Chapter 11

Path Manager
Control Architecture

- Path planner
- Path manager
- Path following
- Autopilot
- Unmanned aircraft

- Destination, obstacles
- Waypoints
- Path definition
- Airspeed, altitude, heading, commands
- Servo commands
- Wind
- Map
- Status
- Tracking error
- Position error
- On-board sensors
- State estimator
- $\hat{x}(t)$
Path Definition

Waypoint path defined as ordered sequence of waypoints

\[ \mathcal{W} = \{ \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N \} \]

where \[ \mathbf{w}_i = (w_{n,i}, w_{e,i}, w_{d,i})^\top \in \mathbb{R}^3. \]
Waypoint Switching

- Two methods
  - $b$-ball around waypoint
  - half plane through waypoint
**Waypoint Switching**

Given point \( \mathbf{r} \in \mathbb{R}^3 \) and normal vector \( \mathbf{n} \in \mathbb{R}^3 \), define half plane

\[
\mathcal{H}(\mathbf{r}, \mathbf{n}) \triangleq \{ \mathbf{p} \in \mathbb{R}^3 : (\mathbf{p} - \mathbf{r})^\top \mathbf{n} \geq 0 \}
\]

Define unit vector pointing in direction of line \( \overrightarrow{\mathbf{w}_i \mathbf{w}_{i+1}} \) as

\[
\mathbf{q}_i \triangleq \frac{\mathbf{w}_{i+1} - \mathbf{w}_i}{\| \mathbf{w}_{i+1} - \mathbf{w}_i \|}
\]

Unit normal to the 3-D half plane that separates the line \( \overrightarrow{\mathbf{w}_{i-1} \mathbf{w}_i} \) from the line \( \overrightarrow{\mathbf{w}_i \mathbf{w}_{i+1}} \) is given by

\[
\mathbf{n}_i \triangleq \frac{\mathbf{q}_{i-1} + \mathbf{q}_i}{\| \mathbf{q}_{i-1} + \mathbf{q}_i \|}
\]

MAV tracks straight-line path from \( \mathbf{w}_{i-1} \) to \( \mathbf{w}_i \) until it enters \( \mathcal{H}(\mathbf{w}_i, \mathbf{n}_i) \), at which point it will track straight-line path from \( \mathbf{w}_i \) to \( \mathbf{w}_{i+1} \)
Waypoint Following

Algorithm 5: Follow Waypoints: \((r, q) = \text{followWpp}(W, p)\)

**Input:** Waypoint path \(W = \{w_1, \ldots, w_N\}\), MAV position \(p = (p_n, p_e, p_d)^\top\).

**Require:** \(N \geq 3\)

1. if New waypoint path \(W\) is received then
2. Initialize waypoint index: \(i \leftarrow 2\)
3. end if
4. \(r \leftarrow w_{i-1}\)
5. \(q_{i-1} \leftarrow \frac{w_i - w_{i-1}}{\|w_i - w_{i-1}\|}\)
6. \(q_i \leftarrow \frac{w_{i+1} - w_i}{\|w_{i+1} - w_i\|}\)
7. \(n_i \leftarrow \frac{q_{i-1} + q_i}{\|q_{i-1} + q_i\|}\)
8. if \(p \in \mathcal{H}(w_i, n_i)\) then
9. Increment \(i \leftarrow (i + 1)\) until \(i = N - 1\)
10. end if
11. return \(r, q = q_{i-1}\) at each time step

In the remainder of this section we will focus on smoothed paths like those shown in figure 11.3. The geometry near the transition is shown in figure 11.4. With the unit vector \(q_i\) aligned with the line between waypoints \(w_i\) and \(w_{i+1}\) defined as in equation (11.2), the angle between \(w_{i-1}w_i\) and \(w_iw_{i+1}\) is given by

\[
\rho_{\Delta} = \cos^{-1}\left(\frac{q_{i-1}^\top q_i}{\|q_{i-1} + q_i\|}\right).
\]  

