Implementing Dubins Airplane Paths on Fixed-wing UAVs *

Randal W. Beard   Timothy W. McLain
Brigham Young University
Provo, UT 84602
beard@byu.edu, mclain@byu.edu

July 8, 2013

Abstract

A well known path-planning technique for mobile robots or planar aerial vehicles is to use Dubins paths, which are minimum-distance paths between two configurations subject to the constraints of the Dubins car model. An extension of this method to a three-dimensional Dubins airplane model has recently been proposed. This chapter builds on that work showing a complete architecture for implementing Dubins airplane paths on small fixed-wing UAVs. The existing Dubins airplane model is modified to be more consistent with the kinematics of a fixed-wing aircraft. The chapter then shows how a recently proposed vector-field method can be used to design a guidance law that causes the Dubins airplane model to follow straight-line and helical paths. Dubins airplane paths are move complicated than Dubins car paths because of the altitude component. Based on the difference between the altitude of the start and end configurations, Dubins airplane paths can be classified as low, medium, or high altitude gain. While for medium and high altitude gain there are many different Dubins airplane paths, this chapter proposes selecting the path that maximizes the average altitude throughout the maneuver. The proposed architecture is implemented on a six degree-of-freedom Matlab/Simulink simulation of an Aerosonde UAV, and results from this simulation demonstrate the effectiveness of the technique.

1   Introduction

Unmanned aerial vehicles (UAVs) are used for a wide variety of tasks that require the UAV to be flown from one particular pose (position and attitude) to another. Most commonly, UAVs are flown from their current position and heading angle to a new desired position and heading angle. The ability to fly

---

from one pose (or waypoint) to another is the fundamental building block upon which more sophisticated UAV navigation capabilities are built. For UAV missions involving sensors, the ability to position and orient the sensor over time is critically important. Example applications include wildlife observation and tracking, infrastructure monitoring [Rathinam et al., 2005, Few et al., 2004, Egbert and Beard, 2011], communication relays [Frew et al., 2009], meteorological measurements [Elston et al., 2010], and aerial surveillance [Rahmani et al., 2010, Elston and Frew, 2008, Spry et al., 2005]. Positioning and orienting the sensor is accomplished in part by planning and following paths through or above the sensing domain. Two-dimensional (2D) path planning and following at a constant altitude through unobstructed airspace is common, but as mission scenarios become increasingly sophisticated by requiring flight in three-dimensional (3D) terrain, the need for full 3D planning and guidance algorithms is becoming increasingly important.

For a vehicle that moves in a 2D plane at constant forward speed with a finite turn-rate constraint, the minimum distance path between two configurations is termed a Dubins path. The initial and final configurations are defined by a 2D position in the plane of motion and an orientation. It has been shown that the optimal Dubins path in the absence of wind is composed of a constant radius turn, followed by a straight-line path, followed by another constant radius turn [Dubins, 1957]. A vehicle that follows Dubins paths is often termed a Dubins car. There have been a wide variety of path planning techniques proposed for mobile robots based on the Dubins car model [Hanson et al., 2011, Balluchi et al., 1996, Anderson and Milutinovic, 2011, Karaman and Frazzoli, 2010, Cowlagi and Tsiotras, 2009, Yong and Barth, 2006]. The Dubins car model has also been used extensively for UAV applications by constraining the air vehicle to fly at a constant altitude [Yu and Beard, 2013, Sujit et al., 2007, Yang and Kapila, 2002, Shima et al., 2007].

The Dubins car model was recently extended to three dimensions to create the Dubins airplane model, where in addition to turn-rate constraints, a climb-rate constraint was added [Hosak and Ghose, 2010, Chitsaz and LaValle, 2007, Rahmani et al., 2010]. Minimum distance paths for the Dubins airplane were also derived in [Chitsaz and LaValle, 2007], using the Pontryagin minimum principle [Lewis, 1986]. However, in [Chitsaz and LaValle, 2007] some practical considerations were not considered, leaving a gap between the theory and implementation on actual UAVs. The purpose of this chapter is to fill that gap. In particular, alternative equations of motion for the Dubins airplane model that include airspeed, flight-path angle, and bank angle are given. The kinematic equations of motion presented in this chapter are standard in the aerospace literature. The chapter also describe how to implement Dubins airplane paths using low-level autopilot loops, vector-field guidance laws for following straight lines and helices, and mode switching between the guidance laws.

In addition to the complexity of a third dimension of motion, Dubins airplane paths are more complicated than Dubins car paths. In particular, when the altitude component of the path falls within a specific range, there are an infinite number of paths that satisfy the minimum-distance objective. This chapter

2

also proposes specific choices for paths that make practical sense for many UAV mission scenarios. Specifically, the path that also maximizes the average altitude of the path is selected.

The software architecture proposed in this chapter is similar to that discussed in [Beard and McLain, 2012] and is shown in Figure 1. At the lowest level is the fixed-wing UAV. A state estimator processes sensors and produces the estimates of the state required for each of the higher levels. A low-level autopilot accepts airspeed, flight-path angle, and bank angle commands. The commands for the low-level autopilot are produced by a vector-field guidance law for following either straight-line paths or helical paths. The minimum distance Dubins airplane path between two configurations is computed by the path manager, which also switches between commanding straight-line paths and helical paths.

Configuration Waypoints:
(position, heading)

Path definition:
straight-line, or
helix

Dubins Path Manager

Tracking error

Vector Field Guidance

Airspeed
Flight path angle
Bank angle

Position error

Low Level Autopilot

Throttle
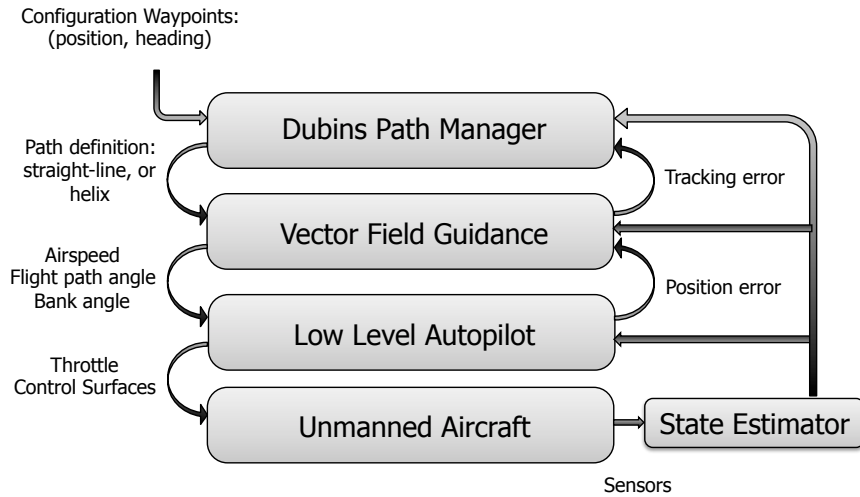Control Surfaces

Unmanned Aircraft

State Estimator

Sensors

Figure 1: Flight control architecture proposed in this chapter.

This chapter will not discuss state estimation using the available sensors, but the interested reader is refered to [Beard and McLain, 2012] and other publications on state estimation for UAVs (see for example [Mahony et al., 2008, Misawa and Hedrick, 1989, Brunke and Campbell, 2004]). Section 2 briefly describes assumptions on the unmanned aircraft and how the low-level autopilot is configured to produce the Dubins airplane kinematic model. Section 3 describes the vector-field guidance strategy used in this chapter for following both straight lines and helices. The Dubins airplane paths and the path manager used to follow them are discussed in Section 4. Section 5 offer concluding remarks.

2   Equations of Motion for the Dubins Airplane

Unmanned aircraft, particularly smaller systems, fly at relatively low airspeeds causing wind to have a significant effect on their performance. Since wind

effects are not known prior to the moment they act on an aircraft, they are typically treated as a disturbance to be rejected in real time by the flight control system, rather than being considered during the path planning. It has been shown that vector-field-based path following methods, such as those employed in this chapter, are particularly effective at rejecting wind disturbances [Nelson et al., 2007]. Treating wind as a disturbance also allows paths to be planned relative to the inertial environment, which is important as UAVs are flown in complex 3D terrain. Accordingly, when the Dubins airplane model is used for path planning, the effects of wind are not accounted for when formulating the equations of motion. In this case, the airspeed V is the same as the groundspeed, the heading angle ψ is the same as the course angle (assuming zero sideslip angle), and the flight-path angle γ is the same as the air-mass-referenced flight-path angle [Beard and McLain, 2012].

