2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Dubins Path Generation for a Fixed Wing UAV *.

Israel Lugo-Cárdenas[1] and Gerardo Flores[1] and Sergio Salazar [2] and Rogelio Lozano[1,2]

***Abstract*— A path generator is proposed for a fixed-wing Unmaned Aereal Vehicle (UAV). Assuming that the vehicle maintain a constant altitude, and airspeed, and that the UAV is constrained by a turning rate. The Dubins paths serve as a strategy to find the shortest path for the non-holonomic model of the UAV. Dubins paths consist of three path segments which are based on straight lines or arcs of circle of a given radius. The Dubins path generation is combined with a nonlinear Lypaunov-based path-following control. Finally we present a complete simulation environment in which the path generator and path following strategy are validated. As an example of application we propose the scenario in which a missing person is located in some known area and we use the path generator along with this path-following strategy applied to the fixed wing UAV to search and find this person.**

I. INTRODUCTION

Dubins[1] showed that a car-like robot with initial prescribed heading can arrive to its final position and heading, with exactly three paths segments which are either arcs of circles with a minimal radius or straight lines segments. Reeds and Sheep [2] solve a similar problem in which the vehicle can move forward as well as backward. Kavaraki and Svestka[3] use the Probabilistic Road Map (PRM) method which explore all the possible paths within the space surrounding the vehicle and finally select the lowest cost route. Other planning techniques used by Kuwata and Richards[4] are based on optimizations methods, such as Mixed Integer Linear Programming or Model Predictive Control techniques. Mehta and Egerstedt[5] used optimal control for constructing control programs from a given collection of motion primitives.

In this paper we present a path generator for a fixed-wing UAV using a reduced kinematic version of the lateral dynamics of an airplane, with constant altitude and velocity. This path generator uses the Dubins paths to generate the new path from the current position and direction of the plane to the desired position and direction. We use the nonlinear Lyapunov-based path-following strategy from our previous work [6] in order to follow the generated path. We combine the path generator with the path-following and validate it in a complete 6DOF simulation environment. A test scenario was developed in which a person(point of interest) is lost in a known area, we use the path generator to define a path to sweep this known area to search and find this point of interest.

The paper is organized as follows. In Section II we present the Dubins aircraft kinematic model used in this work along with the reference frame transformation needed to transform the geodetic coordinates $(Latitude, Longitud, Height)$ to a local tangent reference frame (x, y, z). Section III addresses the path generation for the Dubins paths. The explanation of the nonlinear Lyapunov-based path-following strategy used in this work is described in Section IV. The simulation carried out to validate the path generator in combination with the path-following strategy are presented in Section V. Finally the concluding remarks and future work are discussed in Section VI

II. DUBINS AIRCRAFT MODEL

In this section we present the coordinate transformation from the navigation frame to a local tangent which are required for the path-following strategy used in this work. The Dubins aircraft mathematical model is presented in this section, these are the kinematic dynamics are the basis of the development of the path-following strategy.

A. Coordinates Transformation

The geodetic coordinate system is used in many fields, such as: navigation, surveying and cartography, in order to define the position of an objet on the Earth's surface we use a set of three values called *geodetic coordinates* [7]. However, the geodetic coordinates lack of an intuitive understanding of distance, unlike other coordinate systems as the local East, North, Up (ENU) Cartesian coordinate system. The local ENU coordinates are formed from a plane tangent to the Earth's surface fixed to a specific location and it is known as a Local Tangent Plane (LTP). By convention the east axis is labeled x, the north y and the up z. The three different coordinate systems are represented in the Fig. 1.

1) Geodetic to ECEF coordinates: Here we introduce the equations to convert geodetic coordinates measurements to Local Tangent Plane coordinates. The method used passes through the Earth-Centered, Earth-Fixed (ECEF) rectangular coordinate system on the way to the Local Tangent Plane.

Geodetic coordinates (latitude τ , longitude λ , height h) can be converted into ECEF coordinates using the following relationships:

$$\begin{aligned} X &= (N(\tau)+h)\cos\tau\cos\lambda \\ Y &= (N(\tau)+h)\cos\tau\sin\lambda \\ Z &= \left(N(\tau)\left(1-e^2\right)+h\right)\sin\tau \end{aligned} \tag{1}$$

where

*This work is partially supported by the Mexican National Council for Science and Technology (CONACYT).