(11.3)
Waypoint Following Results

Path Manager – Straight Line

Fillet Transition

Transition between path segments can be smoothed by adding a fillet.
Fillet Geometry

\[
\frac{R}{\sin \frac{\theta}{2}} - R = \frac{R}{\tan \frac{\theta}{2}}
\]

Fillet Smoothing Half Planes

Fillet center defined as

\[ c = w_i - \left( \frac{R}{\sin \frac{\theta}{2}} \right) \frac{q_{i-1} - q_i}{\|q_{i-1} - q_i\|} \]

Half plane \( \mathcal{H}_1 \) is defined by location

\[ r_1 = w_i - \left( \frac{R}{\tan \frac{\theta}{2}} \right) q_{i-1} \]

and normal vector \( q_{i-1} \)

Half plane \( \mathcal{H}_2 \) is defined by location

\[ r_2 = w_i + \left( \frac{R}{\tan \frac{\theta}{2}} \right) q_i \]

and normal vector \( q_i \)
Waypoint Following with Fillets

**Algorithm 6** Follow Waypoints with Fillets: $\text{flag}, r, q, c, \rho, \lambda) = \text{followWppFillet}(W, p, R)$

| Input: | Waypoint path $W = \{w_1, \ldots, w_N\}$, MAV position $p = (p_n, p_e, p_d)^T$, fillet radius $R$. |
| Require: | $N \geq 3$ |
| 1: | if New waypoint path $W$ is received then |
| 2: | Initialize waypoint index: $i \leftarrow 2$, and state machine: state $\leftarrow 1$. |
| 3: | end if |
| 4: | $q_{i-1} \leftarrow \frac{w_i - w_{i-1}}{\|w_i - w_{i-1}\|}$ |
| 5: | $q_i \leftarrow \frac{w_{i+1} - w_i}{\|w_{i+1} - w_i\|}$ |
| 6: | $\varrho \leftarrow \cos^{-1}(q_{i-1}^T q_i)$ |
| 7: | if state $= 1$ then |
| 8: | flag $\leftarrow 1$ |
| 9: | $r \leftarrow w_{i-1}$ |
| 10: | $q \leftarrow q_{i-1}$ |
| 11: | $z \leftarrow w_i - \left(\frac{R}{\tan(\varrho/2)}\right) q_{i-1}$ |
| 12: | if $p \in H(z, q_{i-1})$ then |
| 13: | state $\leftarrow 2$ |
| 14: | end if |
| 15: | else if state $= 2$ then |
| 16: | flag $\leftarrow 2$ |
| 17: | $c \leftarrow w_i + \left(\frac{R}{\sin(\varrho/2)}\right) \frac{q_{i-1} - q_i}{\|q_{i-1} - q_i\|}$ |
| 18: | $\rho \leftarrow R$ |
| 19: | $\lambda \leftarrow \text{sign}(q_{i-1}, w_i, c - q_{i-1}, q_i, n)$ |
| 20: | $z \leftarrow w_i + \left(\frac{R}{\tan(\varrho/2)}\right) q_i$ |
| 21: | if $p \in H(z, q_i)$ then |
| 22: | $i \leftarrow (i + 1)$ until $i = N - 1$. |
| 23: | state $\leftarrow 1$ |
| 24: | end if |
| 25: | end if |
| 26: | return $\text{flag}, r, q, c, \rho, \lambda$. |
Waypoint Following with Fillets

Path Manager – Fillets

Fillet Path Length

Straight-line path length (no fillets):

$$|\mathcal{W}| \triangleq \sum_{i=2}^{N} \|\mathbf{w}_i - \mathbf{w}_{i-1}\|.$$ 

Path length with fillets:

$$|\mathcal{W}|_F = |\mathcal{W}| + \sum_{i=2}^{N} \left( R\varphi_i - \frac{2R}{\tan \frac{\varphi_i}{2}} \right).$$
Dubins Paths

For vehicle with kinematics given by

\[
\begin{align*}
\dot{p}_n &= V \cos \vartheta \\
\dot{p}_e &= V \sin \vartheta \\
\dot{\vartheta} &= u,
\end{align*}
\]

where $V$ is constant and $u \in [-\bar{u}, \bar{u}]$, time-optimal path between two different configurations consists of circular arc, followed by straight line, and concluding with another circular arc to the final configuration, where the radius of the circular arcs is $V/\bar{u}$.