Figure 2 depicts a UAV flying with airspeed V, heading angle ψ and flight-path angle γ. Denoting the inertial position of the UAV as $(r_n, r_e, r_d)^\top$, the kinematic relationship between the inertial velocity, $\mathbf{v} = (\dot{r}_n, \dot{r}_e, \dot{r}_d)^\top$, and the airspeed, heading angle, and flight-path angle can be easily visualized as

$$\begin{pmatrix} \dot{r}_n \\ \dot{r}_e \\ \dot{r}_d \end{pmatrix} = \begin{pmatrix} V \cos\psi \cos\gamma \\ V \sin\psi \cos\gamma \\ -V \sin\gamma \end{pmatrix},$$

where $V = \|\mathbf{v}\|$.

This chapter assumes that a low-level autopilot regulates the airspeed V to a constant commanded value V^c, the flight-path angle γ to the commanded flight-path angle γ^c, and the bank angle ϕ to the commanded bank angle ϕ^c. In addition, the dynamics of the flight-path angle and bank angle loops are assumed to be sufficiently fast that they can be ignored for the purpose of path following. The relationship between the heading angle ψ and the bank angle ϕ is given by the coordinated turn condition [Beard and McLain, 2012]

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

where g is the acceleration due to gravity.

Under the assumption that the autopilot is well tuned and the airspeed, flight-path angle, and bank angle states converge with the desired response to their commanded values, then the following kinematic model is a good description of the UAV motion

$$\begin{aligned}
\dot{r}_n &= V \cos\psi \cos\gamma^c \\
\dot{r}_e &= V \sin\psi \cos\gamma^c \\
\dot{r}_d &= -V \sin\gamma^c \\
\dot{\psi} &= \frac{g}{V} \tan\phi^c.
\end{aligned} \tag{1}$$

Physical capabilities of the aircraft place limits on the achievable bank and flight-path angles that can be commanded. These physical limits on the aircraft
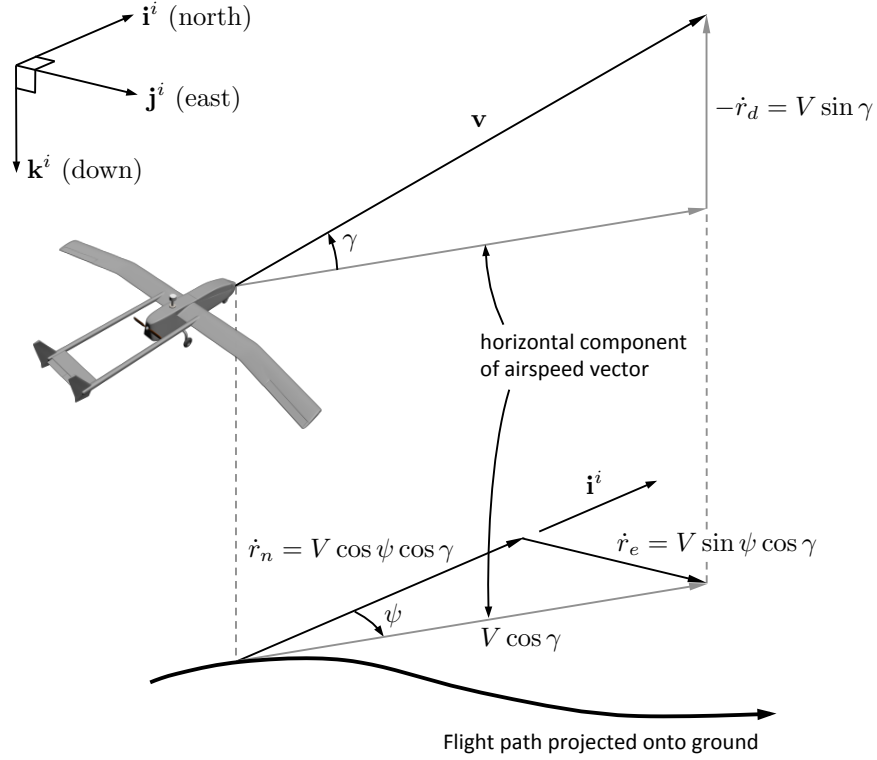
Figure 2: Graphical representation of aircraft kinematic model.

are represented by the following constraints

$$|\phi^c| \leq \bar{\phi} \tag{2}$$
$$|\gamma^c| \leq \bar{\gamma}. \tag{3}$$

The kinematic model given by (1) with the input constraints (2) and (3) will be referred to as the Dubins airplane. This model builds upon the model originally proposed for the Dubins airplane in [Chitsaz and LaValle, 2007], which is given by

$$
\begin{aligned}
\dot{r}_n &= V \cos \psi \\
\dot{r}_e &= V \sin \psi \\
\dot{r}_d &= u_1 \quad |u_1| \leq 1 \\
\dot{\psi} &= u_2 \quad |u_2| \leq 1.
\end{aligned}
\tag{4}
$$

Although (1) is similar to (4), it captures the aircraft kinematics with greater accuracy and provides greater insight into the aircraft behavior, and is more consistent with commonly used aircraft guidance models. Note however, that (1)

5

is only a kinematic model that does not include aerodynamics, wind effects, or engine/thrust limits. While it is not sufficiently accurate for low-level autopilot design, it is well suited for for high level path planning and path following control design. In-depth discussions of aircraft dynamic models can be found in [Phillips, 2004, Stevens and Lewis, 2003, Nelson, 1998, Yechout et al., 2003].

3   3D Vector-field Path Following

This section shows how to develop guidance laws to ensure that the kinematic model (1) follows straight-line and helical paths. Section 4 shows how straight-line and helical paths are combined to produce minimum-distance paths between start and end configurations.

3.1   Vector-field Methology

The guidance strategy will use the vector-field methodology proposed in [Goncalves et al., 2010], and this section provides a brief overview. The path to be followed in $\mathbb{R}^3$ is specified as the intersection of two two-dimensional manifolds given by $\alpha_1(\mathbf{r}) = 0$ and $\alpha_2(\mathbf{r}) = 0$ where α_1 and α_2 have bounded second partial derivatives, and where $\mathbf{r} \in \mathbb{R}^3$. An underlying assumption is that the path given by the intersection is connected and one-dimensional. Defining the function

$$V(\mathbf{r}) = \frac{1}{2}\alpha_1^2(\mathbf{r}) + \frac{1}{2}\alpha_2^2(\mathbf{r}),$$

gives

$$\frac{\partial V}{\partial \mathbf{r}} = \alpha_1(\mathbf{r})\frac{\partial \alpha_1}{\partial \mathbf{r}}(\mathbf{r}) + \alpha_2(\mathbf{r})\frac{\partial \alpha_2}{\partial \mathbf{r}}(\mathbf{r}).$$

Note that $-\frac{\partial V}{\partial \mathbf{r}}$ is a vector that points toward the path. Therefore following $-\frac{\partial V}{\partial \mathbf{r}}$ will transition the Dubins airplane onto the path. However simply following $-\frac{\partial V}{\partial \mathbf{r}}$ is insufficient since the gradient is zero on the path. When the Dubins airplane is on the path, its direction of motion should be perpendicular to both $\frac{\partial \alpha_1}{\partial \mathbf{r}}$ and $\frac{\partial \alpha_2}{\partial \mathbf{r}}$. Following [Goncalves et al., 2010] the desired velocity vector $\mathbf{u}' \in \mathbb{R}^3$ can be chosen as

$$\mathbf{u}' = -K_1\frac{\partial V}{\partial \mathbf{r}} + K_2\frac{\partial \alpha_1}{\partial \mathbf{r}} \times \frac{\partial \alpha_2}{\partial \mathbf{r}}, \tag{5}$$

where K_1 and K_2 are symmetric tuning matrices. It is shown in [Goncalves et al., 2010] that the dynamics $\dot{\mathbf{r}} = \mathbf{u}'$ where $\mathbf{u}'$ is given by Equation (5), results in $\mathbf{r}$ asymptotically converging to a trajectory that follows the intersection of α_1 and α_2 if K_1 is positive definite, and where the definiteness of K_2 determines the direction of travel along the trajectory.

The problem with Equation (5) is that the magnitude of the desired velocity $\mathbf{u}'$ may not equal V, the velocity of the Dubin's airplane. Therefore $\mathbf{u}'$ is normalized as

$$\mathbf{u} = V\frac{\mathbf{u}'}{\|\mathbf{u}'\|}. \tag{6}$$

Fortunately, the stability proof in [Goncalves et al., 2010] is still valid when $\mathbf{u}'$ is normalized as in Equation (6).

