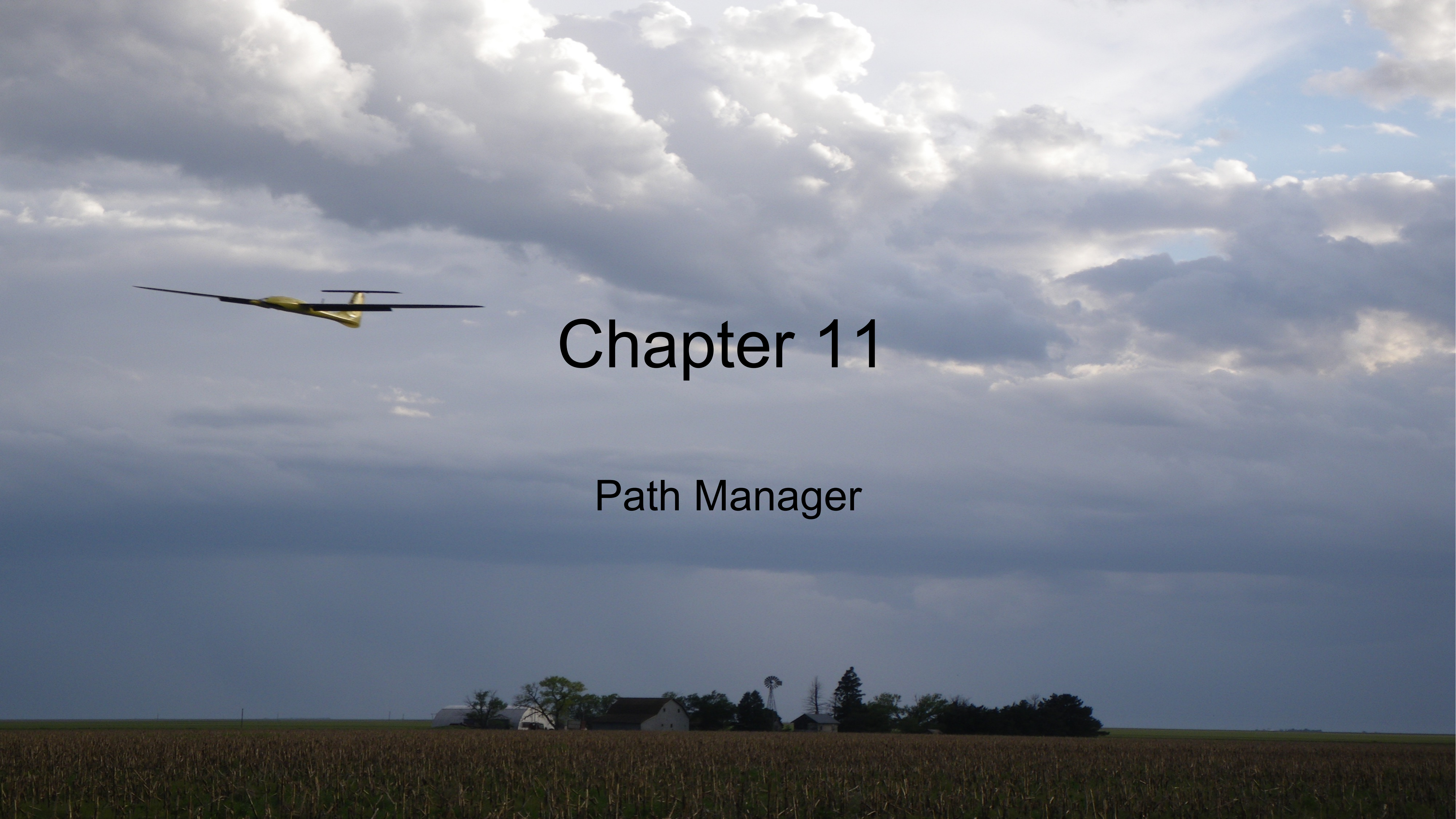