Case I: R-S-R

Case II: R-S-L

Case III: L-S-R

Case IV: L-S-L

Dubins path is defined as shortest of four cases

Path Length Preliminaries

Given position $p$, course $\chi$, and radius $R$, centers of right and left turning circles are given by

\[ c_r = p + R \left( \cos(\chi + \frac{\pi}{2}), \sin(\chi + \frac{\pi}{2}), 0 \right)^T \]
\[ c_l = p + R \left( \cos(\chi - \frac{\pi}{2}), \sin(\chi - \frac{\pi}{2}), 0 \right)^T \]

For clockwise circles, angular distance between $\vartheta_1$ and $\vartheta_2$ given by

\[ |\vartheta_2 - \vartheta_1|_C W \triangleq \langle 2\pi + \vartheta_2 - \vartheta_1 \rangle, \]

where

\[ \langle \varphi \rangle \triangleq \varphi \mod 2\pi \]

For counter clockwise circles,

\[ |\vartheta_2 - \vartheta_1|_{C CW} \triangleq \langle 2\pi - \vartheta_2 + \vartheta_1 \rangle \]
Alternative Viewpoint

For clockwise circles, angular distance between $\vartheta_1$ and $\vartheta_2$ given by

$$|\vartheta_2 - \vartheta_1|_{CW} \triangleq \langle 2\pi + \vartheta_2 - \vartheta_1 \rangle,$$

where

$$\langle \varphi \rangle \triangleq \varphi \mod 2\pi$$
Alternative Viewpoint

For counter clockwise circles,

\[ |\vartheta_2 - \vartheta_1|_{CCW} \triangleq \langle 2\pi - \vartheta_2 + \vartheta_1 \rangle \]
Dubins Case I: R-S-R

Distance traveled along $c_{rs}$

$$R\langle 2\pi + \langle \vartheta - \frac{\pi}{2} \rangle - \langle \chi_s - \frac{\pi}{2} \rangle \rangle$$

Distance traveled along $c_{re}$

$$R\langle 2\pi + \langle \chi_e - \frac{\pi}{2} \rangle - \langle \vartheta - \frac{\pi}{2} \rangle \rangle$$

Total path length:

$$L_1 = \|c_{rs} - c_{re}\| + R\langle 2\pi + \langle \vartheta - \frac{\pi}{2} \rangle - \langle \chi_s - \frac{\pi}{2} \rangle \rangle + R\langle 2\pi + \langle \chi_e - \frac{\pi}{2} \rangle - \langle \vartheta - \frac{\pi}{2} \rangle \rangle$$
Dubins Case II: R-S-L

Distance traveled along $c_{rs}$

$$R \langle 2\pi + (\vartheta - \vartheta_2) - (\chi_s - \frac{\pi}{2}) \rangle$$

Distance traveled along $c_{le}$

$$R \langle 2\pi + (\vartheta_2 + \pi) - (\chi_e + \frac{\pi}{2}) \rangle$$

Total path length:

$$L_2 = \sqrt{l^2 - 4R^2} + R \langle 2\pi + (\vartheta - \vartheta_2) - (\chi_s - \frac{\pi}{2}) \rangle + R \langle 2\pi + (\vartheta_2 + \pi) - (\chi_e + \frac{\pi}{2}) \rangle$$