[1] are with the Heudiasyc UMR 6599 Laboratory, University of Technology of Compiègne, France (ilugocar,gfloresc,rlozano)@hds.utc.fr

[2] are with Laboratoire Franco-Mexicain d'Informatique et Automatique, LAFMIA UMI 3175 CNRS-CINVESTAV Mexico. (ssalazar)@ctrl.cinvestav.mx

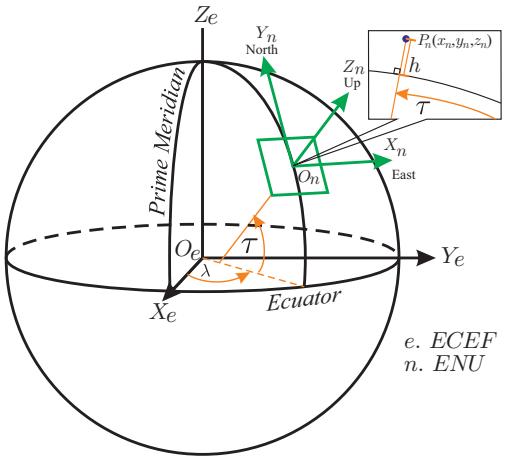

Fig. 1. Geodetic, ECEF and ENU coordinates frames.

$$N(\tau) = \frac{a}{\sqrt{1 - e^2 \sin^2 \tau}}$$

The semi-major axis and the first numerical eccentricity of the ellipsoid are represented by a and e, respectively, the numeric value of this constants can be found in the definition of the World Geodetic System 1984 [8]. $N(\tau)$ is the distance from the surface to the to the Z-axis along the ellipsoid normal.

2) ECEF to Local Tangent coordinates: A local reference point is needed to perform a coordinate transformation from ECEF to the local ENU coordinates. The launching site position will serve as the local reference point. If the launching site is at (λ_0, τ_0, h_0) in geodetic coordinates, then using the previous coordinate transformation we obtain (X_0, Y_0, Z_0), the launching site expressed in ECEF coordinates. The aircraft location is defined as (λ, τ, h); we use the same coordinate transformation to obtain (X, Y, Z), the aircraft position expressed in ECEF coordinates. The vector pointing from the launching site to the aircraft in the ENU coordinate system is computed as follows

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix} \tag{2}$$

where

$$\mathbf{R} = \begin{bmatrix} -\sin\lambda_0 & \cos\lambda_0 & 0 \\ -\sin\tau_0\cos\lambda_0 & -\sin\tau_0\sin\lambda_0 & \cos\tau_0 \\ \cos\tau_0\cos\lambda_0 & \cos\tau_0\sin\lambda_0 & \sin\tau_0 \end{bmatrix}$$

The World Geodetic System of 1984 (WGS84)[8] comprises a standard coordinate system and is one of the most used coordinate system used on GPS devices and we will use the coordinate transformations defined in this section to express the position of the airplane in the local ENU tangent plane which is suitable for the mathematical model and control purposes.

B. Mathematical model

The Dubins Airplane is described by the subsequent equations:

$$\begin{aligned} \dot{x} &= V_t \cos\psi \\ \dot{y} &= V_t \sin\psi \\ \dot{\psi} &= \omega \end{aligned} \tag{3}$$

in which x and y denotes the inertial position of the aircraft, ψ is the heading angle, ω is the heading rate, ϕ is the roll angle, V_t is the airspeed, i.e. the speed of an aircraft relative to the surrounding air. In this kinematic model the aircraft is considered to be moving with constant velocity V_t at a constant altitude h_d, Also, we assume no sideslip at a banked-turn maneuver.

The heading rate ω is induced by the roll angle of the airplane as

$$\omega = \frac{g}{V_t} \tan\phi \tag{4}$$

where g is the gravity acceleration. The roll angle is considered bounded under the following condition

$$|\phi| \leq \phi_{\max} \tag{5}$$

Assuming a coordinated turn, and given the boundedness of the roll angle ϕ the minimum turn radius ρ that the aircraft can fly is given by

$$\rho = \frac{V_t^2}{g \tan(\phi_{\max})} \tag{6}$$

In this kinematic model the position of the airplane can be represented by $p(x, y, \psi)$ with ψ measured from the y axis and (x, y) measured in the local ENU reference frame.

III. PATH GENERATION

In this section, the classical result of Dubins[1] is used as a basis for path generation. Dubins showed that the shortest path consist of exactly three path segments which are either a) arcs of a minimal radius or b) straight lines. The four different configurations for the Dubins paths which

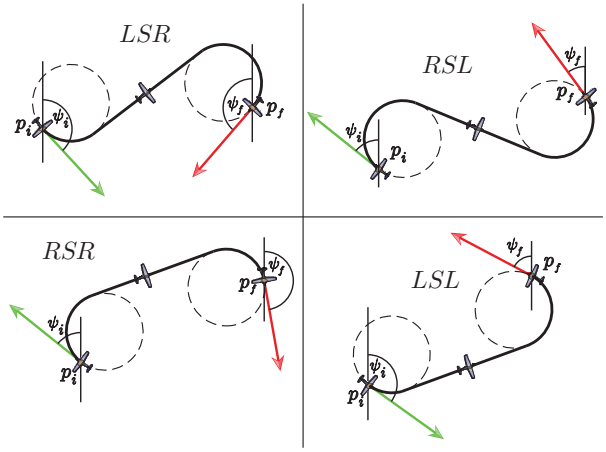

Fig. 2. Dubins shortest paths.

TABLE I

DUBINS PATH SELECTION.

Shortest distance	Dubins Path
CR_iCR_f	RSR
CR_iCL_f	RSL
CL_iCL_f	LSL
CL_iCR_f	LSR

are composed by two curved segments and a straight line segment are arranged as shown in Fig. 2. The four cases of Dubins paths are LSL, LSR, RSR, RSL; in which L stands for Left, R stands for Right and S for Straight.

