Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 15946–15951

Lyapunov Based 3D Path Following Kinematic Controller for a Fixed Wing UAV [*]

I. Lugo-Cárdenas [*] S. Salazar [**] R. Lozano [***]

[*] Heudiasyc UMR CNRS 7253 Laboratory, Université de Technologie
de Compiègne, Compiègne 60205, France (e-mail:
ilugocar@hds.utc.fr).

[**] UMI-LAFMIA CNRS 3175, Cinvestav 07300, Mexico (e-mail:
sergio.salazar.cruz@gmail.com).

[***] Sorbonne Universités, UTC, CNRS UMR 7253 Heudiasyc,
Compiègne France, and UMI-LAFMIA CNRS 3175, Cinvestav 07300,
Mexico (e-mail: rlozano@hds.utc.fr).

Abstract: Path following is a basic autonomous navigation capability for UAV's and in this paper we present a nonlinear 3D path following kinematic controller. In order to solve this problem, we use the typical multi-loop control structure in which the navigation controller directs the aircraft to the desired path, while the inner loop acts as an autopilot which stabilizes the nonlinear dynamics and tracks the command generated by the outer loop. Under the assumption that the autopilot is well tuned and the airspeed, fight-path angle, and bank angle states converge to the desired response to their commanded values, we use a kinematic model as a suitable control model for the fixed wing UAV. With the introduction of a virtual target, the proposed Lyapunov based kinematic controller for outer loop can steer the UAV asymptotically to approach the desired path. The proposed controller is validated in a 6-DoF simulation environment platform named MAV3DSim.

Keywords: 3D Path following, kinematic controller, Fixed wing UAV,.

1. INTRODUCTION

In recent years the development of UAV has increased with the low cost of sensors, airframes and electronics. Different non-military applications like aerial mapping, photography and agriculture. require that the UAV to autonomously follow a predefined path at a desired speed, in three-dimensional (3D) space.

A path following method using a virtual reference vehicle is proposed in the robotics area in D. Soetanto and Pascoal (2003) and G. V. Raffo and Normey-Rico (2009). On P. B. Sujit and Sousa (2014) the reader can find a survey on different path following algorithms for the guidance loop of fixed wing UAV. The algorithms addressed in the survey are Carrot-chasing, NLGL (Non Linear Guidance Law)S. Park (2007), VF (Vector Field)D. Nelson and Beard (2007), LQR (Linear Quadratic Regulator)A. Ratnoo and Kothari (2011) and PLOS (Pure Line of Sight)M. Kothari and Gu (2010) . The carrot-chasing algorithm was first introduced in Micaelli and Samson (1993) and the key idea is to make the UAV follow the virtual target by adjusting its attitude and approach the desired path.

It is common to solve the problem of path following by separating the vehicle guidance and control problem into an outer guidance loop and an inner control loop 1. The inner loop is designed to accurately follow the commands provided by the outer loop. There are alternative methods which integrates the inner loop and outer loop and they are designed simultaneously, as in Keviczky and Balas (2003) and Johnson and Corban (2002). Linear controllers are typically used for the outer loop of an aircraft. It is common to see proportional and derivative (PD) controller to reduce the magnitude of the cross-track error, which is the lateral deviation from the desired path (I. Rhee (2010)).

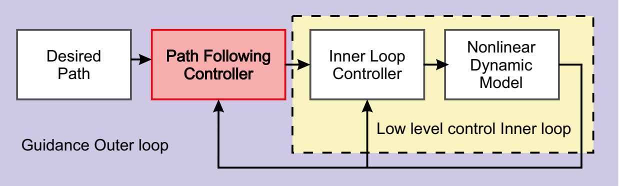

Fig. 1. Conceptual control structure.

The present paper is concerned only with the outer-loop method, as there are well-researched inner loop methods, in B. Kada (2011) and Y.Shin and Calise (1993),for implementing these commands in the aircraft. We propose a nonlinear path-following strategy, extending the 2D version we have already developed in G.Flores (2013), The key idea behind the proposed strategy, is to minimize the error of the path-following trajectory by using a virtual

[*] Research supported in part by the Mexican National Council for Science and Technology (CONACYT) under grant 218857

2405-8963 © 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.1747

particle s, which should be tracked along the path. We developed the error dynamic model suitable for control poruses and use a Lyapunov candidate function to design the controller.