Dubins Case III: L-S-R

Distance traveled along $c_{ls}$

$$R\langle 2\pi + \frac{\chi_s}{2} - \varphi + \varphi_2 \rangle$$

Distance traveled along $c_{re}$

$$R\langle 2\pi + \frac{\chi_e}{2} - \varphi + \varphi_2 - \pi \rangle$$

Total path length:

$$L_3 = \sqrt{l^2 - 4R^2} + R\langle 2\pi + \frac{\chi_s}{2} - \varphi + \varphi_2 \rangle + R\langle 2\pi + \frac{\chi_e}{2} - \varphi + \varphi_2 - \pi \rangle$$

Dubins Case IV: L-S-L

Distance traveled along $c_{ls}$

$$R\langle 2\pi + \langle \chi_s + \frac{\pi}{2} \rangle - \langle \vartheta + \frac{\pi}{2} \rangle \rangle$$

Distance traveled along $c_{le}$

$$R\langle 2\pi + \langle \vartheta + \frac{\pi}{2} \rangle - \langle \chi_e + \frac{\pi}{2} \rangle \rangle$$

Total path length:

$$L_4 = \|c_{ls} - c_{le}\| + R\langle 2\pi + \langle \chi_s + \frac{\pi}{2} \rangle - \langle \vartheta + \frac{\pi}{2} \rangle \rangle + R\langle 2\pi + \langle \vartheta + \frac{\pi}{2} \rangle - \langle \chi_e + \frac{\pi}{2} \rangle \rangle$$

Dubins Path Half-plane Switching

Given desired start and end configurations

- $p_s$: start location
- $\chi_s$: start direction
- $p_e$: end location
- $\chi_e$: end direction

Calculate Dubins path parameters

- $c_s$: start circle location
- $\lambda_s$: start circle direction
- $c_e$: end circle location
- $\lambda_e$: end circle direction
- $z_1, q_1$: half-plane $\mathcal{H}_1$ parameters
- $z_2, q_2$: half-plane $\mathcal{H}_2$ parameters
- $z_3, q_3$: half-plane $\mathcal{H}_3$ parameters

Dubins Path Parameters Algorithm

Algorithm 7 Find Dubins Parameters:

(L, c₁, λ₁, c₂, λ₂, z₁, z₂, z₃, q₃) =
findDubinsParameters(p₁, x₁, pₑ, xₑ, R)

Input: Start configuration (p₁, x₁), End configuration (pₑ, xₑ),
Radius R.