The first step in determining the Dubins paths is to choose what type of path must be used. We have the initial and final configuration of the airplane, this is the initial position p_i, the initial heading ψ_i the final position p_f and the final heading ψ_f and with every initial-final configuration we can generate the 4 types of Dubins paths,i.e. from the starting point it can turn to the right or the left and arrive to the final point from the right or the left. We choose the shortest path by comparing the distance between the center of the circles, see Fig. 3. The smallest distance between the center of the circles gives us the shortest Dubin path according to the Table I.

Based on the initial and final configuration $(p_i, \psi_i)$and(p_f, ψ_f), respectively, and the minimal turn radius ρ from (6), the center of each circle is computed as follows

$$\begin{aligned} CR_i &= (x_{Ri}, y_{Ri}) = (x_i + \rho\cos\psi_i, y_i - \rho\sin\psi_i) \\ CL_i &= (x_{Li}, y_{Li}) = (x_i - \rho\cos\psi_i, y_i + \rho\sin\psi_i) \\ CR_f &= (x_{Rf}, y_{Rf}) = (x_f + \rho\cos\psi_f, y_f - \rho\sin\psi_f) \\ CL_f &= (x_{Lf}, y_{Lf}) = (x_f - \rho\cos\psi_f, y_f + \rho\sin\psi_f) \end{aligned}$$

A. Dubins path RSR

The initial and final configuration (p_i, ψ_i) and (p_f, ψ_f),respectively, are given w.r.t. an inertial frame

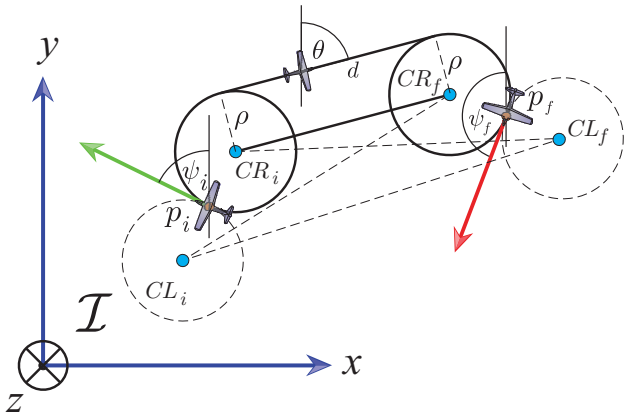

Fig. 3. The Dubins paths are chosen by comparing the distance between the center of the circles segments.

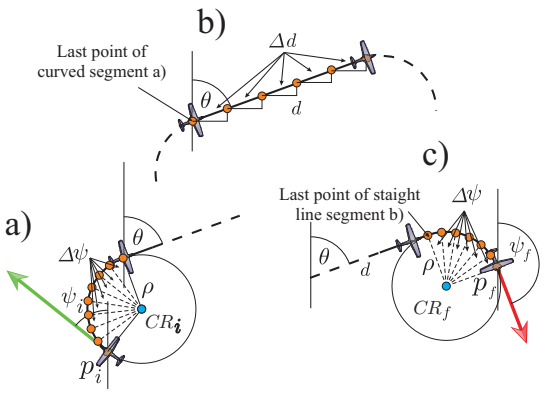

Fig. 4. The path generator algorithm produce an array of points p_n

(Loca ENU frame). The RSR is generated by a clockwise rotation from the initial position describing an arc of radius ρ and center CR_i with coordinates (x_{Ri}, y_{Ri}) until the aircraft heading achieves an angle of θ degrees. Then it follows a straight line segment d, finally it continues with a turn to the right describing an arc of radius ρ and center in CR_f with coordinates (x_{Rf}, y_{Rf}) until the plane arrives to the final heading ϕ_f as seen in Fig 3

The angle θ is the angle of the straight line segment d which is measured from the vertical y axis and computed as follows

$$\theta = \frac{\pi}{2} - \tan^{-1}\left(\frac{y_{Rf} - y_{Ri}}{x_{Rf} - x_{Ri}}\right) \tag{7}$$

The length $\bar{d}$ of the straight line segment d equals the distance $\overline{CR_iCR_f}$ between the center of the circles CR_i and CR_f and is computed as

$$d = \sqrt{(x_{Rf} - x_{Ri})^2 + (y_{Rf} - y_{Ri})^2} \tag{8}$$

The path generator algorithm produce an array of n points p_n which starts in $p_0 = p_i$ and ends in $p_n = p_f$.

The coordinates of the $n-th$ point p_n of the arc segments are obtained by rotating the initial point p_i clockwise around CR_i as a center

$$p_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_{Ri} + \rho\sin(\psi_n) \\ y_{Ri} + \rho\cos(\psi_n) \end{bmatrix} \tag{9}$$

where ψ_n starts at ψ_i and is incremented by given $\Delta\psi$ each time. These procedure is repeated until $\psi_n = \theta$, see Figure 4a.