The rest of this paper is organized as follows. In section II, the kinematic model for the path following problem of a fixed-wing UAV is constructed and the error kinematic model is derived. In section III, the guidance controller based on Lyapunov approach is designed to control the path following kinematics. Simulation results performed on the MAV3Dsim are shown in Section IV. Finally conclusions and future work are presented in Section V.

2. PROBLEM FORMULATION

This section provides a rigorous kinematic formulation for the problem of steering the aircraft along a desired path.

2.1 Aircraft Kinematic Model

The Dubins aircraft was first introduced in Chitsaz and LaValle (2007). We have build upon this model to increase accuracy in modeling the aircraft kinematics and to be more consistent with the commonly used aircraft models. Our model works under the assumption that the autopilot is well tuned, this means that the airspeed, flight-path angle and bank angles states converge with desired response to their commanded values. The following kinematic model describes the motion of the UAV:

$$\begin{aligned} \dot{x} &= V\cos\psi\cos\theta \\ \dot{y} &= V\sin\psi\cos\theta \\ \dot{z} &= -V\sin\theta \\ \dot{\psi} &= \omega \end{aligned} \tag{1}$$

where x, y and z denote the inertial position of the aircraft in a 3D inertial frame. ψ is the heading angle, ω is the heading angular rate, θ is the pitch angle, V is the airspeed of the aircraft which remains constant.

If we consider coordinated turn conditions, then we have the sideslip angle β equal to zero, the model for the heading angle, as in F.A.A. (2016), is given by,

$$\dot{\psi} = \frac{g}{V}\tan\phi \tag{2}$$

where g is the magnitude of gravity at sea level, ϕ is the roll angle, and $\dot{\psi}$ is the heading angular rate.

The state and control vectors of the above model are described respectively as:

$$\begin{aligned} \mathbf{x} &= [x\ y\ z\ \psi] \\ \mathbf{u} &= [\theta_c\ \phi_c] \end{aligned}$$

2.2 Problem Statement

The key idea behind the path-following controller relies on reducing two expressions to zero: The distance d between the aircrafts center of mass p and the point q on the path and the angle between the airspeed vector and the vector tangent to the trajectory.

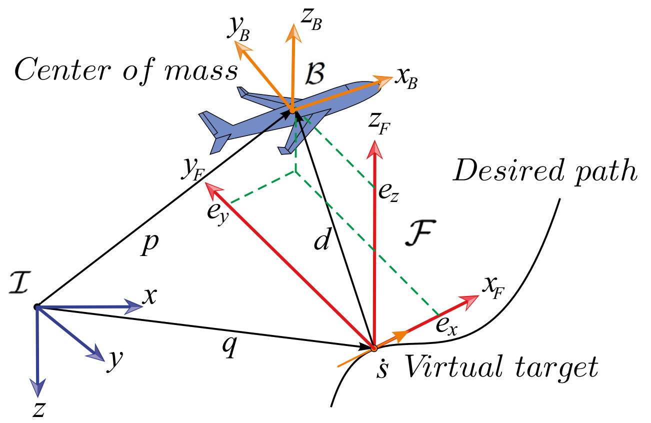

Fig. 2. Path following problem.

As depicted in Fig. 2, using a moving particle s along the trajectory at a velocity $\dot{s}$. Considering a frame attached to the virtual particle, the so called Serret Frame denoted by $\mathcal{S}$. We use $\dot{s}$ as a control input.

The angle and distance will become in the coordinates of the error space.