Require: ||pₑ - p₁|| ≥ 3R

Require: R is larger than minimum turn radius of MAV

1: c₁ ← p₁ + RRₑ\(\left(\frac{\pi}{2}\right)\)(cos xₑ, sin xₑ, 0)\(^T\)
2: c₀ ← p₁ + RRₑ\(\left(\frac{-\pi}{2}\right)\)(cos x₁, sin x₁, 0)\(^T\)
3: cₑ ← pₑ + RRₑ\(\left(\frac{\pi}{2}\right)\)(cos xₑ, sin xₑ, 0)\(^T\)
4: cₑ ← pₑ + RRₑ\(\left(\frac{-\pi}{2}\right)\)(cos xₑ, sin xₑ, 0)\(^T\)
5: Compute L₁, L₂, L₃, and L₄ using equations (11.9) through (11.12).
6: L ← min{L₁, L₂, L₃, L₄}
7: if arg min{L₁, L₂, L₃, L₄} = 1 then
8: c₁ ← c₁₁, λ₁ ← +1, cₑ ← cₑ₁, λₑ ← +1
9: q₁ ← c₁ - cₑ
10: z₁ ← c₁ + RRₑ\(\left(-\frac{\pi}{2}\right)\)q₁
11: z₂ ← cₑ + RRₑ\(\left(-\frac{\pi}{2}\right)\)q₁
12: else if arg min{L₁, L₂, L₃, L₄} = 2 then
13: c₁ ← c₃₁, λ₁ ← +1, cₑ ← cₑ₁, λₑ ← -1
14: ℓ ← ||cₑ - c₁||
15: \(\theta₁ \leftarrow \angle \left(cₑ - c₁\right)\)
16: \(\theta₂ \leftarrow \theta₁ + \sin^{-1} \frac{2\ell}{R}\)
17: q₁ ← Rₑ(\(\theta₂ + \frac{\pi}{2}\))e₁
18: z₁ ← cₑ + RRₑ\(\left(\theta₂\right)\)e₁
19: z₂ ← cₑ + RRₑ\(\left(\theta₂ - \pi\right)\)e₁
20: else if arg min{L₁, L₂, L₃, L₄} = 3 then
21: c₁ ← c₁₁, λ₁ ← -1, cₑ ← cₑ₁, λₑ ← +1
22: ℓ ← ||cₑ - c₁||
23: \(\theta₁ \leftarrow \angle \left(cₑ - c₁\right)\)
24: \(\theta₂ \leftarrow \cos^{-1} \frac{2\ell}{R}\)
25: q₁ ← Rₑ(\(\theta₁ + \theta₂ - \frac{\pi}{2}\))e₁
26: z₁ ← cₑ + RRₑ\(\left(\theta₁ + \theta₂\right)\)e₁
27: z₂ ← cₑ + RRₑ\(\left(\theta₁ + \theta₂ - \pi\right)\)e₁
28: else if arg min{L₁, L₂, L₃, L₄} = 4 then
29: c₁ ← c₁₁, λ₁ ← -1, cₑ ← cₑ₁, λₑ ← -1
30: q₁ ← cₑ - c₁
31: z₁ ← cₑ + RRₑ\(\left(\frac{\pi}{2}\right)\)q₁
32: z₂ ← cₑ + RRₑ\(\left(\frac{\pi}{2}\right)\)q₂
33: end if
34: z₃ ← pₑ
35: q₃ ← Rₑ(xₑ)e₁

Dubins Path Following Algorithm

**Algorithm 8** Follow Waypoints with Dubins: \((\text{flag, r, q, c, } \rho, \lambda) = \text{followWppDubins}(\mathcal{P}, \mathbf{p}, R)\)

**Input:** Configuration path \(\mathcal{P} = (w_1, x_1), \ldots, (w_N, x_N)\), MAV position \(\mathbf{p} = (p_n, p_e, p_d)^T\), fillet radius \(R\).

**Require:** \(N \geq 3\)
1: if New configuration path \(\mathcal{P}\) is received then
2: Initialize waypoint pointer: \(i \leftarrow 2\), and state machine: state \(\leftarrow 1\).
3: end if
4: \((L, c_s, \lambda_s, c_e, \lambda_e, z_1, q_1, z_2, z_3, q_3) \leftarrow \text{findDubinsParameters}(w_{i-1}, x_{i-1}, w_i, x_i, R)\)
5: if state = 1 then
6: flag \(\leftarrow 2\), c \(\leftarrow c_s, \rho \leftarrow R, \lambda \leftarrow \lambda_s\)
7: if \(\mathbf{p} \in \mathcal{H}(z_1, -q_1)\) then
8: state \(\leftarrow 2\)
9: end if
10: else if state = 2 then
11: if \(\mathbf{p} \in \mathcal{H}(z_1, q_1)\) then
12: state \(\leftarrow 3\)
13: end if
14: else if state = 3 then
15: flag \(\leftarrow 1\), r \(\leftarrow z_1, q \leftarrow q_1\)
16: if \(\mathbf{p} \in \mathcal{H}(z_2, q_1)\) then
17: state \(\leftarrow 4\)
18: end if
19: else if state = 4 then
20: flag \(\leftarrow 2\), c \(\leftarrow c_e, \rho \leftarrow R, \lambda \leftarrow \lambda_e\)
21: if \(\mathbf{p} \in \mathcal{H}(z_3, -q_3)\) then
22: state \(\leftarrow 5\)
23: end if
24: else if state = 5 then
25: if \(\mathbf{p} \in \mathcal{H}(z_3, q_3)\) then
26: state \(\leftarrow 1\)
27: \(i \leftarrow (i + 1)\) until \(i = N\).
28: \((L, c_s, \lambda_s, c_e, \lambda_e, z_1, q_1, z_2, z_3, q_3) \leftarrow \text{findDubinsParameters}(w_{i-1}, x_{i-1}, w_i, x_i, R)\)
29: end if
30: end if
31: return flag, r, q, c, \(\rho, \lambda\).
Dubins Path Following Results