Each point in the straight line segment is computed by incrementing the previous point p_{n-1} in a given Δd in direction of the angle θ as follows

$$p_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_{n-1} + \Delta d\sin(\theta) \\ y_{n-1} + \Delta d\cos(\theta) \end{bmatrix} \tag{10}$$

The elements p_n of the final segment are computed by rotating the final point of the straight line clockwise around CR_f as a center; see Fig. 4c.

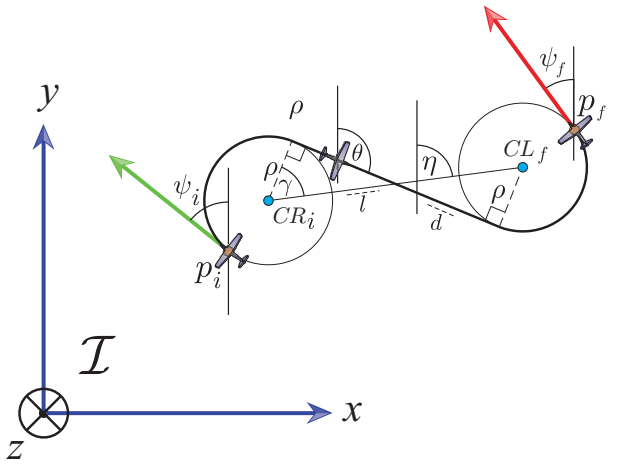

Fig. 5. Right-Straight-Left (RSL) Dubins path.

$$p_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_{Rf} + \rho \sin(\psi_n) \\ y_{Rf} + \rho \cos(\psi_n) \end{bmatrix} \tag{11}$$

where ψ_n starts in θ and each time is incremented by $\Delta\psi$. This procedure is repeated until $\psi_n = \psi_f$; see Figure 4c.

The complete path generation is summarized in algorithm 1

B. Dubins path RSL

This is the case where the closest circles are CR_i and CL_f, see Table I. From the initial and final configuration, (p_i, ψ_i) (p_f, ψ_f) the RSL path is generated with a clockwise rotation from the initial position p_i describing an arc of circle of radius ρ with center CR_i with coordinates (x_{Ri}, y_{Ri}) until the heading aircraft achieves the angle θ. Then it follows a straight line segment d, finally it will turn to the left describing an arc of radius ρ and center in CL_f with coordinates (x_{Lf}, y_{Lf}) until the aircraft reaches the final heading. See Figure 5.

Algorithm 1 Generate Dubin path RSR

$n = 1$; $p_0 = p_i$
$\psi_n = 0$
$\theta = \frac{\pi}{2} - \tan^{-1}\left(\frac{y_{Rf} - y_{Ri}}{x_{Rf} - x_{Ri}}\right)$
while $\psi_n \leq \theta$ **do**
  $p_n.x = x_{Ri} + \rho \sin(\psi_n)$; $p_n.y = y_{Ri} + \rho \cos(\psi_n)$
  $\psi_n = \psi_n + \Delta\psi$; $n = n + 1$
end while
$d_{sum} = 0$
while $d_{sum} \leq \bar{d}$ **do**
  $p_n.x = p_{n-1}.x + \Delta d \sin\theta$; $p_n.y = p_{n-1}.y + \Delta d \cos(\theta)$
  $d_{sum} = d_{sum} + \Delta d$; $n = n + 1$
end while
while $\psi_n \leq \psi_f$ **do**
  $p_n.x = x_{Ri} + \rho \sin(\psi_n)$; $p_n.y = y_{Ri} + \rho \cos(\psi_n)$
  $\psi_n = \psi_n + \Delta\psi$; $n = n + 1$
end while

Algorithm 2 Generate Dubin path RSL

$n = 1$; $p_0 = p_i$
$\psi_n = 0$
$\eta = \frac{\pi}{2} - \tan^{-1}\left(\frac{y_{Lf} - y_{Ri}}{x_{Lf} - x_{Ri}}\right)$
$\gamma = \tan^{-1}\left(\frac{2\rho}{d}\right)$
$\theta = \eta - \gamma + \frac{\pi}{2}$
while $\psi_n \leq \theta$ **do**
  $p_n.x = x_{Ri} + \rho \sin(\psi_n)$; $p_n.y = y_{Li} + \rho \cos(\psi_n)$
  $\psi_n = \psi_n + \Delta\psi$; $n = n + 1$
end while
$d_{sum} = 0$
while $d_{sum} \leq \bar{d}$ **do**
  $p_n.x = p_{n-1}.x + \Delta d \sin\theta$; $p_n.y = p_{n-1}.y + \Delta d \cos(\theta)$
  $d_{sum} = d_{sum} + \Delta d$; $n = n + 1$
end while
while $\psi_n \leq \psi_f$ **do**
  $p_n.x = x_{Li} + \rho \sin(\psi_n)$; $p_n.y = y_{Li} + \rho \cos(\psi_n)$
  $\psi_n = \psi_n + \Delta\psi$; $n = n + 1$
end while