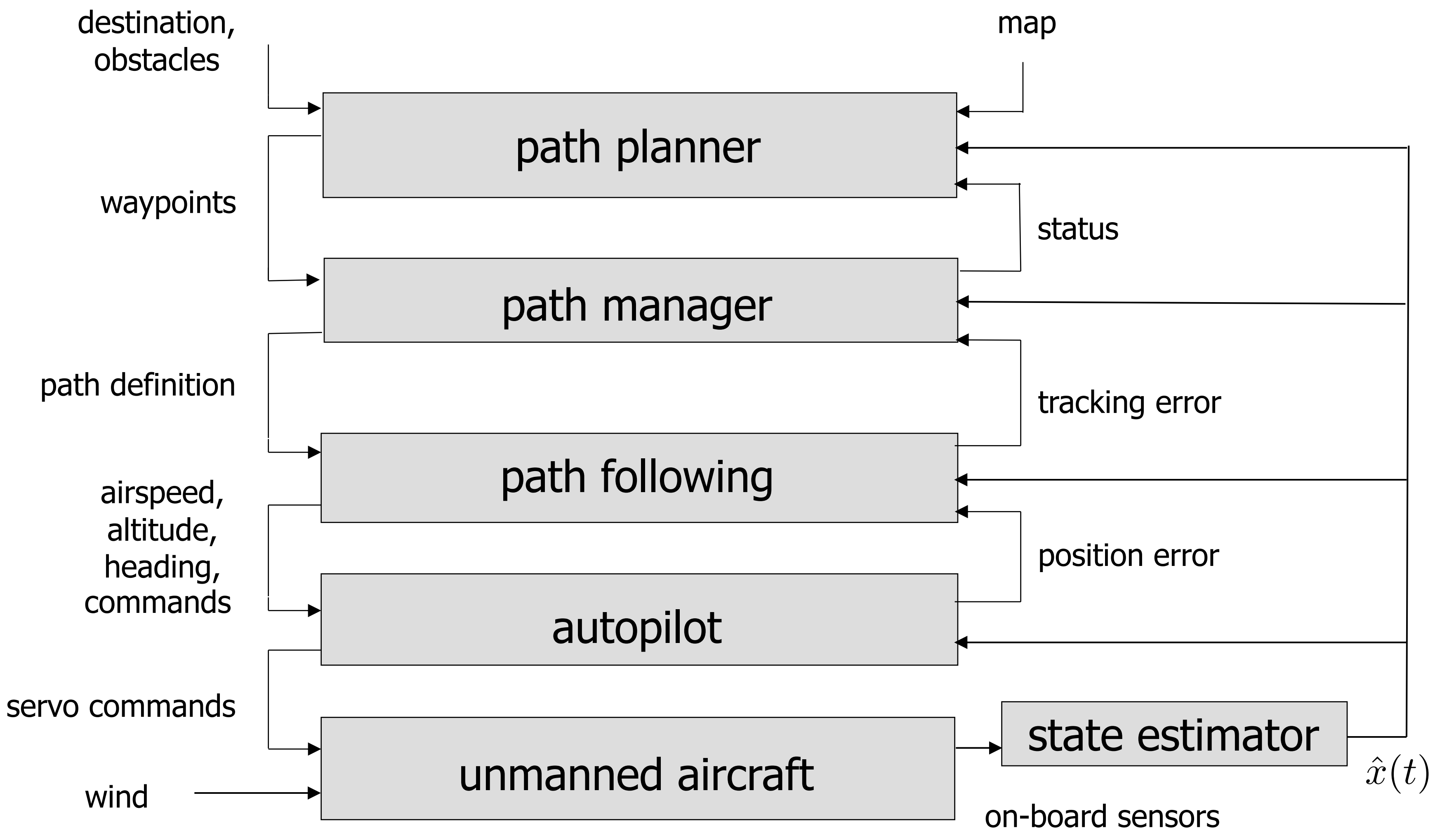