Setting the NED components of the velocity of the Dubins airplane model given in Equation (1) to $\mathbf{u} = (u_1, u_2, u_3)^\top$ gives

$$V \cos \psi^d \cos \gamma^c = u_1$$
$$V \sin \psi^d \cos \gamma^c = u_2$$
$$-V \sin \gamma^c = u_3.$$

Solving for the commanded flight-path angle γ^c, and the desired heading angle ψ^d results in the expressions

$$\gamma^c = -\text{sat}_{\bar{\gamma}} \left[\sin^{-1} \left(\frac{u_3}{V} \right) \right] \tag{7}$$
$$\psi^d = \text{atan2}(u_2, u_1),$$

where atan2 is the four quadrant inverse tangent, and where the saturation function is defined as

$$\text{sat}_a[x] = \begin{cases} a & \text{if } x \geq a \\ -a & \text{if } x \leq -a \\ x & \text{otherwise} \end{cases} .$$

Assuming the inner-loop lateral-directional dynamics are accurately modeled by the coordinated-turn equation, roll-angle commands yielding desirable turn performance can be obtained from the expression

$$\phi^c = \text{sat}_{\bar{\phi}} \left[k_\phi (\psi^d - \psi) \right], \tag{8}$$

where k_ϕ is a positive constant.

Sections 3.2 and 3.3 applies the framework described in this section to straight-line following and helix following, respectively.

3.2 Straight-line Paths

A straight-line path is described by the direction of the line and a point on the line. Let $\mathbf{c}_\ell = (c_n, c_e, c_d)^\top$ be an arbitrary point on the line, and let the direction of the line be given by the desired heading angle from north ψ_ℓ, and the desired flight-path angle γ_ℓ measured from the inertial north-east plane. Therefore

$$\mathbf{q}_\ell = \begin{pmatrix} q_n \\ q_e \\ q_d \end{pmatrix} \triangleq \begin{pmatrix} \cos \psi_\ell \cos \gamma_\ell \\ \sin \psi_\ell \cos \gamma_\ell \\ -\sin \gamma_\ell \end{pmatrix}$$

is a unit vector that points in the direction of the desired line. The straight-line path is given by

$$\mathcal{P}_{\text{line}}(\mathbf{c}_\ell, \psi_\ell, \gamma_\ell) = \left\{ \mathbf{r} \in \mathbb{R}^3 : \mathbf{r} = \mathbf{c}_\ell + \sigma \mathbf{q}_\ell, \sigma \in \mathbb{R} \right\}. \tag{9}$$

A unit vector that is perpendicular to the longitudinal plane defined by $\mathbf{q}_\ell$ is given by

$$\mathbf{n}_{\text{lon}} \triangleq \begin{pmatrix} -\sin\psi_\ell \\ \cos\psi_\ell \\ 0 \end{pmatrix}.$$

Similarly, a unit vector that is perpendicular to the lateral plane defined by $\mathbf{q}_\ell$ is given by

$$\mathbf{n}_{\text{lat}} \triangleq \mathbf{n}_{\text{lon}} \times \mathbf{q}_\ell = \begin{pmatrix} -\cos\psi_\ell \sin\gamma_\ell \\ -\sin\psi_\ell \sin\gamma_\ell \\ -\cos\gamma_\ell \end{pmatrix}.$$

It follows that $\mathcal{P}_{\text{line}}$ is given by the intersection of the surfaces defined by

$$\alpha_{\text{lon}}(\mathbf{r}) \triangleq \mathbf{n}_{\text{lon}}^\top (\mathbf{r} - \mathbf{c}_\ell) = 0 \tag{10}$$

$$\alpha_{\text{lat}}(\mathbf{r}) \triangleq \mathbf{n}_{\text{lat}}^\top (\mathbf{r} - \mathbf{c}_\ell) = 0. \tag{11}$$

Figure 3 shows $\mathbf{q}_\ell$, $\mathbf{c}_\ell$, and the surfaces defined by $\alpha_{\text{lon}}(\mathbf{r}) = 0$ and $\alpha_{\text{lat}}(\mathbf{r}) = 0$.
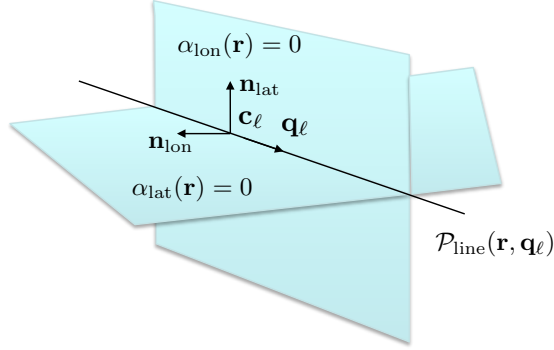

Figure 3: This figure shows how the straight-line path $\mathcal{P}_{\text{line}}(\mathbf{c}_\ell, \psi_\ell, \gamma_\ell)$ is defined by the intersection of the two surfaces given by $\alpha_{\text{lon}}(\mathbf{r}) = 0$ and $\alpha_{\text{lat}}(\mathbf{r}) = 0$.

The gradients of α_{lon} and α_{lat} are given by

$$\frac{\partial \alpha_{\text{lon}}}{\partial \mathbf{r}} = \mathbf{n}_{\text{lon}}$$

$$\frac{\partial \alpha_{\text{lat}}}{\partial \mathbf{r}} = \mathbf{n}_{\text{lat}}.$$

Therefore, before normalization, the desired velocity vector is given by

$$\mathbf{u}'_{\text{line}} = K_1 \left(\mathbf{n}_{\text{lon}} \mathbf{n}_{\text{lon}}^\top + \mathbf{n}_{\text{lat}} \mathbf{n}_{\text{lat}}^\top \right) (\mathbf{r} - \mathbf{c}_\ell) + K_2 \left(\mathbf{n}_{\text{lon}} \times \mathbf{n}_{\text{lat}} \right). \tag{12}$$

3.3 Helical Paths

A time parameterized helical path is given by

$$\mathbf{r}(t) = \mathbf{c}_h + \begin{pmatrix} R_h \cos(\lambda_h t + \psi_h) \\ R_h \sin(\lambda_h t + \psi_h) \\ -t R_h \tan \gamma_h \end{pmatrix}, \tag{13}$$

where $\mathbf{r}(t) = \begin{pmatrix} r_n \\ r_e \\ r_d \end{pmatrix}(t)$ is the position along the path, $\mathbf{c}_h = (c_n, c_e, c_d)^\top$ is the center of the helix, and the initial position of the helix is

$$\mathbf{r}(0) = \mathbf{c}_h + \begin{pmatrix} R_h \cos \psi_h \\ R_h \sin \psi_h \\ 0 \end{pmatrix},$$

and where R_h is the radius, $\lambda_h = +1$ denotes a clockwise helix ($N{\to}E{\to}S{\to}W$), and $\lambda_h = -1$ denotes a counter-clockwise helix ($N{\to}W{\to}S{\to}E$), and where γ_h is the desired flight-path angle along the helix.

To find two surfaces that define the helical path, the time parameterization in (13) needs to be eliminated. Equation (13) gives

$$(r_n - c_n)^2 + (r_e - c_e)^2 = R_h^2.$$

In addition, divide the east component of $\mathbf{r} - \mathbf{c}_h$ by the north component to get

$$\tan(\lambda_h t + \psi_h) = \frac{r_e - c_e}{r_n - c_n}$$

Solving for t and plugging into the third component of (13) gives

$$r_d - c_d = -\frac{R_h \tan \gamma_h}{\lambda_h} \left(\tan^{-1} \left(\frac{r_e - c_e}{r_n - c_n} \right) - \psi_h \right).$$

Therefore, normalizing these equations by R_h results in

$$\alpha_{\text{cyl}}(\mathbf{r}) = \left(\frac{r_n - c_n}{R_h} \right)^2 + \left(\frac{r_e - c_e}{R_h} \right)^2 - 1$$

$$\alpha_{\text{pl}}(\mathbf{r}) = \left(\frac{r_d - c_d}{R_h} \right) + \frac{\tan \gamma_h}{\lambda_h} \left(\tan^{-1} \left(\frac{r_e - c_e}{r_n - c_n} \right) - \psi_h \right).$$

Normalization by R_h makes the gains on the resulting control strategy invariant to the size of the orbit.