In this case the angle θ is computed is computed aided by the triangle formed by the center of the circle CR_i the midpoint of the segment d and the point of the circle tangent to the straight line d using the following formula

$$\theta = \eta - \gamma + \frac{\pi}{2} \tag{12}$$

where η is the angle of the segment $\overline{CR_iCL_i}$ measured from the y axis as in Fig. 6 and is computed as follows

$$\eta = \frac{\pi}{2} - \tan^{-1}\left(\frac{y_{Lf} - y_{Ri}}{x_{Lf} - x_{Ri}}\right) \tag{13}$$

γ is the angle between the segment $\overline{CR_iCL_f}$ and the normal to the tangent point of circle CR_i and the segment d. γ is computed as follows

$$\gamma = \tan^{-1}\left(\frac{2\rho}{d}\right) \tag{14}$$

The length of the straight line segment d is computed with the distance l from the segment $\overline{CR_iCL_f}$ and the radius ρ as

$$d = \sqrt{l^2 - 4\rho^2} \tag{15}$$

The coordinates of the $n-th$ point p_n of the arc segments are obtained by rotating the initial point p_i clockwise around CR_i as a center using (9), see Figure 4a.

Each point in the straight line segment is computed by incrementing the previous point p_{n-1} in a given Δd in direction of the angle θ as in (10)

The elements p_n of the final segment are computed by rotating the final point of the straight line clockwise around CL_f as a center; see Fig. 4c.

$$p_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_{Lf} + \rho \sin(\psi_n) \\ y_{Lf} + \rho \cos(\psi_n) \end{bmatrix}$$

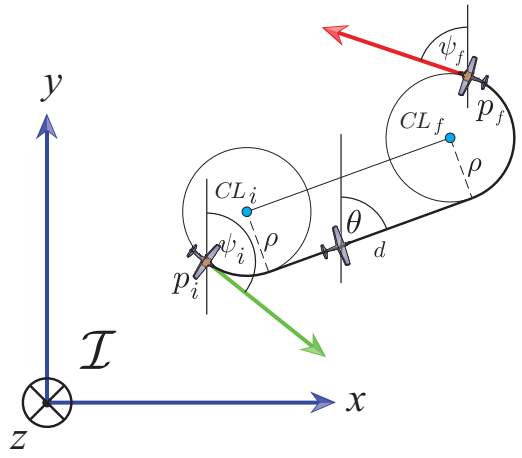

Fig. 6. Left-Straight-Left (LSL) Dubins path.

where ψ_n starts in θ and each time is incremented by $\Delta\psi$. This procedure is repeated until $\psi_n = \psi_f$; see Figure 4c.

The complete path generation is summarized in algorithm 2

C. Dubins path LSL

The LSL case is very similar to the RSR but with the turns to the left instead of right and it occurs when the smallest distance between the circles (see Fig. 3) is $\overline{CLiCLf}$. The LSL path is generated with a counterclockwise rotation from the initial position p_i describing an arc of a circle of radius ρ and center in CL_i with coordinates (x_{Li}, y_{Li}) until the aircraft heading achieves an angle of θ degrees. The it follows a straight line segment d and finally it continues with the a turn to the left describing an arc of radius ρ and center in CL_f with coordinates (x_{Lf}, y_{Lf}) until the airplane achieves the final heading ψ_f, as depicted in Fig. 6.

The angle θ measured from the vertical y axis is

$$\theta = \frac{\pi}{2} - \tan^{-1}\left(\frac{y_{Lf} - y_{Li}}{x_{Lf} - x_{Li}}\right) \tag{16}$$

The length of the segment d equals the distance $\overline{CL_iCL_f}$ and it is computed as

$$d = \sqrt{(x_{Lf} - x_{Li})^2 + (y_{Lf} - y_{Li})^2} \tag{17}$$

The coordinates of the $n-th$ point p_n of the arc segments are obtained by rotating the initial point p_i counterclockwise arround the CL_i as a center, as follows

$$p_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_{Li} + \rho\sin(\psi_n) \\ y_{Li} + \rho\cos(\psi_n) \end{bmatrix} \tag{18}$$

where ψ_n starts at zero and is incremented each time by $\Delta\psi$ until it reach the angle θ.

Each point in the straight line segment is computed by incrementing the previous point p_{n-1} in Δd in the same direction as θ using equation (**??**). The last curved segment is a turn to the left and the segment coordinates are computed as follows

$$p_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_{Lf} + \rho\sin(\psi_n) \\ y_{Lf} + \rho\cos(\psi_n) \end{bmatrix} \tag{19}$$

D. Dubins path LSR

According to Table I the Dubins path LSR is when the shortest distance is the one between the circles CL_i and CR_f. The first segment of this path is a left turn which generated with a counter-clockwise rotation from the initial position p_i describing an arc of radius ρ with center in $CL_i = (x_{Li}, y_{Li})$ until the airplane reach the heading θ, then it follows a straight line segment of length d and it finish with a right turn described by the arc of the circle of radius ρ with center in $CR_f = (x_{Rf}, y_{Rf})$ and it will turn until it achieve the angle ψ_f as depicted in Figure 7.