Chapter 11

Path Manager

Control Architecture

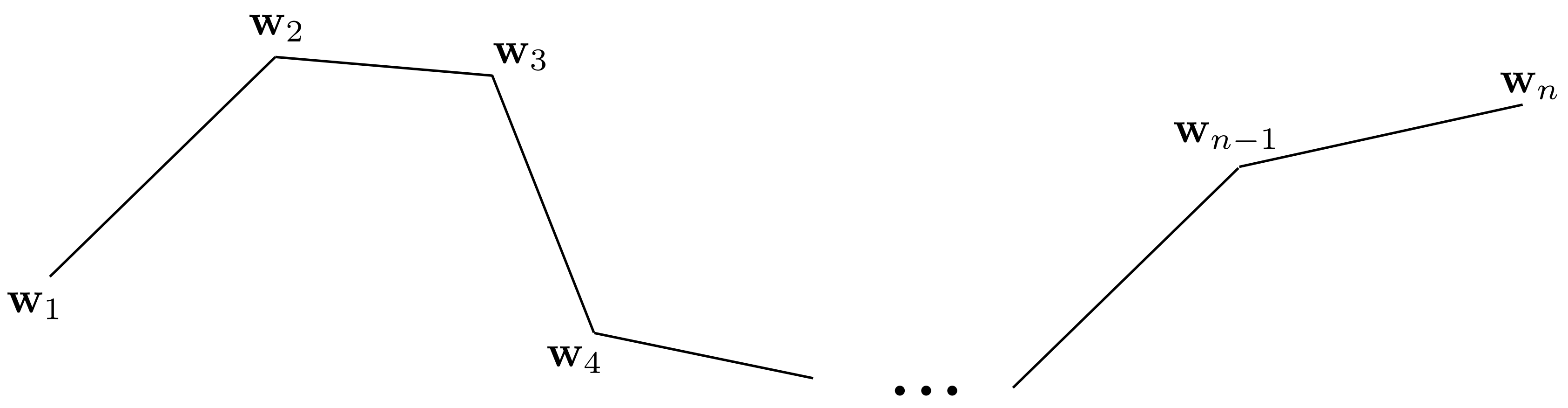


Path Definition

Waypoint path defined as ordered sequence of waypoints

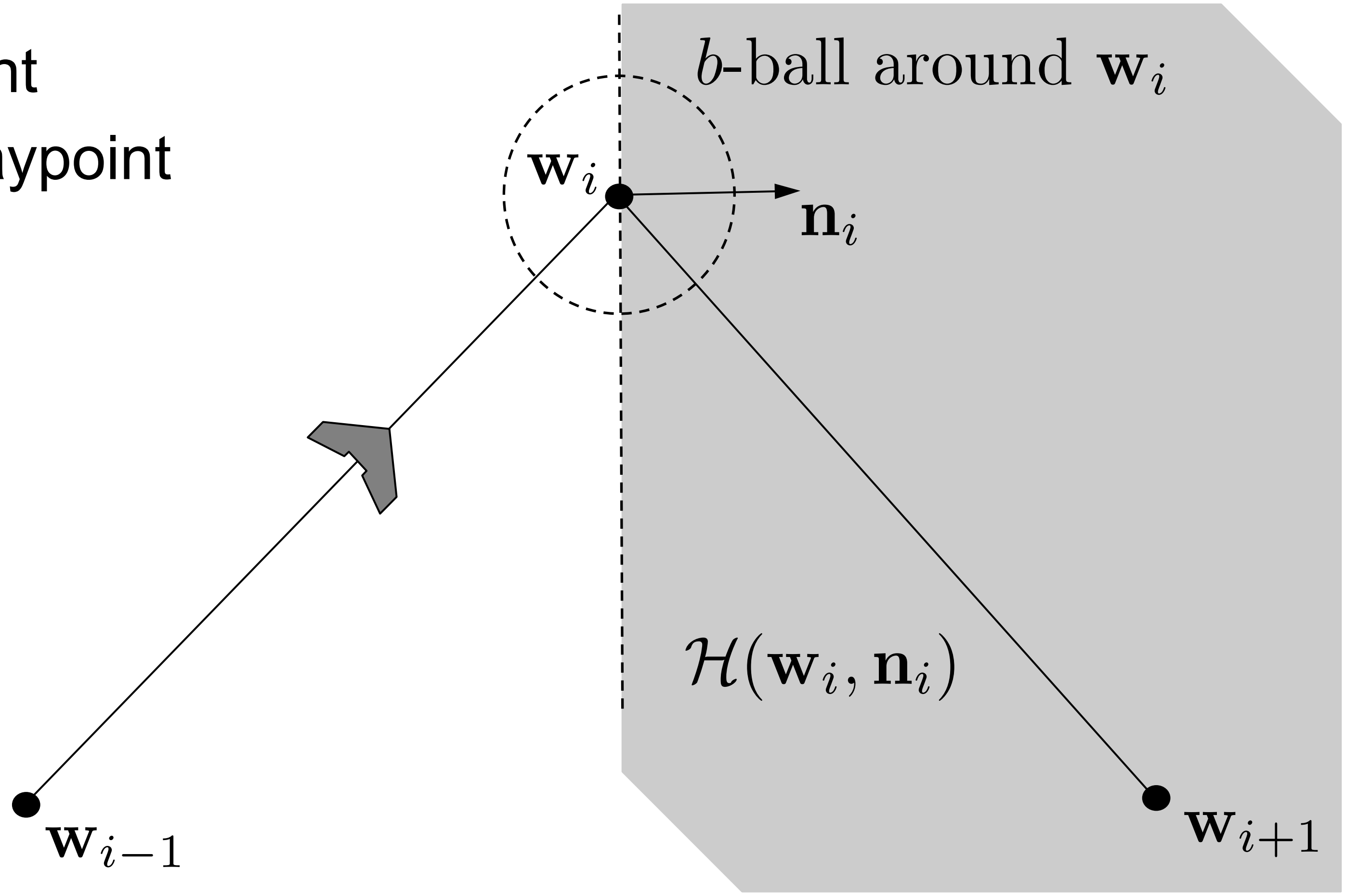
$$\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N\}$$