A helical path is then defined as

$$\mathcal{P}_{\text{helix}}(\mathbf{c}_h, \psi_h, \lambda_h, R_h, \gamma_h) = \{\mathbf{r} \in \mathbb{R}^3 : \alpha_{\text{cyl}}(\mathbf{r}) = 0 \text{ and } \alpha_{\text{pl}}(\mathbf{r}) = 0\}. \tag{14}$$

The two surfaces $\alpha_{\text{cyl}}(\mathbf{r}) = 0$ and $\alpha_{\text{pl}}(\mathbf{r}) = 0$ are shown in Figure 4 for parameters $\mathbf{c}_h = (0, 0, 0)^\top$, $R_h = 30$ m, $\gamma_h = \frac{15\pi}{180}$ rad, and $\lambda_h = +1$. The associated helical path is the intersection of the two surfaces.
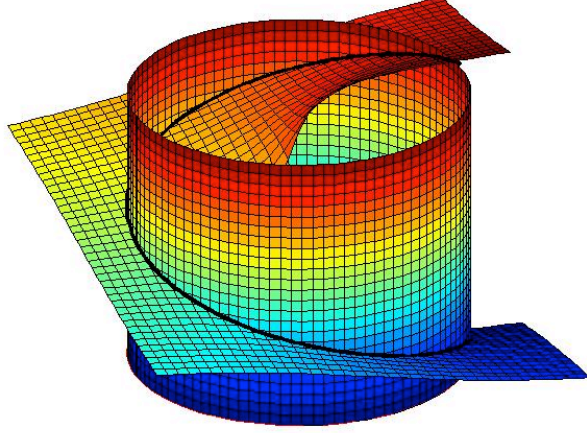
Figure 4: A helical path for parameters $\mathbf{c}_h = (0, 0, 0)^\top$, $R_h = 30$ m, $\gamma_h = \frac{15\pi}{180}$ rad, and $\lambda_h = +1$.

The gradients of α_{cyl} and α_{pl} are given by

$$\frac{\partial \alpha_{\text{cyl}}}{\partial \mathbf{r}} = \begin{pmatrix} 2\frac{r_n - c_n}{R_h}, & 2\frac{r_e - c_e}{R_h}, & 0 \end{pmatrix}^\top$$

$$\frac{\partial \alpha_{\text{pl}}}{\partial \mathbf{r}} = \begin{pmatrix} \frac{\tan \gamma_h}{\lambda_h} \frac{-(r_e - c_e)}{(r_n - c_n)^2 + (r_e - c_e)^2}, & \frac{\tan \gamma_h}{\lambda_h} \frac{(r_n - c_e)}{(r_n - c_n)^2 + (r_e - c_e)^2}, & \frac{1}{R_h} \end{pmatrix}^\top.$$

Before normalization, the desired velocity vector is given by

$$\mathbf{u}'_{\text{helix}} = K_1 \left(\alpha_{\text{cyl}} \frac{\partial \alpha_{\text{cyl}}}{\partial \mathbf{r}} + \alpha_{\text{pl}} \frac{\partial \alpha_{\text{pl}}}{\partial \mathbf{r}} \right) + \lambda K_2 \left(\frac{\partial \alpha_{\text{cyl}}}{\partial \mathbf{r}} \times \frac{\partial \alpha_{\text{pl}}}{\partial \mathbf{r}} \right), \qquad (15)$$

where

$$\frac{\partial \alpha_{\text{cyl}}}{\partial \mathbf{r}} \times \frac{\partial \alpha_{\text{pl}}}{\partial \mathbf{r}} = \frac{2}{R_h} \begin{pmatrix} \frac{r_e - c_e}{R_h}, & -\frac{r_n - c_n}{R_h}, & \lambda_h \tan \gamma_h \end{pmatrix}^\top.$$

4 Minimum Distance Airplane Paths

This section describes how to concatenate straight-line and helix paths to produce minimum-distance paths between two configurations for the Dubins airplane model. A configuration is defined as the tuple (z_n, z_e, z_d, ψ), where $(z_n, z_e, z_d)^\top$ is a north-east-down position referenced to an inertial frame, and ψ is a heading angle measured from north. Given the kinematic model (1) subject to the constraints (2) and (3), a Dubins airplane path refers to a minimum-distance path between a start configuration $(z_{ns}, z_{es}, z_{ds}, \psi_s)$ and an end configuration $(z_{ne}, z_{ee}, z_{de}, \psi_e)$. Minimum-distance paths for the Dubins airplane are derived in [Chitsaz and LaValle, 2007] using the Pontryagin Maximum Principle for the dynamics given in (4) with constraints $\bar{\gamma} = 1$ and $\bar{\phi} = 1$. This section

recasts the results from [Chitsaz and LaValle, 2007] using the standard aircraft kinematic model given in (1) using the constraints (2) and (3).

4.1 Dubins Car Paths

Minimum-distance paths for the Dubins airplane are closely related to minimum-distance paths for the Dubins car. This section briefly reviews Dubins car paths, which were originally developed in [Dubins, 1957] using the notation and methods defined in [Beard and McLain, 2012].

The Dubins car model is a subset of (4) given by

$$
\begin{aligned}
\dot{r}_n &= V \cos \psi \\
\dot{r}_e &= V \sin \psi \\
\dot{\psi} &= u,
\end{aligned}
\tag{16}
$$

where $|u| \leq \bar{u}$. For the Dubins car, the minimum turn radius is given by

$$
R_{\min} = V/\bar{u}.
\tag{17}
$$

The Dubins car path is defined as the minimum-distance path from the start configuration (z_{ns}, z_{es}, ψ_s) to the end configuration (z_{ne}, z_{ee}, ψ_e). As shown in [Dubins, 1957], the minimum-distance path between two different configurations consists of a circular arc of radius $R_{\min}$ that starts at the initial configuration, followed by a straight line, and concluding with another circular arc of radius $R_{\min}$ that ends at the final configuration.

As shown in Figure 5, for any given start and end configurations, there are four possible paths consisting of an arc, followed by a straight line, followed by an arc. RSR is a right-handed arc followed by a straight line followed by another right-handed arc. RSL is a right-handed arc followed by a straight line followed by a left-handed arc. LSR is a left-handed arc followed by a straight line followed by a right-handed arc. LSL is a left-handed arc followed by a straight line followed by another left-handed arc. The Dubins path is defined as the case with the shortest path length.

As explained in [Beard and McLain, 2012], the guidance algorithm for following a Dubins car path consists of switching between orbit following and straight-line following. Figure 6 shows the parameters that are required by the guidance algorithm to follow a Dubins car path. Given that the vehicle configuration is close to the start configuration $(\mathbf{z}_s, \psi_s)$, the vehicle is initially commanded to follow an orbit with center $\mathbf{c}_s$ and orbit direction λ_s. The orbit is followed until the vehicle crosses half-plane $\mathcal{H}_s(\mathbf{w}_s, \mathbf{q}_s)$, or in other words until its position $\mathbf{r}$ satisfies

$$
(\mathbf{r} - \mathbf{w}_s)^\top \mathbf{q}_s \geq 0,
$$

where $\mathbf{w}_s$ is a position on the half-plane and $\mathbf{q}_s$ is a unit vector orthogonal to the half-plane. The vehicle then follows the straight line defined by $(\mathbf{w}_s, \mathbf{q}_s)$ until it crosses half-plane $\mathcal{H}_\ell(\mathbf{w}_\ell, \mathbf{q}_\ell)$. It then follows the orbit with center $\mathbf{c}_e$ and direction λ_e until it crosses half-plane $\mathcal{H}_e(\mathbf{w}_e, \mathbf{q}_e)$ and completes the Dubins
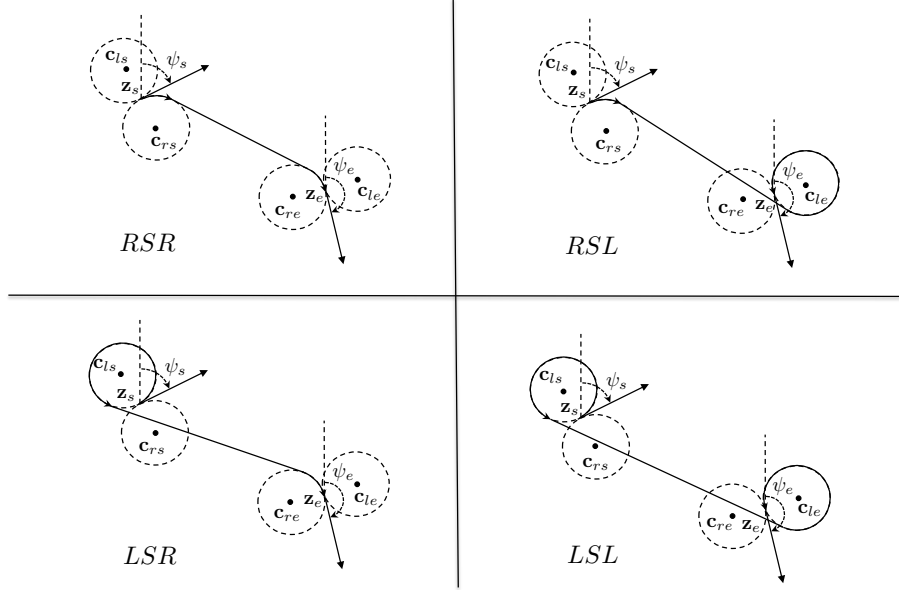
Figure 5: Given a start configuration (z_{ns}, z_{es}, ψ_s), an end configuration (z_{ne}, z_{ee}, ψ_e), and a radius R, there are four possible paths consisting of an arc, a straight line, and an arc. The Dubins path is defined as the case that results in the shortest path length, which for this scenario is RSR.
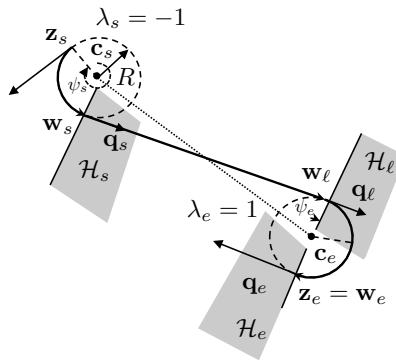