Path Manager – Dubins

Dubins Airplane Model


Dubins Airplane model:

\[
\begin{align*}
\dot{r}_n &= V \cos \psi \cos \gamma^c \\
\dot{r}_e &= V \sin \psi \cos \gamma^c \\
\dot{r}_d &= -V \sin \gamma^c \\
\dot{\psi} &= \frac{g}{V} \tan \phi^c
\end{align*}
\]

Where the commanded flight path angle \( \gamma^c \) and the commanded roll angle \( \phi^c \) are constrained by

\[
|\phi^c| \leq \bar{\phi} \quad |\gamma^c| \leq \bar{\gamma}.
\]
3D Vector Field Path Following


The path is specified as the intersection of two 2D manifolds given by

\[ \alpha_1(r) = 0 \]
\[ \alpha_2(r) = 0 \]

\( r \in \mathbb{R}^3 \). Define the composite function

\[ W(r) = \frac{1}{2} \alpha_1^2(r) + \frac{1}{2} \alpha_2^2(r), \]

Note that the gradient

\[ \frac{\partial W}{\partial r} = \alpha_1(r) \frac{\partial \alpha_1}{\partial r}(r) + \alpha_2(r) \frac{\partial \alpha_2}{\partial r}(r). \]

points away from the path.
3D Vector Field Path Following

The desired velocity vector can be chosen as

\[
u' = \begin{pmatrix} -K_1 \frac{\partial W}{\partial r} \\ + \begin{pmatrix} K_2 \frac{\partial \alpha_1}{\partial r} \\ \times \frac{\partial \alpha_2}{\partial r} \end{pmatrix}\end{pmatrix}
\]

velocity directed toward the path velocity directed along the path

where \( K_1 > 0 \) and \( K_2 \) are symmetric tuning matrices, and the definiteness of \( K_2 \) determines the direction of travel along the path.

Since \( u' \) may not equal \( V_a \), normalize to get

\[
u = V_a \frac{u'}{\|u'\|}.
\]
3D Vector Field Path Following

Setting the NED components of the velocity of the Dubins airplane model 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 $\gamma^c$, and $\psi^d$ results in

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

Assuming the inner-loop lateral-directional dynamics are accurately modeled by the coordinated-turn equation, the commanded roll angle is

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

where $k_{\phi}$ is a positive constant.
3D Vector Field – Straight Line path

The straight line path is given by

\[ \mathcal{P}_{\text{line}}(c_\ell, \psi_\ell, \gamma_\ell) = \{ r \in \mathbb{R}^3 : r = c_\ell + \sigma q_\ell, \sigma \in \mathbb{R} \}, \]

where

\[ q_\ell = \begin{pmatrix} q_n \\ q_e \\ q_d \end{pmatrix} = \begin{pmatrix} \cos \psi_\ell \cos \gamma_\ell \\ \sin \psi_\ell \cos \gamma_\ell \\ -\sin \gamma_\ell \end{pmatrix}. \]

Define

\[ n_{\text{lon}} = \begin{pmatrix} -\sin \psi_\ell \\ \cos \psi_\ell \\ 0 \end{pmatrix}, \]

\[ n_{\text{lat}} = n_{\text{lon}} \times q_\ell = \begin{pmatrix} -\cos \psi_\ell \sin \gamma_\ell \\ -\sin \psi_\ell \sin \gamma_\ell \\ -\cos \gamma_\ell \end{pmatrix}, \]

to get

\[ \alpha_{\text{lon}}(r) = n_{\text{lon}}^T (r - c_\ell) = 0 \]

\[ \alpha_{\text{lat}}(r) = n_{\text{lat}}^T (r - c_\ell) = 0. \]
3D Vector Field – Helical Path