The computation of the angle θ is aided by the triangle formed by the center of the circle CL_i the midpoint of the segment d and the point of the circle tangent to the segment d using the following equation

$$\theta = \eta + \gamma - \frac{\pi}{2} \tag{20}$$

where

$$\eta = \frac{\pi}{2} + \tan^{-1}\left(\frac{y_{Rf} - y_{Li}}{x_{Rf} - x_{Li}}\right)$$

and

$$\gamma = \cos^{-1}\left(\frac{2\rho}{d}\right)$$

The length of the straight line segment d is computed with the following equation

$$d = \sqrt{l^2 - 4\rho^2} \tag{21}$$

The coordinates of the $n-th$ point p_n of the arc segments are obtained by rotating the initial point p_i counterclockwise around the CL_i as a center, as follows

$$p_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_{Li} + \rho\sin(\psi_n) \\ y_{Li} + \rho\cos(\psi_n) \end{bmatrix} \tag{22}$$

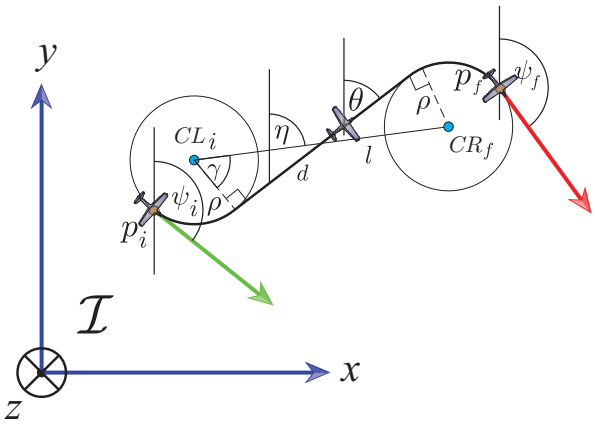

Fig. 7. Left-Straight-Left (LSR) Dubins path.

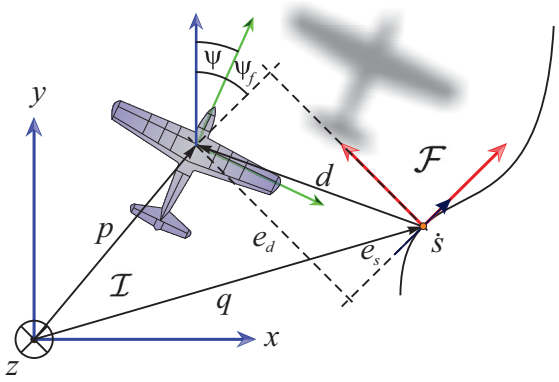

Fig. 8. Path following control problem schema.

where ψ_n starts at zero and is incremented each time by $\Delta\psi$ until it reach the angle θ.

Each point in the straight line segment is computed by incrementing the previous point p_{n-1} in Δd in the same direction as θ using equation (10). The last curved segment is a turn to the right and the segment coordinates are computed as follows

$$p_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_{Rf} + \rho \sin(\psi_n) \\ y_{Rf} + \rho \cos(\psi_n) \end{bmatrix} \tag{23}$$

IV. Path-Following Strategy

A Lyapunov-based path-following strategy which was developed in our previous work [6], is presented to steer a fixed-wing UAV along a desired path. Considering Fig. 8, the key idea behind the path-following controller relies on reducing the distance between the aircraft's center of mass p and the point q on the path to zero, as well as the angle between the airspeed vector and the tangent to the path at q. To accomplish these objectives, we introduce a virtual particle moving along the geometric path at a velocity $\dot{s}$. Consider a frame attached to such particle, this frame plays the role of a body axis of the virtual particle, and is the so called Serret-Frenet frame denoted by $\mathcal{F}$ [9]. It is worth noting that the particle velocity evolves according to a conveniently defined control law $\dot{s}$, yielding an extra controller design parameter.

Consider that the 2-D geometric path is represented by smooth functions parameterized by t, i.e. $x_s(t)$ and $y_s(t)$. Thus, $(x_s(t), y_s(t))$ represent the virtual particle coordinates.

The inertial position of the aircraft is defined by $p = [x \;\; y]^T$ in the inertial reference frame $\mathcal{I}$. For the purpose of following the given path, we define the inertial vector error $d^I = p - q(s)$ expressed in $\mathcal{F}$, which will be minimized in order to track the path. Such error vector d^I has been decomposed into its components e_s and e_d, corresponding to the error in the x-axis of the frame $\mathcal{F}$ and the error in the y-axis of the frame $\mathcal{F}$, respectively as it is shown in Fig. 8.

From the Fig. 8, we can see that the tangent vector to the path at $q(s)$ is parallel to x-axis of the frame $\mathcal{F}$. The angle ψ_f is measured from the inertial frame to the tangent vector of $q(s)$.