Figure 6: The parameters that are required by the guidance algorithm to follow a Dubins car path.

car path. Accordingly the parameters that define a Dubins car path are given by

$$\mathcal{D}_{\mathrm{car}} = (R, \mathbf{c}_s, \lambda_s, \mathbf{w}_s, \mathbf{q}_s, \mathbf{w}_\ell, \mathbf{q}_\ell, \mathbf{c}_e, \lambda_e, \mathbf{w}_e, \mathbf{q}_e). \tag{18}$$

The length of the Dubins car path depends explicitly on the turning radius R and will be denoted as $L_{\mathrm{car}}(R)$. Details of how to compute $L_{\mathrm{car}}(R)$ as well as the parameters $\mathcal{D}_{\mathrm{car}}$ are given in [Beard and McLain, 2012].

4.2 Dubins Airplane Paths

Dubins airplane paths are more complicated than Dubins car paths because of the altitude component. As described in [Chitsaz and LaValle, 2007] there are three different cases for Dubins airplane paths that depend on the altitude difference between the start and end configuration, the length of the Dubins car path, and the flight-path limit $\bar{\gamma}$. The three cases are defined in [Chitsaz and LaValle, 2007] to be *low altitude*, *medium altitude*, and *high altitude*. In contrast to (17), the minimum turn radius for a Dubins airplane is given by

$$R_{\min} = \frac{V^2}{g} \tan\bar{\phi}. \tag{19}$$

The altitude gain between the start and end configuration is said to be *low altitude* if

$$|z_{de} - z_{ds}| \le L_{\mathrm{car}}(R_{\min}) \tan\bar{\gamma},$$

where the term on the right is the maximum altitude gain that can be obtained by flying at flight-path angle $\pm\bar{\gamma}$ for a distance of $L_{\mathrm{car}}(R_{\min})$. The altitude gain is said to be *medium altitude* if

$$L_{\mathrm{car}}(R_{\min}) \tan\bar{\gamma} < |z_{de} - z_{ds}| \le [L_{\mathrm{car}}(R_{\min}) + 2\pi R_{\min}] \tan\bar{\gamma},$$

where the addition of the term $2\pi R_{\min}$ accounts for adding one orbit at radius $R_{\min}$ to the path length. The altitude gain is said to be *high altitude* if

$$|z_{de} - z_{ds}| > [L_{\mathrm{car}}(R_{\min}) + 2\pi R_{\min}] \tan\bar{\gamma}.$$

The following three sections describe how Dubins car paths are modified to produce Dubins airplane paths for low, high, and medium-altitude cases.

4.2.1 Low-altitude Dubins Paths

In the low-altitude case, the altitude gain between the start and end configurations can be achieved by flying the Dubins car path with a flight-path angle satisfying constaint (3). Therefore, the optimal flight-path angle can be computed by

$$\gamma^* = \tan^{-1}\left(\frac{|z_{de} - z_{ds}|}{L_{\mathrm{car}}(R_{\min})}\right).$$

The length of the Dubins airplane path is given by

$$L_{\mathrm{air}}(R_{\min}, \gamma^*) = \frac{L_{\mathrm{car}}(R_{\min})}{\cos \gamma^*}.$$

The parameters required to define a low altitude Dubins airplane path, are the same parameters for the Dubins car given in (18) with the addition of the optimal flight-path angle γ^*, and the angles of the start and end helices ψ_s and ψ_e. Note that for the Dubins car path ψ_s and ψ_e are not required since the orbit is flat and does not have a starting location. However, as described in Section 3.3, to follow a helix, the start angle is required. Figure 7 shows several Dubins airplane paths for the low-altitude case where the altitude difference is 25 meters over a typical Dubins car path length of 180 meters.
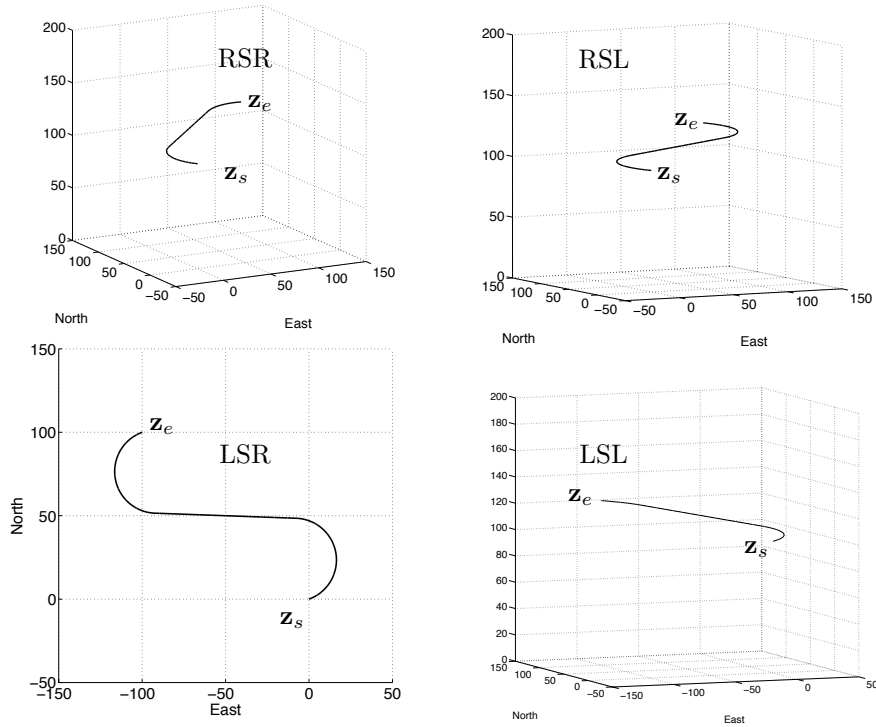
Figure 7: Dubins airplane paths for the low-altitude case.

4.2.2 High-altitude Dubins Paths

In the high-altitude case, the altitude gain cannot be achieved by flying the Dubins car path within the flight-path angle constraints. As shown in [Chitsaz and LaValle, 2007], the minimum distance path is achieved when the flight-path angle is set at its limit of $\pm\bar{\gamma}$, and the Dubins car path is extended to facilitate

the altitude gain. While there are many different ways to extend the Dubins car path, this chapter extends the path by spiraling a certain number of turns at the beginning or end of the path, and then by increasing the turn radius by the appropriate amount.