A helical path is then defined as

\[ P_{\text{helix}}(c_h, \psi_h, \lambda_h, R_h, \gamma_h) = \{ r \in \mathbb{R}^3 : \alpha_{\text{cyl}}(r) = 0 \text{ and } \alpha_{\text{pl}}(r) = 0 \}. \]

where

\[ \alpha_{\text{cyl}}(r) = \left( \frac{r_n - c_n}{R_h} \right)^2 + \left( \frac{r_e - c_e}{R_h} \right)^2 - 1 \]
\[ \alpha_{\text{pl}}(r) = \left( \frac{r_d - c_d}{R_h} \right) + \frac{\tan \gamma_h}{\lambda_h} \left( \tan^{-1} \left( \frac{r_e - c_e}{r_n - c_n} \right) - \psi_h \right) \]

where the initial position along the helix is

\[ r(0) = c_h + \begin{pmatrix} R_h \cos \psi_h \\ R_h \sin \psi_h \\ 0 \end{pmatrix}. \]

\( c_h \) is the center of the helix, \( R_h \) is the radius, \( \gamma_h \) is the climb angle.
Dubins Airplane Paths

Given the start configuration $z_s = (z_{ns}, z_{es}, z_{ds}, \psi_s)^\top$ and the end configuration $z_e = (z_{ne}, z_{ee}, z_{de}, \psi_e)^\top$ and the turn radius $R$, let $L_{\text{car}}(R, z_s, z_e)$ be the length of the Dubins car path.

Recall that $\bar{\gamma}$ is the limit of the flight path angle. There are three possible cases for the commanded altitude gain:

**Low Altitude:**

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

i.e., the altitude gain can be achieved by following the Dubins car path with a flight path angle $|\gamma^c| \leq \bar{\gamma}$.

**High Altitude:**

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

i.e., the altitude gain is larger than following the Dubins car path plus one orbit, at flight path angle $\bar{\gamma}$.

**Medium Altitude:**

$$L_{\text{car}}(R_{\text{min}}) \tan \bar{\gamma} < |z_{de} - z_{ds}| \leq [L_{\text{car}}(R_{\text{min}}) + 2\pi R_{\text{min}}] \tan \bar{\gamma}.$$
Low Altitude Dubins Airplane Paths

\[ \gamma^* = \tan^{-1} \left( \frac{|z_{de} - z_{ds}|}{L_{\text{car}}(R_{\text{min}})} \right) \]

\[ R^* = R_{\text{min}} \]

\[ L_{\text{air}}(R_{\text{min}}, \gamma^*) = \frac{L_{\text{car}}(R_{\text{min}})}{\cos \gamma^*}. \]
High Altitude Dubins Airplane Paths

Find smallest integer \( k \) such that

\[
(L_{\text{car}}(R_{\text{min}}) + 2\pi k R_{\text{min}}) \tan \tilde{\gamma} \\
\leq |z_{de} - z_{ds}| < \\
(L_{\text{car}}(R_{\text{min}}) + 2\pi (k + 1) R_{\text{min}}) \tan \tilde{\gamma}.
\]

Increase the radius \( R^* \) so that

\[
(L_{\text{car}}(R^*) + 2\pi k R^*) \tan \tilde{\gamma} = |z_{de} - z_{ds}|.
\]

The resulting path is

\[
L_{\text{air}}(R^*, \tilde{\gamma}) = \frac{L_{\text{car}}(R^*)}{\cos \tilde{\gamma}}.
\]
Medium Altitude Dubins Airplane Paths

Key idea: Add an intermediate helix along the path.

Could add intermediate helix at start, end, or in the middle of path.
Medium Altitude Dubins Airplane Paths