2.3 Error dynamics

Consider the 3D curve represented by smooth functions parameterized by t.Thus $(x_S(t), y_S(t), z_S(t))$ represent the virtual particle coordinates The inertial position of the aircraft is defined by $p = [x\ y\ z]^T$ in the inertial reference frame $\mathcal{I}$. To follow the given trajectory we define the inertial vector error $d^{\mathcal{I}} = p - q(s)$ expressed in $\mathcal{I}$, which will be minimized in order to track the desired trajectory. Note that the tangent vector coincides with the x-axis of the Serret frame $\mathcal{S}$, now consider the rotation of the vector tangent to the path in the z and y axis w.r.t the inertial frame $\mathcal{I}$ by the angles ψ_S and θ_S, respectively, also the aircraft has in itself another reference frame known as the body frame denoted by $\mathcal{B}$ and the body frame is not an inertial system and it is fixed to the moving vehicle. The. The angles ψ_S and θ_S can be obtained, as in Neto and Campos, from the parameterized curve as

$$\psi_S = \tan^{-1}\frac{y'_S(t)}{x'_S(t)}; \tag{3}$$

$$\theta_S = \frac{z'_S(t)}{\sqrt{x'_S(t)^2 + y'_S(t)^2}} \tag{4}$$

where $x'_S = \frac{dx_S(t)}{dt}$, $y'_S = \frac{dx_{yS}(t)}{dt}$, $z'_S = \frac{dz_S(t)}{dt}$

Consider the rotation matrix R_S from $\mathcal{I}$ to $\mathcal{S}$ and the rotation matrix R_B from $\mathcal{I}$ to $\mathcal{B}$:

$$R_S = \begin{bmatrix} \cos\theta_S\cos\psi_S & \cos\theta_S\sin\psi_S & -\sin\theta_S \\ -\sin\psi_S & \cos\psi_S & 0 \\ \sin\theta_S\cos\psi_S & \sin\theta_S\sin\psi_S & \cos\theta_S \end{bmatrix} \tag{5}$$

$$R_B = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\sin\psi & \cos\psi & 0 \\ \cos\psi\sin\theta & \sin\theta\sin\psi & \cos\theta \end{bmatrix} \tag{6}$$

The error d^I expressed in the Serret frame is given by

$$d^S = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} = R_S d^I = R_S \left(p - q\right) \tag{7}$$

Furthermore we define the yaw angle error as

$$e_\psi = \psi - \psi_S \tag{8}$$

To obtain the error state dynamic equations, we must compute the time derivative of (7) and (8). By differentiating (7) we obtain:

$$\begin{aligned} \dot{d}_S &= \dot{R}_S \left(p - q\right) + R_S \left(\dot{p} - \dot{q}\right) \\ &= \omega_S \times R_S \left(p - q\right) + R_S \left(\dot{p} - \dot{q}\right) \end{aligned} \tag{9}$$

where the ω_S represents the angular rate of $\mathcal{F}$ with respect to inertial frame $\mathcal{I}$, resolved in $\mathcal{F}$, then

$$\begin{aligned} \omega_S &= \begin{bmatrix} 0 \\ \dot{\theta}_S \\ 0 \end{bmatrix} + \begin{bmatrix} \cos\theta_S & 0 & -\sin\theta_S \\ 0 & 1 & 0 \\ \sin\theta_S & 0 & \cos\theta_S \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi}_S \end{bmatrix} \\ &= \begin{bmatrix} -\dot{\psi}_S \sin\theta_S \\ \dot{\theta}_S \\ \dot{\psi}_S \cos\theta_S \end{bmatrix} \end{aligned} \tag{10}$$

From (1), the time derivative of p and q can be expressed as

$$\dot{p} = R_B^T \begin{bmatrix} V \\ 0 \\ 0 \end{bmatrix} \tag{11}$$

$$\dot{q} = R_S^T \begin{bmatrix} \dot{s} \\ 0 \\ 0 \end{bmatrix} \tag{12}$$

The time derivative of (8) and using (2) results in

$$\begin{aligned} \dot{e}_\psi &= \dot{\psi} - \dot{\psi}_S \\ &= \frac{g}{V} \tan\phi - \dot{\psi}_S \end{aligned}$$

with

$$\begin{aligned} \dot{\psi}_S &= \kappa \dot{s} \\ \dot{\theta}_S &= \tau \dot{s} \end{aligned}$$

where $\frac{d\psi_s}{dt} = \kappa$ is the path curvature. The path curvature is expressed as a function of the trajectory coordinates $\left(x\left(t\right), y\left(t\right)\right)$ and its first and second derivatives w.r.t. the parameter t, i.e $x'_S = \frac{dx_S}{dt}$, $y'_S = \frac{dy_S}{dt}$. Thus the path curvature $\frac{d\psi_S}{dt} = \kappa$ is given by

$$\kappa = \frac{\left|y''_S x'_S - y'_S x''_S\right|}{\left(x'^2_S + y'^2_S\right)^{3/2}} \tag{13}$$