Considering an arbitrary point q on the path, and let

$$R = \begin{bmatrix} \cos\psi_f & -\sin\psi_f \\ \sin\psi_f & \cos\psi_f \end{bmatrix}$$

the rotation matrix from $\mathcal{F}$ to $\mathcal{I}$, parameterized locally by ψ_f. Thus, the error d^I expressed in the Serret-Frenet frame is given by

$$d^{\mathcal{F}} = \begin{bmatrix} es \\ ed \end{bmatrix} = R^T d^{\mathcal{I}} = R^T (p - q(s)) \tag{24}$$

We define the yaw angle error as

$$\tilde{\psi} = \psi - \psi_f \tag{25}$$

Details regarding the calculations involved in obtaining the error kinematic dynamic are omitted in this paper for reasons of brevity, the reader can be referred to [6] for a detailed explanation.

$$\begin{aligned} \dot{e}_s &= V_t \cos\tilde{\psi} - (1 - \mathcal{C}_{\mathcal{C}}(s) e_d)\dot{s} \\ \dot{e}_d &= V_t \sin\tilde{\psi} - \mathcal{C}_{\mathcal{C}}(s) e_s \dot{s} \\ \dot{\tilde{\psi}} &= \omega - \mathcal{C}_{\mathcal{C}}(s)\dot{s} \end{aligned} \tag{26}$$

where $\frac{d\psi_f}{dt} = \mathcal{C}_{\mathcal{C}}(s)$ is the path curvature. The path curvature is expressed as a function of the path coordinates $(x_s(t), y_s(t))$ and its first and second derivatives with respect to the parameter t, i.e $x'_s = \frac{dx_s}{dt}$, $y'_s = \frac{dy_s}{dt}$. Thus, the path curvature $\frac{d\psi_f}{dt} = \mathcal{C}_{\mathcal{C}}(s)$ is given by

$$\mathcal{C}_{\mathcal{C}} = \frac{|y''_s x'_s - y'_s x''_s|}{({x'_s}^2 + {y'_s}^2)^{3/2}} \tag{27}$$

These curvature calculation is given for any path described by the parametric curve $\alpha(s)$, but in the Dubins paths there are only curves of a circle of radius ρ and straight lines. The curvature of a straight line is zero and using 27 we can calculate the curvature of the circle of radius ρ as follows

$$\mathcal{C}_c = \frac{\left|\rho^2 \sin^2\psi + \rho^2 \cos^2\psi\right|}{\left(\rho^2 \sin^2\psi + \rho^2 \cos^2\psi\right)^{3/2}} = \frac{1}{\rho}$$

The kinematic controller was developed by adopting the yaw rate ω from (3) and the particle velocity $\dot{s}$ as virtual controllers as

$$\begin{aligned} \dot{s} &= V_t \cos\tilde{\psi} + k_s e_s \\ \omega &= -\beta - k_{\omega_1}\left(\tilde{\psi} - \delta(e_d)\right) \end{aligned} \tag{28}$$

where k_s, k_{ω_1} are positive real numbers and

$$\begin{aligned} \beta = &-\mathcal{C}_{\mathcal{C}}(s)\dot{s} - \dot{\delta}(e_d)\left(V_t \sin\tilde{\psi} - \mathcal{C}_{\mathcal{C}}(s) e_s \dot{s}\right) \\ &+ (V_t e_d)\left(\frac{\sin\tilde{\psi} - \sin(\delta(e_d))}{\tilde{\psi} - \delta(e_d)}\right) \end{aligned}$$

The sigmoid function $\delta(ed)$ is bounded and differentiable with respect to the error e_d. It provides the desired relative course transition of the fixed-wing MAV to the path as a function of e_d. Moreover, (29) satisfies the condition $e_d \delta(e_d) \leq 0 \;\; \forall e_d$. Such condition guides the MAV to the

correct direction, i.e., turn left when the MAV is on the right side of the path, and turn right in the opposite situation. The function $\delta(ed)$ and its derivative $\dot{\delta}(ed)$ are defined as follows

$$\delta(e_d) = -\psi_a \frac{e^{2k_\delta e_d} - 1}{e^{2k_\delta e_d} + 1} \tag{29}$$

and

$$\dot{\delta}(e_d) = -\frac{4\psi_a k_\delta e^{2k_\delta e_d}}{(e^{2k_\delta e_d} + 1)^2} \tag{30}$$

V. SIMULATION EXAMPLE

Simulations were done on a complete simulation platform, the MAV3DSim(Multi-Aerial Vehicle 3D Simulator) provides a complete 6 degrees of freedom (DoF) computer model of fixed wing aircraft. The MAV3DSim software layers are described briefly in this section. The application scenario is in the use of the path generation and path-following algorithms to command a desired path to the fixed wing UAV. The results from the simulation are presented at the end of this section.

A. MAV3DSim Simulation Platform

The MAV3DSim is a custom C# .Net based application and implements a complete 6DoF nonlinear model. It has a 3D representation to visualiza the position and orientation of the plane, also, it has the capability to load maps directly from Google Maps servers and set the launching site on any location on Earth. The trajectory generated by the plane can be seen on the map, this map is the tangential plane to the Earth.