For UAV scenarios, the most judicious strategy is typically to spend most of the trajectory at as high an altitude as possible. Therefore, if the altitude at the end configuration is higher than the altitude at the start configuration, then the path will be extended by a climbing helix at the beginning of the path, as shown in the RSR and RSL cases in Figure 8. If on the other hand, the altitude at the start configuration is higher than the end configuration, then the path will be extended by a descending helix at the end of the path, as shown in the LSR and LSL cases in Figure 8. If multiple turns around the helix are required, then the turns could be split between the start and end helices and still result in the same path length. For high altitude Dubins paths, the required number of turns in the helix will be the smallest integer k such that

$$(L_{\mathrm{car}}(R_{\min}) + 2\pi k R_{\min}) \tan \bar{\gamma} \leq |z_{de} - z_{ds}| < (L_{\mathrm{car}}(R_{\min}) + 2\pi(k+1)R_{\min}) \tan \bar{\gamma},$$

or in other words

$$k = \left\lfloor \frac{1}{2\pi R_{\min}} \left(\frac{|z_{de} - z_{ds}|}{\tan \bar{\gamma}} - L_{\mathrm{car}}(R_{\min}) \right) \right\rfloor,$$

where $\lfloor x \rfloor$ is the floor function that rounds x down to the nearest integer. The radius of the start and end helices is then increased to R^* so that

$$(L_{\mathrm{car}}(R^*) + 2\pi k R^*) \tan \bar{\gamma} = |z_{de} - z_{ds}|. \tag{20}$$

A bisection search is used to find R^* satisfying (20). The resulting path is a minimum distance Dubins airplane path with path length

$$L_{\mathrm{air}}(R^*, \bar{\gamma}) = \frac{L_{\mathrm{car}}(R^*)}{\cos \bar{\gamma}}.$$

The parameters required to define a high altitude Dubins airplane path, are the same parameters for the Dubins car given in (18) with $R_{\min}$ replaced by R^*, the addition of the optimal flight-path angle $\pm\bar{\gamma}$, the additions of the start and end angles ψ_s and ψ_e, and the addition of the required number of turns at the start helix k_s and the required number of turns at the end helix k_e. Figure 8 shows several Dubins airplane paths for the high-altitude case where the altitude difference is 300 meters over a typical Dubins car path length of 180 meters.

4.2.3 Medium-altitude Dubins Paths

In the medium-altitude case, the altitude difference between the start and end configurations is too large to obtain by flying the Dubins car path at the flight-path angle constraint, but small enough that adding a full turn on the helix at the beginning or end of the path and flying so that $\gamma = \pm\bar{\gamma}$ results in more
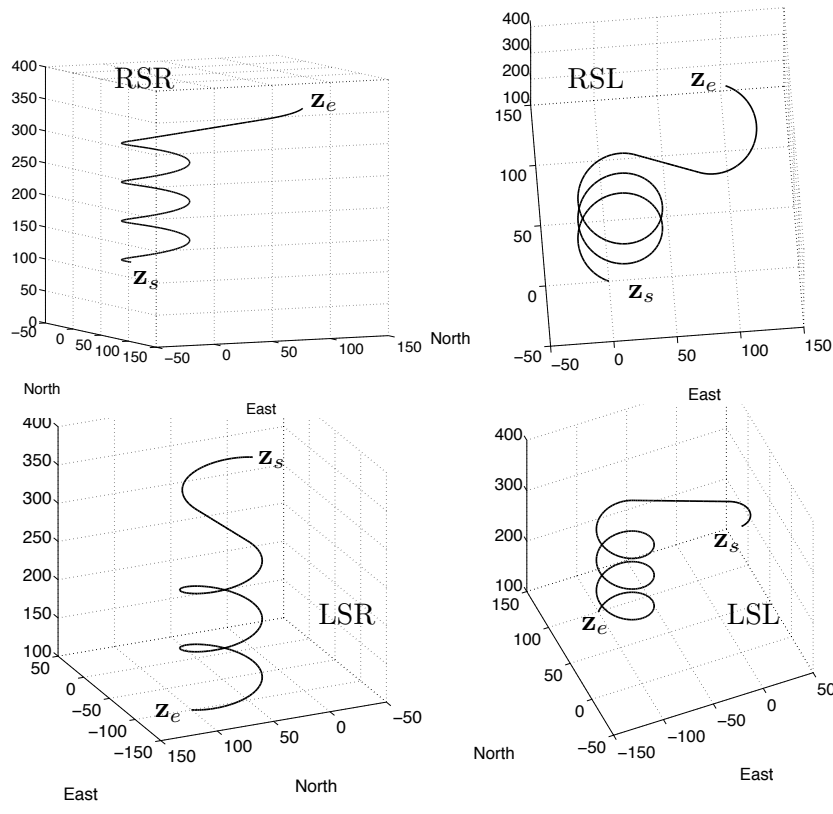
Figure 8: Dubins airplane paths for the high-altitude case.

altitude gain than is needed. As shown in [Chitsaz and LaValle, 2007], the minimum distance path is achieved by setting $\gamma = \text{sign}\,(z_{de} - z_{ds})\,\bar{\gamma}$ and inserting an extra maneuver in the Dubins car path that extends the path length so that the altitude gain when $\gamma = \pm\bar{\gamma}$ is exactly $z_{de} - z_{ds}$. While there are numerous possible ways to extend the path length, the method proposed in this chapter is to add an additional intermediate arc to the start or end of the path, as shown in Figure 9. If the start altitude is lower than the end altitude, then the intermediate arc is inserted immediately after the start helix, as shown for cases RLSR and RLSL in Figure 11. If on the other hand, the start altitude is higher than the end altitude, then the intermediate arc is inserted immediately before the end helix, as shown for cases LSLR and LSRL in Figure 11.

To find the Dubins path in the medium-altitude case, the position of the intermediate arc is parameterized by φ as shown in Figure 9, where

$$\mathbf{z}_i = \mathbf{c}_s + R(\varphi)(\mathbf{z}_s - \mathbf{c}_s).$$

A standard Dubins car path is planned from configuration $(\mathbf{z}_i, \psi_s + \varphi)$ to the end configuration, and the new path length is given by

$$L(\varphi) = \varphi R_{\min} + L_{\text{car}}(\mathbf{z}_i, \psi_s + \varphi, \mathbf{z}_e, \psi_e).$$

A bisection search algorithm is used to find the angle φ^* so that

$$L(\varphi^*)\tan\bar{\gamma} = |z_{de} - z_{ds}|\,.$$

The length of the corresponding Dubins airplane path is given by

$$L_{\text{air}} = \frac{L(\varphi^*)}{\cos\bar{\gamma}}.$$

The parameters needed to describe the Dubins airplane path for the medium-altitude case are shown in Figure 10. The introduction of an intermediate arc requires the additional parameters $\mathbf{c}_i$, ψ_i, λ_i, $\mathbf{w}_i$, and $\mathbf{q}_i$. Therefore, in analogy to (18), the parameters that define a Dubins airplane path are

$$\mathcal{D}_{\text{air}} = (R, \gamma, \mathbf{c}_s, \psi_s, \lambda_s, \mathbf{w}_s, \mathbf{q}_s, \mathbf{c}_i, \psi_i, \lambda_i, \mathbf{w}_i, \mathbf{q}_i, \mathbf{w}_\ell, \mathbf{q}_\ell, \mathbf{c}_e, \psi_e, \lambda_e, \mathbf{w}_e, \mathbf{q}_e). \quad (21)$$

Figure 11 shows several Dubins airplane paths for the medium-altitude case where the altitude difference is 100 meters over a typical Dubins car path length of 180 meters.

4.3   Path Manager for Dubins Airplane

The path manager for the Dubins airplane is shown in Figure 12. With reference to (14), the start helix is defined as $\mathcal{P}_{\text{helix}}(\mathbf{c}_s, \psi_s, \lambda_s, R, \gamma)$. Similarly, the intermediate arc, if it exists, is defined by $\mathcal{P}_{\text{helix}}(\mathbf{c}_i, \psi_i, \lambda_i, R, \gamma)$, and the end helix is given by $\mathcal{P}_{\text{helix}}(\mathbf{c}_e, \psi_e, \lambda_e, R, \gamma)$. With reference to (9), the straight-line path is given by $\mathcal{P}_{\text{line}}(\mathbf{w}_\ell, \mathbf{q}_\ell)$.
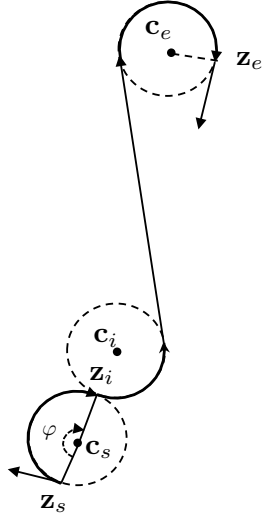
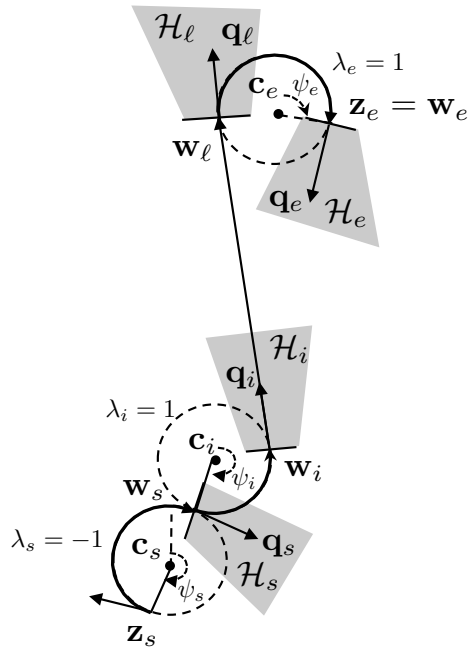Figure 9: The start position of the intermediate arc is found by varying φ.