Finally using (9) and substituting (10), (11) and (12) we obtain the error kinematics model suitable for control purposes

$$\begin{aligned} \dot{e}_x &= \tau \dot{s} e_z - \kappa \dot{s} \cos\theta_S e_y + V \sin\theta \sin\theta_S \\ &\quad + V \cos\theta \cos\theta_S \cos e_\psi - \dot{s} \\ \dot{e}_y &= \kappa \dot{s} e_x \cos\theta_S + \kappa \dot{s} e_z \sin\theta_S + V \cos\theta \sin e_\psi \\ \dot{e}_z &= -\kappa \dot{s} e_y \sin\theta_S - \tau \dot{s} e_x + V \cos\theta \sin\theta_S \cos e_\psi \\ &\quad - V \sin\theta \cos\theta_S \\ \dot{e}_\psi &= \frac{g}{V} \tan\phi - \kappa \dot{s} \end{aligned} \tag{14}$$

Using the previous error kinematics model we design a feedback control laws for θ_c, ϕ_c and $\dot{s}$ such that all closed-loop signals are bounded and all the errors converge to zero.

3. CONTROLLER DESIGN

In this section we present a nonlinear guidance controller to follow a desired path. We design the controller for the guidance using the error kinematic model (14). As we are considering that the aircraft is equipped with an autopilot unit, which takes over the inner loop controller and it is well tuned, the outter loop controller outputs directly the desired pitch angle θ_c and the desired roll angle ϕ_c and passes it to the autopilot system. We also introduce a virtual controller in the form of the velocity $\dot{s}$ of the virtual particle which moves along the desired trajectory to stabilize all the error signals e_x, e_y, e_z, e_ψ to zero.

The Lyapunov design is a very powerful tool for control system design. The construction of the Lyapunov function is not an easy task and it usually involves a trial-and-error process. The Lyapunov design can be divided into two steps

- Choose a candidate Lyapunov function V for the system.
- Design a controller which renders its derivative $\dot{V}$ negative.

The control objective can be archived using a Lyapunov function candidate given by

$$V = \frac{1}{2} e_x^2 + \frac{1}{2} e_y^2 + \frac{1}{2} e_z^2 + \frac{1}{2} \left(e_\psi - \delta\left(e_y\right)\right)^2 \tag{15}$$

where $\delta\left(e_y\right)$ is a sigmoid function, as in our previous controller in G.Flores (2013), that introduce a desired approach angle as

$$\delta\left(e_y\right) = -\psi_a \frac{e^{2k_\delta e_y} - 1}{e^{2k_\delta e_y} + 1} \tag{16}$$

where $k_\delta > 0$ and $\psi_a \in \left[0, \frac{\pi}{2}\right]$.The sigmoid function (16) is bounded and differentiable w.r.t. the error e_y. It provides the desired relative course transition of the aircraft to the path as a function of e_y., and satisfies the following condition

$$e_y \sin\left(\delta\left(e_y\right)\right) \leq 0$$

which steer the aircraft in the correct direction, i.e., turn left when the aircraft is on the right side of the path, and turn right in the opposite situation. We note that the approach rate can be controlled by the adjustment of the parameter k_δ.