The data generated by the simulator is coded in the same manner as the common sensors,i.e. it send data emulating an inertial measurement unit(IMU) sending inertial gyroscope, accelerometer and magnetometer, a GPS radio in the latitude/longitude format, altitude and airspeed. It can receive commands to move the control surfaces aileron elevators, rudder, and the thrust of the fixed-wing UAV. The position provided by the simulator is in a standard geodetic WGS84 Latitude(λ), Longitude(τ) and Height(h), and we will use the transformation to the local tangent ENU described in Section II-A.

The software layers, depicted in the Fig 9, are briefly described as follows

1) Path Generator: This layer is in charge of the generation of paths using the Dubins path generation described in Section III. It can generate new paths and maintain the old ones for later use. It is possible to interact online with the path generation and change the course of action of the aircraft in any time either by an autonomous action or by a human interaction. Once the path is fully generated, it is transmitted to the path-following strategy.

2) Path-Following Strategy: The path-following control described in section IV is implemented in this layer. The path is stored in an array of n points of the form (x_m, y_m) starting with $m = 0$ then the path following strategy computes the errors $es, ed, \tilde{\psi}$ from (24) and (25), with this information it

Fig. 9. Communication scheme between the MAV3DSim and the CRRC-Sim.

computes control input ω and $\dot{s}$ from (28). The control ω is a desired heading rate and is induced into the aircraft dynamics through the roll angle using (4). The computed roll angle ϕ will be used by the low level autopilot

3) Low Level Autopilot: The role of low-level autopilot is to stabilize the aircraft in roll and pitch angles, maintain a constant altitude and airspeed by implementing a PD controller for each dynamic (roll, pitch, altitude and airspeed). The altitude and airspeed setpoints are manually introduced by a graphic user interface, the altitude controller outputs the pitch setpoint and the roll setpoint is obtained from the path following controller.

4) Aircraft Dynamics: This layer integrates the set of differential equations representing the aircraft dynamics. The input of this layer are the inputs of the low level autopilot layer and the outputs are the data from the simulated sensors: GPS position, aircraft attitude, airspeed. The aircraft dynamics layer sends the outputs to the upper layers.

B. Simulation Scenario

We use the MAV3DSim simulation platform along with the Dubins path generator and the path-following strategy previously described to present a simulation scenario. The description of the scenario is as follows: A person is missing and is located somewhere in a known area. The main task of the UAV is to find this person, so it will sweep this area in order to find the missing person. First we need to define the search area as a rectangle with the aid of a user interface, then using the proposed path generator algorithm define the a path

Fig. 10. The path generated to weep the search area.

Fig. 11. Circular path generated to surround the missing person.

for sweeping the rectangle area. The starting point of the path will be one of the corners of the rectangle and it selects the closest to the current position of the UAV as depicted in Fig. 11. The UAV will travel along the path until it is sufficiently close to the lost person (red dot in Fig.11). When the missing person is found a circular path is generated to surround the missing person. The simulation can be seen in `https://www.youtube.com/watch?v=_AUW8_g-jb0`

VI. CONCLUSIONS AND FUTURE WORK

The presented study attempts to propose a framework as a part of a more general project in which more tasks are needed in order to complete a predefined task without human interaction. Some of these tasks are trajectory generation as a function of different variables as time, user commands, visual marks, energy consumption or sensor information, just to mention a few. Dubins paths have been utilized as a tool to estimate the shortest path from the current MAV pose (attitude and position) to a given point provided by the user. Despite the fact that the proposed controller has been designed taking into consideration a simplified aircraft model, the controller performance is proved on the MAV3Dsim simulator by using a full six degree-of-freedom aerodynamic model. Besides these developments, conducing experiments on a platform is indispensable to consolidate the results of the presented study w.r.t claims of modeling simplifications and performance.

REFERENCES

[1] L.E.Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, jul 1957.

[2] J.A.Reeds and R.A.Sheep, "Optimal paths for a car that goes both fordward and backward," *Pacific Journal of Mathematics*, vol. 2, pp. 367–393, 1990.

[3] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces." *IEEE T. Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[4] Y. Kuwata, A. Richards, T. Schouwenaars, and J. P. How, "Decentralized robust receding horizon control for multi-vehicle guidance," in *American Control Conference (ACC)*, Minneapolis, MN, June 2006, pp. 2047–2052.

[5] T. Mehta and M. Egerstedt, "An optimal control approach to mode generation in hybrid systems." *Nonlinear Analysis: Theory, Methods and Applications*, vol. 65, no. 5, pp. 963–983, Sept 2006.

[6] G. Flores, I. Lugo-Cardenas, and R. Lozano, "A nonlinear path-following strategy for a fixed-wing mav," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, Grand Hyatt Atlanta, Atlanta, GA, May, 2013, pp. 1014–1021.

[7] J. Farrel and M. Barth, *The Global Positioning System and Inertial Navigation*. Hoboken, NJ, USA: John Wiley and Sons, 2007.

[8] Anonymous, "World geodetic system 1984 (wgs-84)its definition and relationships with local geoderic systems," Defense Mapping Agency, Tech. Rep. NIMA TR 8350.2, 2000.

[9] A. Micaelli and C. Samson, "Trajectory tracking for unicycle-type and two-steering-wheels mobile robots," INRIA, Tech. Rep. 2097, 1993.