Figure 10: Parameters that define Dubins airplane path when an intermediate arc is inserted.
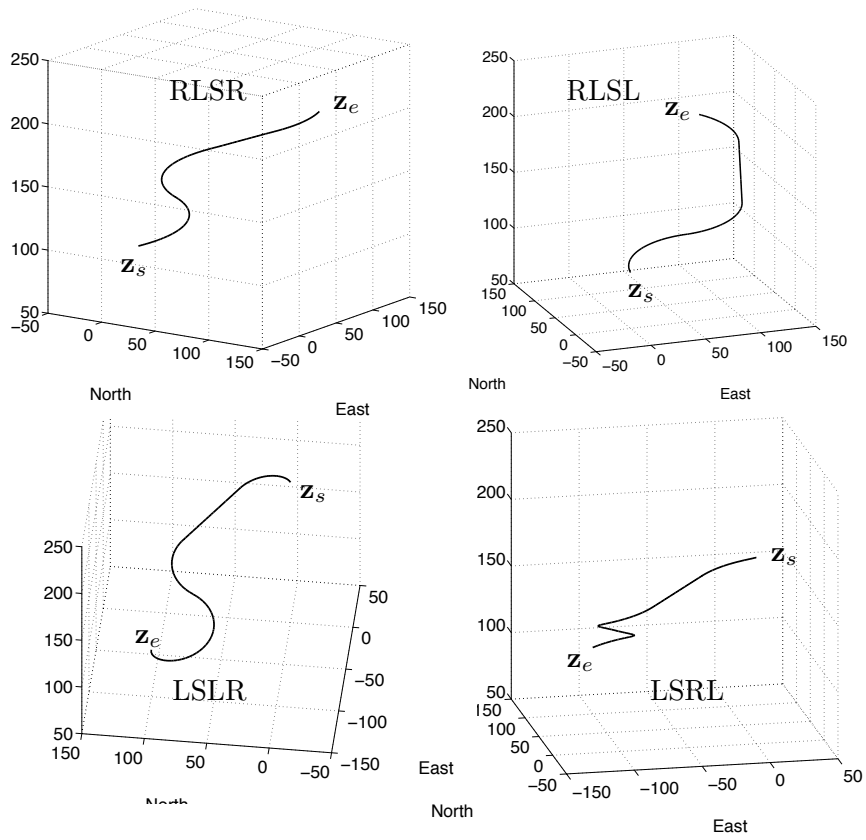
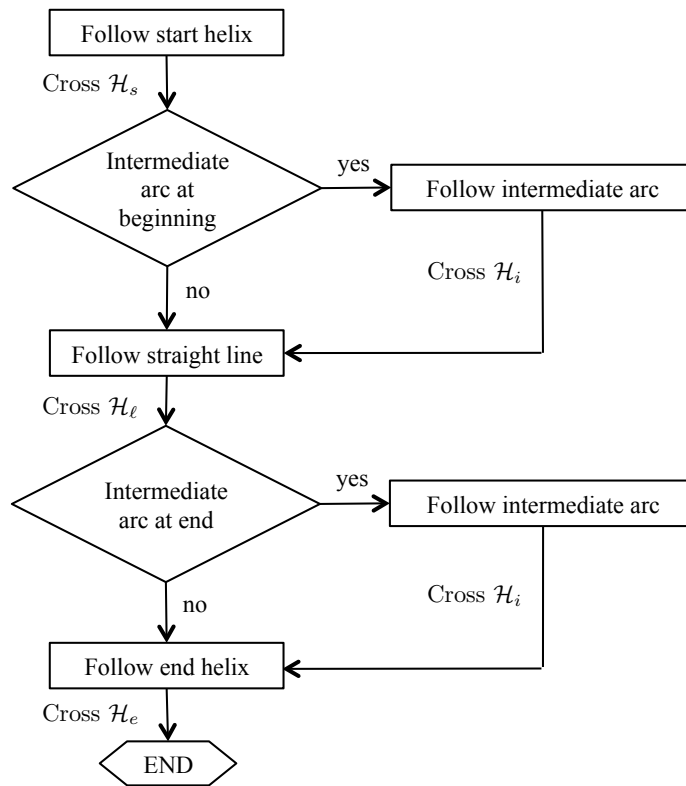Figure 11: Dubins airplane paths for the medium-altitude case.

Figure 12: Flow chart for the Path Manager for following a Dubins airplane path.

4.4 Simulation Results

This section provides some simulation results where Dubins airplane paths are flown on a full six-degree-of-freedom UAV simulator. The aircraft used for the simulation is the Aerosonde model described in [Beard and McLain, 2012]. A low-level autopilot is implemented to regulate the commanded airspeed, bank angle, and flight-path angle. The windspeed in the simulation is set to zero. The simulation is implemented in Matlab/Simulink as described in [Beard and McLain, 2012].

The simulation results for a low altitude gain maneuver are shown in Figure 13, where the planned trajectory is shown in green, and the actual trajectory is shown in black. The difference between the actual and planned trajectories is due to fact that the actual dynamics are much more complicated than the kinematic model given in (1). Simulation results for a medium altitude gain maneuver are shown in Figure 14, and simulation results for a high altitude gain maneuver are shown in Figure 15.
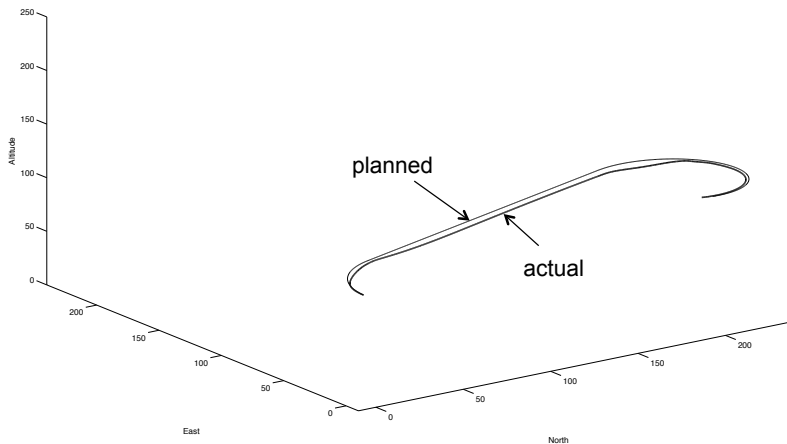

Figure 13: Simulation results for Dubins path with low altitude gain.

5 Conclusion

This chapter describes how to plan and implement Dubins airplane paths for small fixed-wing UAVs. In particular, the Dubins airplane model has been refined to be more consistent with standard aeronautics notation. A complete architecture for following Dubins airplane paths has been defined and implemented and is shown in Figure 1. Dubins airplane paths consists of switching between helical and straight-line paths. The vector-field method described in [Goncalves et al., 2010] has been used to develop guidance laws that regulate
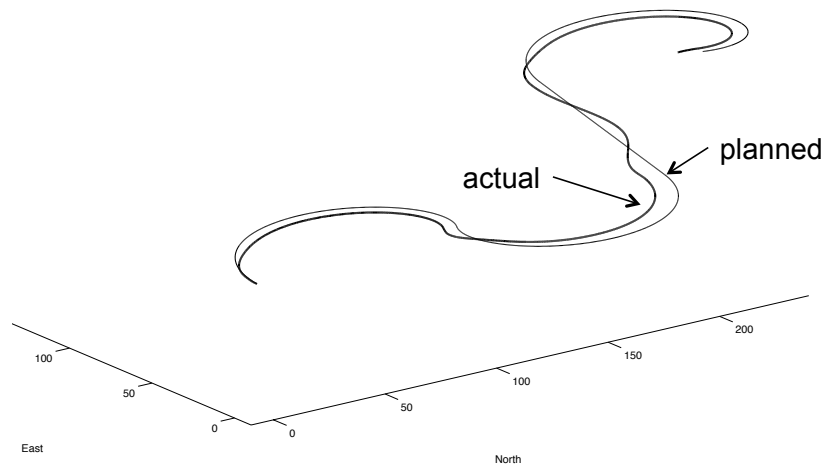
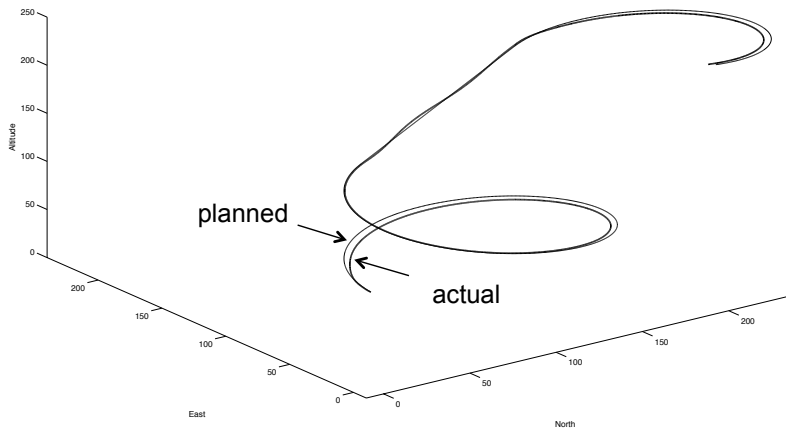Figure 14: Simulation results for Dubins path with medium altitude gain.