The time derivative of (15) along the trajectory of (14)

$$\begin{aligned}\dot V = {} & e_x\left(V\sin\theta\sin\theta_S + V\cos\theta\cos\theta_S\cos e_\psi - \dot s\right) + \\ & e_y V\cos\theta\sin e_\psi + \\ & e_z\left(V\cos\theta\sin\theta_S\cos e_\psi - V\cos\theta_S\sin\theta\right) + \\ & \left(e_\psi - \delta\left(e_y\right)\right)\left(\begin{array}{c}\dfrac{g}{V}\tan\phi - \kappa\dot s - \\ \dot\delta\left(e_y\right)\left(\begin{array}{c}\kappa\dot s e_x\cos\theta_S \\ +\kappa\dot s e_z\sin\theta_S + V\cos\theta\sin e_\psi\end{array}\right)\end{array}\right)\end{aligned} \tag{17}$$

Rearranging the terms of (17) we arrived to the expression

$$\begin{align}\dot V = e_x\left(\alpha - \theta\dot s\right) - e_z V\sin\theta\cos\theta_S + V e_y\sin\left(\delta\left(e_y\right)\right) \tag{18}\\ + \left(e_\psi - \delta\left(e_y\right)\right)\left(\frac{g}{V}\tan\phi + \beta\right) \tag{19}\end{align}$$

where

$$\begin{aligned}\alpha &= V\sin\theta\sin\theta_S + V\cos\theta\cos\theta_S\cos e_\psi \\ \beta &= -\kappa\dot s - \dot\delta\left(e_y\right)\left(\kappa\dot s e_x\cos\theta_S + \kappa\dot s e_z\sin\theta_S + V\cos\theta\sin e_\psi\right) \\ & \quad + V\frac{e_y\cos\theta\sin e_\psi + e_y\sin\delta\left(e_y\right) + e_z\cos\theta\sin\theta_S\cos e_\psi}{e_\psi - \delta\left(e_y\right)}\end{aligned}$$

and substituting the following kinematic control law

$$\begin{aligned}\dot s &= \left(\alpha + k_x e_x\right) \\ \theta &= \sin^{-1}\left(\frac{k_z e_z}{V\cos\theta_S}\right) \\ \phi &= \tan^{-1}\left(\frac{V}{g}\left(-\beta - k_y\left(e_\psi - \delta\left(e_y\right)\right)\right)\right)\end{aligned}$$

where k_x, k_y, k_z are positive real numbers, in (18), yields $\dot V = -k_x e_x^2 - k_z e_z^2 - k_y\left(e_\psi - \delta\left(e_y\right)\right)^2 + V e_y\sin\left(\delta\left(e_y\right)\right) \leq 0$

To conclude convergence of the states e_x, e_y, e_z, e_ψ to zero we use the LaSalle's invariance principle.

Theorem 1. LaSalle's Theorem

Let $\mathcal{O}$ be a positively invariant set of system (14). Let $\Omega \subset \mathcal{O}$ a set in which every solution which starts in Ω remains in Ω. Furthermore, let $\mathcal{M}$ be the largest invariant set contained in Ω. Then, as $t \to \infty$, every bounded solution starting in $\mathcal{O}$ converges to $\mathcal{M}$.

Proof: Convergence of the states (e_x, e_x, e_z, e_ψ) *to zero.*

The proof relies on Theorem 1. Consider the system (14) and the radially unbounded Lyapunov function candidate (15). Let us define the compact set $\mathcal{O}$ as $\mathcal{O} = \{V(e_x, e_y, e_z, e_\psi) \leq a\}$, where $a \in \Re^+$. Define the set Ω as

$$\Omega = \{[e_x \;\; e_y \;\; e_z \;\; e_\psi]^T \in \mathcal{O} \;\; : \;\; \dot V(e_x, e_y, e_z, e_\psi) = 0\} \tag{20}$$

Equivalently, the expression $\dot V(e_x, e_y, e_z, e_\psi) = 0$ means that $e_x = e_y = e_z = 0$ and $e_\psi = \delta(e_y)$. Since δ is a function of the error e_d, it is easy to verify that any point starting from Ω is an invariant set. Hence, by LaSalle Theorem, every trajectory starting in $\mathcal{O}$ converges to 0 as $t \to \infty$, and the following limits are true

- $\lim_{t\to\infty} e_x = 0$
- $\lim_{t\to\infty} e_y = 0$
- $\lim_{t\to\infty} e_z = 0$
- $\lim_{t\to\infty} e_\psi = 0$

