaim, F. Lie, and D. Gebre-Egziabher, "Principles of guidance, navigation, and control of UAVs," in *Handbook of Unmanned Aerial Vehicles*, K. P. Valavanis and G. J. Vachtsevanos, Eds. Springer Netherlands, 2015, pp. 347–380.

[4] R. Gudi, M. Ramana, S. A. Varma, and M. Kothari, "4th ifac conference on advances in control and optimization of dynamical systems acods 2016 motion planning for a fixed-wing UAV in urban environments," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 419 – 424, 2016.

[5] S. H. Mathisen, K. Gryte, T. I. Fossen, and T. A. Johansen, "Non-linear model predictive control for longitudinal and lateral guidance of a small fixed-wing UAV in precision deep stall landing," in *AIAA Infotech@ Aerospace*, 2016, p. 0512.

[6] W. Jung, S. Lim, D. Lee, and H. Bang, "Unmanned aircraft vector field path following with arrival angle control," *Journal of Intelligent & Robotic Systems*, pp. 1–15, 2016.

[7] S. Park, J. Deyst, and J. P. How, "Performance and lyapunov stability of a nonlinear path following guidance method," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1718–1728, November 2007.

[8] R. Curry, M. Lizarraga, B. Mairs, and G. Elkaim, "L+2, an improved line of sight guidance law for UAVs," in *American Control Conference (ACC), 2013*, June 2013, pp. 1–6.

[9] S. Kukreti, M. Kumar, and K. Cohen, "Genetically tuned LQR based path following for UAVs under wind disturbance," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2016, pp. 267–274.

[10] S. U. Ali, M. Z. Shah, R. Samar, and A. Waseem, "Wind estimation for lateral path following of uavs using higher order sliding mode," in *2016 International Conference on Intelligent Systems Engineering (ICISE)*, Jan 2016, pp. 364–371.

[11] M. A. H. Abozied and S. Qin, "High performance path following for uav based on advanced vector field guidance law," in *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Oct 2016, pp. 555–564.

[12] P. Sujit, S. Saripalli, and J. Sousa, "An evaluation of UAV path following algorithms," in *2013 European Control Conference (ECC)*, July 2013, pp. 3332–3337.

[13] P. B. Sujit, S. Saripalli, and J. B. Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless," *IEEE Control Systems*, vol. 34, no. 1, pp. 42–59, Feb 2014.

[14] G. V. Pelizer, N. B. F. da Silva, and K. R. L. J. Branco, "Comparison of 3d path-following algorithms for unmanned aerial vehicles," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, Miami, FL, USA, June 2017, pp. 498–505.

[15] G. V. Pelizer, N. B. F. Silva, and K. R. L. J. C. Branco, *3D Path-Following Algorithms for Unmanned Aerial Vehicles Adjusted with Genetic Algorithm*. Cham: Springer International Publishing, 2017, pp. 63–80. [Online]. Available: https://doi.org/10.1007/978-3-319-61403-8_4

[16] M. Breivik and T. I. Fossen, "Principles of guidance-based path following in 2D and 3D," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Dec 2005, pp. 627–634.

[17] K. Venkatraman, V. Mani, M. Kothari, I. Postlethwaite, and D.-W. Gu, "A suboptimal path planning algorithm using rapidly-exploring random trees," *International Journal of Aerospace Innovations*, vol. 2, no. 1-2, pp. 93–104, 2010.

[18] D. Nelson, D. Barber, T. McLain, and R. Beard, "Vector field path following for miniature air vehicles," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 519–529, June 2007.

[19] R. Jain, *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling.*, ser. Wiley professional computing. Wiley, 1991.