Figure 15: Simulation results for Dubins path with high altitude gain.

the Dubins airplane to follow the associated helical and straight-line paths. For medium and high altitude gain scenarios, there are many possible Dubins paths. This chapter suggests selecting the path that maximizes the average altitude of the aircraft during the maneuver.

There are many possible extensions that warrant future work. First, there is a need to extend these methods to windy environments, including both constant wind as well as heavy gusts. Second, the assumed fast inner loops on airspeed, roll angle, and flight-path angle is often violated, especially for flight-path angle. There may be some advantage, for path optimality in particular, to factoring the time constants of the inner loops into the planning procedure. Third, this chapter assumes a decoupling between flight-path angle and airspeed. Except for highly overpowered vehicles, however, achieving a positive flight-path angle will reduce the airspeed, and achieving a negative flight-path angle will increase the airspeed. Taking these effects into account will obviously change the optimality of the paths. Finally, there are a variety of methods that have been proposed for achieving vector-field following (see [Lawrence et al., 2008, Park et al., 2007, Nelson et al., 2007]). The method used in this chapter is only one possibility, that in fact, proved challenging to tune. One of the issues is that the method assumes single integrator dynamics in each direction of motion. More robust 3D vector-field following techniques that account for the nonholonomic kinematic model of the Dubins airplane need to be developed.

References

Ross Anderson and Dejan Milutinovic. A stochastic approach to Dubins feedback control for target tracking. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, September 2011.

A. Balluchi, A. Bicchi, A. Balestrino, and G. Casalino. Path tracking control for Dubins cars. In *Proceedings of the International Conference on Robotics and Automation*, pages 3123–3128, Minneapolis, MN, 1996.

Randal W. Beard and Timothy W. McLain. *Small Unmanned Aircraft: Theory and Practice.* Princeton University Press, 2012.

Shelby Brunke and Mark E. Campbell. Square root sigma point filtering for real-time, nonlinear estimation. *Journal of Guidance*, 27(2):314–317, 2004.

Hamidreza Chitsaz and Steven M. LaValle. Time-optimal paths for a Dubins airplane. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 2379–2384, December 2007.

Raghvendra V. Cowlagi and Panagiotis Tsiotras. Shortest distance problems in graphs using history-dependent transition costs with application to kinodynamic path planning. In *Proceedings of the American Control Conference*, pages 414–419, St. Louis, June 2009.

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*, 79:497–516, 1957.

Joseph Egbert and Randal W. Beard. Low-altitude road following using strapdown cameras on miniature air vehicles. *Mechatronics*, 21(5):831–843, August 2011.

Jack Elston and Erik W. Frew. Hierarchical distributed control for search and track by heterogeneous aerial robot network. In *Proceedings of the International Conference on Robotics and Automation*, pages 170–175, Pasadena, CA, May 2008.

Jack Elston, Brian Argrow, Adam Houston, and Eric Frew. Design and validation of a system for targeted observations of tornadic supercells using unmanned aircraft. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System*, pages 101–106E, Taipei, Taiwan, October 2010.

Eric Few, Tim McGee, ZuWhan Kim, Xiao Xiao, Stephen Jackson, Michael Morimoto, Sinvakumar Rathinam, Jose Padial, and Raja Sengupta. Vision-based road-following using a small autonomous aircraft. In *Proceedings of the Aerospace Conference*, volume 5, pages 3006–3015, March 2004.

Eric W. Frew, Cory Dixon, Jack Elston, and Maciej Stachura. Active sensing by unmanned aircraft systems in realistic communications environments. In *Proceedings of the IFAC Workshop on Networked Robotics*, Golden, Colorado, October 2009.

Vinicius M. Goncalves, Luciano C. A. Pimenta, Carlos A. Maia, Bruno C. O. Durtra, and Guilherme A. S. Pereira. Vector fields for robot navigation along time-varying curves in n-dimensions. *IEEE Transactions on Robotics*, 26(4): 647–659, August 2010.

Clarence Hanson, Jeremy Richardson, and Anouck Girard. Path planning of a Dubins vehicle for sequential target observation with ranged sensors. In *Proceedings of the American Control Conference*, pages 1698–1703, San Francisco, CA, June 2011.

Sikha Hosak and Debasiah Ghose. Optimal geometrical path in 3D with curvature constraint. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 113–118, Taipei, Taiwan, October 2010.

Sertac Karaman and Emilio Frazzoli. Incremental sampling-based algorithms for optimal motion planning. *International Journal of Robotic Research*, 2010. In review.

Dale A. Lawrence, Eric W. Frew, and William J. Pisano. Lyapunov vector fields for autonomous unmanned aircraft flight control. In *AIAA Journal of Guidance, Control, and Dynamics*, volume 31, pages 1220–12229, September-October 2008.

Frank L. Lewis. *Optimal Control.* John Wiley and Sons, New York, 1986.

Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear complementary filters on special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5):1203–1218, June 2008.

E. A. Misawa and J. K. Hedrick. Nonlinear observers – state-of-the-art survey. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 111:344–352, September 1989.

Derek R. Nelson, D. Blake Barber, Timothy W. McLain, and Randal W. Beard. Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics*, 37(3):519–529, June 2007.

Robert C. Nelson. *Flight Stability and Automatic Control.* McGraw-Hill, Boston, Massachusetts, 2nd edition, 1998.

Sanghyuk Park, John Deyst, and Jonathan P. How. Performance and Lyapunov stability of a nonlinear path-following guidance method. *AIAA Journal of Guidance, Control, and Dynamics*, 30(6):1718–1728, November-December 2007.

Warren F. Phillips. *Mechanics of Flight.* Wiley, 2004.

A. Rahmani, X. C. Ding, and M. Egerstedt. Optimal motion primitives for multi-UAV convoy protection. In *Proceedings of the International Conference on Robotics and Automation*, pages 4469–4474, Anchorage, AK, May 2010.

Sivakumar Rathinam, Zu Kim, Aram Soghikian, and Raja Sengupta. Vision based following of locally linear structure using an unmanned aerial vehicle. In *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference*, pages 6085–6090, Seville, Spain, December 2005.

Tal Shima, Steve Rasmussen, and Dave Gross. Assigning micro UAVs to task tours in an urban terrain. *IEEE Transactions on Control Systems Technology*, 15(4):601–612, July 2007.

Stephen C. Spry, Anouck R. Girard, and J. Karl Hedrick. Convoy protection using multiple unmanned aerial vehicles: Organization and coordination. In *Proceedings of the American Control Conference*, pages 3524–3529, Portland, OR, June 2005.

Brian L. Stevens and Frank L. Lewis. *Aircraft Control and Simulation.* John Wiley & Sons, Inc., Hoboken, New Jersey, 2nd edition, 2003.

P. B. Sujit, J. M. George, and R. W. Beard. Multiple UAV coalition formation. In *Proceedings of the American Control Conference*, pages 2010–2015, Seattle, WA, June 2007.

Guang Yang and Vikram Kapila. Optimal path planning for unmanned air vehicles with kinematic and tactical constraints. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1301–1306, Las Vegas, NV, 2002.

Thomas R. Yechout, Steven L. Morris, David E. Bossert, and Wayne F. Hallgren. *Introduction to Aircraft Flight Mechanics*. AIAA Education Series. American Institute of Aeronautics and Astronautics, 2003.

Chao Yong and Eric J. Barth. Real-time dynamic path planning for Dubins' nonholonomic robot. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2418–2423, San Diego, CA, December 2006.

Huili Yu and Randal W. Beard. A vision-based collision avoidance technique for micro air vehicles using local-level frame mapping and path planning. *Autonomous Robots*, 34(1-2):93–109, 2013.

Index