4. SIMULATION

Simulations were done on a complete simulation platform, the MAV3DSim(Multi-Aerial Vehicle 3D Simulator) provide a complete 6 degrees of freedom (DoF) computer model of fixed wing aircraft. Simulation were done on a PC(CPU: Core i5, 2.3 GHz, RAM: 6 GB), the operating system is a Microsoft Windows 64Bits. The MAV3DSim hardware in the loop simulator software and functionality are fully described in I.Lugo-Cárdenas and R.Lozano (2014) and I. Lugo-Cárdenas and Lozano (2016). The MAV3DSim provides different configurations, we use it as a Hardware-in-the-loop simulator, this means that we are using the hardware used in the onboard electronics, in this case is a PIXHAWK autopilot Dev-Team (2016)which is a high-performance autopilot-on-module suitable for fixed wing, multi rotors, helicopters and many other robotic platform. It has a wide range target which goes from the high-end research to the amateur enthusiast. The data flow can be depicted in Fig.3. The MAV3DSim simulator collects the data from the simulation engine and it will send it to the Pixhawk autopilot, then, it will process it as if it was data collected from the physical sensors (inertial measurement unit, GPS, airspeed, barometer, etc), then the autopilot will compute the command for the control surfaces and send it back to the simulation engine.

Figure 4 shows the MAV3DSim graphic user interface, it contains an interactive map to which we can add a reference for trajectory tracking. A 3D view is also available in the simulator, and it useful for the 3D representation of the simulated aircraft attitude. Avionics instruments like those used in commercial aircrafts are used to display some of the state variables of the aircraft: An altimeter which indicates the altitude relative to a reference level at which the aircraft is flying. The attitude indicator shows the position of the longitudinal and transversal aircraft axes with respect to the natural horizon. The Heading indicator displays the aircraft heading with respect to magnetic north. Airspeed indicator Gives the aircraft speed relative to the surrounding air. Vertical speed indicator displays the vertical speed of the airplane, going down towards the center of the earth is negative velocity and going up is positive velocity.

In order to show the controller performance, we have chosen the following scenario: The 3D reference path has been chosen as

$$\begin{aligned}q_x &= R\cos\left(s/R\right) \\ q_y &= R\sin\left(s/R\right) \\ q_h &= bs/R + 200\end{aligned} \tag{21}$$

where $R = 200m$ is the rotation radius, s is the archlenght of the curve, the reference path has an initial altitude of $200m$. Using (21) we can now compute the curvature of the path using (13) as

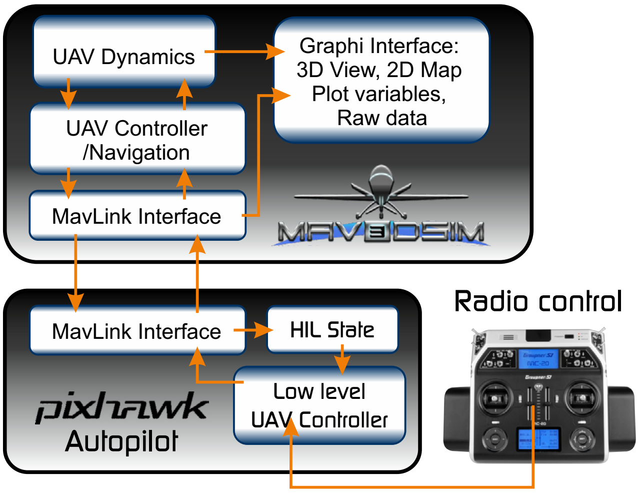

Fig. 3. MAV3DSim Hardware in the loop block diagram.

$$\kappa = \frac{|y''_S x'_S - y'_S x''_S|}{(x'^2_S + y'^2_S)^{3/2}} = \frac{1}{R} \qquad (22)$$

The constant velocity of the UAV was set to $10m/s$. The MAV3DSim simulator provides a full set of utilities for online gain tuning, with which we arrive to the following set of parameters for the guidance controller: $k_x = 0.1, k_y = 0.05, k_z = 0.05, \psi_a = \frac{\pi}{4}, k_\delta = 0.1$. The MAV3DSim is a complete simulation environment and the simulation starts with the vehicle in land, we start the simulation and manually take-off the aircraft to gain some altitude and then activate the path-following controller to follow the reference path.

Figure 5 shows the evolution of the reference in red line and the actual flight path with the blue line. It can be seen that the initial position of the aircraft is different from the initial position of the path and the control laws can eliminate this initial offset and steer the UAV along the path with a smooth movement. Figure 6 shows the position and attitude errors which converges to zero in 10s approximately. The output controllers are shown in fig.7, here we can notice that the virtual target converge to the constant velocity of the aircraft.

5. CONCLUSION

In this paper we present a nonlinear path-following kinematic controller for the fixed-wing aircraft based on a Lyapunov function candidate. The controller performance was tested in the MAV3DSim simulation environment which has been proven to be an excellent test bed for UAV controllers development. The controller was designed using a kinematic model of the aircraft, but it was tested on a full 6DoF simulation environment with good performance. The error space dynamics presented in this paper can be used with other types of controller to follow the virtual target.

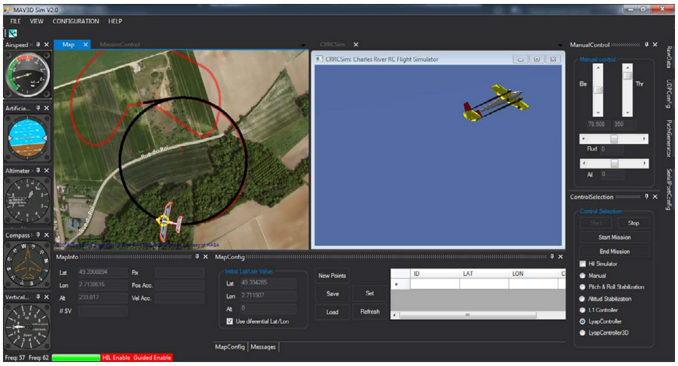

Fig. 4. The MAV3DSim simulation platform.

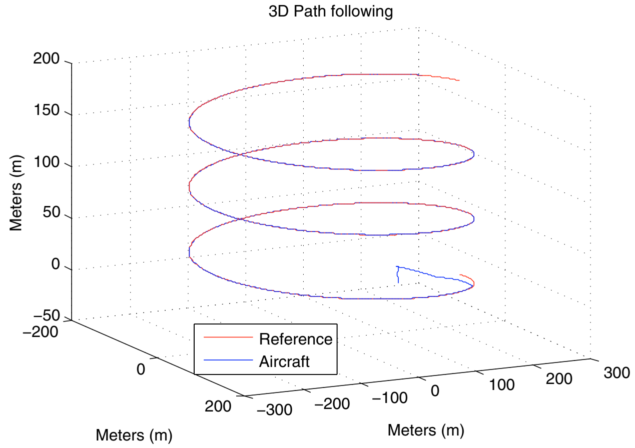

Fig. 5. Reference and actual 3D path of the UAV.

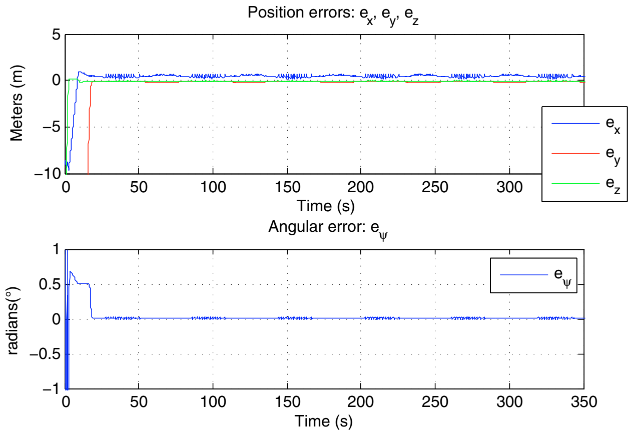

Fig. 6. Position errors and yaw error.

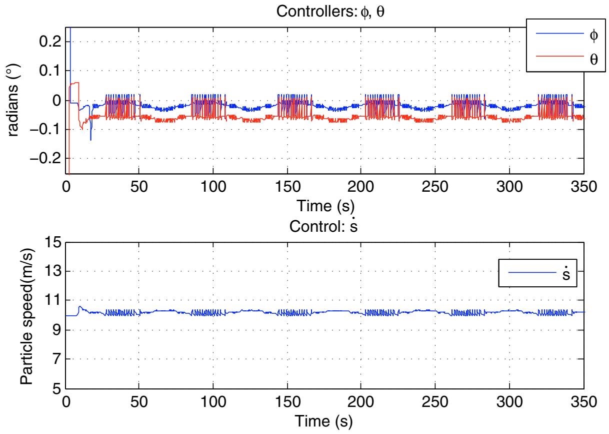

Fig. 7. Commanded roll θ and pitch ϕ angles.

The hardware in the loop simulation is an intermediate step between the pure simulation and the implementation on the experimental platform, and the next step is to implement this controller on a fixed-wing UAV experimental platform. Using the 3D path following presented in this paper we can design another level of autonomy and create a path generator to design new trajectories, this can also be used to perform an automatic take-off and landing on the UAV.

REFERENCES

A. Ratnoo, P.S. and Kothari, M. (2011). Optimal path following for high wind flights.

B. Kada, Y.G. (2011). Robust pid controller design for an uav flight control system.

Chitsaz, H. and LaValle, S. (2007). Time-optimal paths for a dubins airplane. In *Proc. IEEE Conference on Decision and Control (CDC'2007)*, 2379–2384. New Orleans, LA, USA.

D. Nelson, D. Barber, T.M. and Beard, R. (2007). Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics*.

D. Soetanto, L.L. and Pascoal, A. (2003). Adaptive, non-singular pathfollowing control of dynamic wheeled robots. In *Proc. IEEE Conference on Decision and Control (CDC'2003)*, 1765–1770. Maui, Hawaii,USA.

Dev-Team, P. (2016). Pixhawk. URL `https://pixhawk.org/choice`.

F.A.A. (2016). *Pilot's handbook of aeronautical knowledge*. United States Department of Transportation, U.S.A.

G. V. Raffo, G.K.G. and Normey-Rico, J.E. (2009). A predictive controller for autonomous vehicle path tracking. *IEEE Transactions on Intelligent Transportation Systems*, 92 – 102.

G.Flores, I. Lugo-Cárdenas, R. (2013). A nonlinear path-following strategy for a fixed-wing mav. *National Aeronautics and Space Administration*, 1014 – 1021.

I. Lugo-Cárdenas, S.S. and Lozano, R. (2016). The mav3dsim hardware in the loop simulation platform for research and validation of uav controllers.

I. Rhee, S. Park, C.R. (2010). A tight path following algorithm of an uas based on pid control.

I.Lugo-Cárdenas, G. and R.Lozano (2014). The mav3dsim: A simulation platform for research, education and validation of uav controllers. *Proceedings of the 19th IFAC World Congress, IFAC2014*, 713–717.

Johnson, E., C.A. and Corban, E. (2002). A six degree-of-freedom adaptive flight control architecture for trajectory following.

Keviczky, T. and Balas, G.J. (2003). Software enabled flight control using receding horizon techniques.

M. Kothari, I.P. and Gu, D. (2010). A suboptimal path planning algorithm using rapidly-exploring random trees. *International Journal of Aerospace Innovations*, 93104.

Micaelli, A. and Samson, C. (1993). Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. Technical Report 2097, INRIA.

Neto, A. and Campos, M. (????). A path planning algorithm for uavs with limited climb angle. In *2009 International Conference on Intelligent Robots and Systems (IROS)*.

P. B. Sujit, S.S. and Sousa, J.B. (2014). Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless. *Control Systems, IEEE*, 42 – 59.

S. Park, J. Deysty, J.H. (2007). Performance and lyapunov stability of a nonlinear path-following guidance method.

Y.Shin and Calise, A. (1993). Application of adaptive autopilot designs for an unmanned aerial vehicle. Technical Report 2097, INRIA.