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

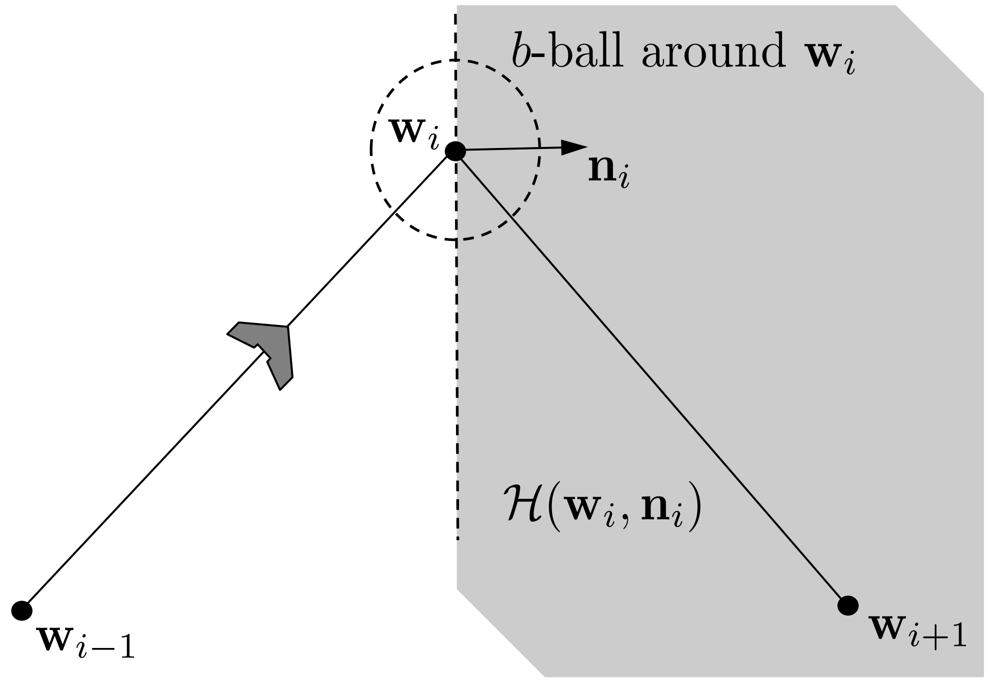


Waypoint Switching

- 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

$$\mathcal{H}(\mathbf{r}, \mathbf{n}) \stackrel{\triangle}{=} \left\{ \mathbf{p} \in \mathbb{R}^3 : (\mathbf{p} - \mathbf{r})^\top \mathbf{n} \geq 0 \right\}$$

Define unit vector pointing in direction of line $\overline{\mathbf{w}_i \mathbf{w}_{i+1}}$ as

$$\mathbf{q}_i \stackrel{\triangle}{=} \frac{\mathbf{w}_{i+1} - \mathbf{w}_i}{\|\mathbf{w}_{i+1} - \mathbf{w}_i\|}$$

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

$$\mathbf{n}_i \stackrel{\triangle}{=} \frac{\mathbf{q}_{i-1} + \mathbf{q}_i}{\|\mathbf{q}_{i-1} + \mathbf{q}_i\|}$$

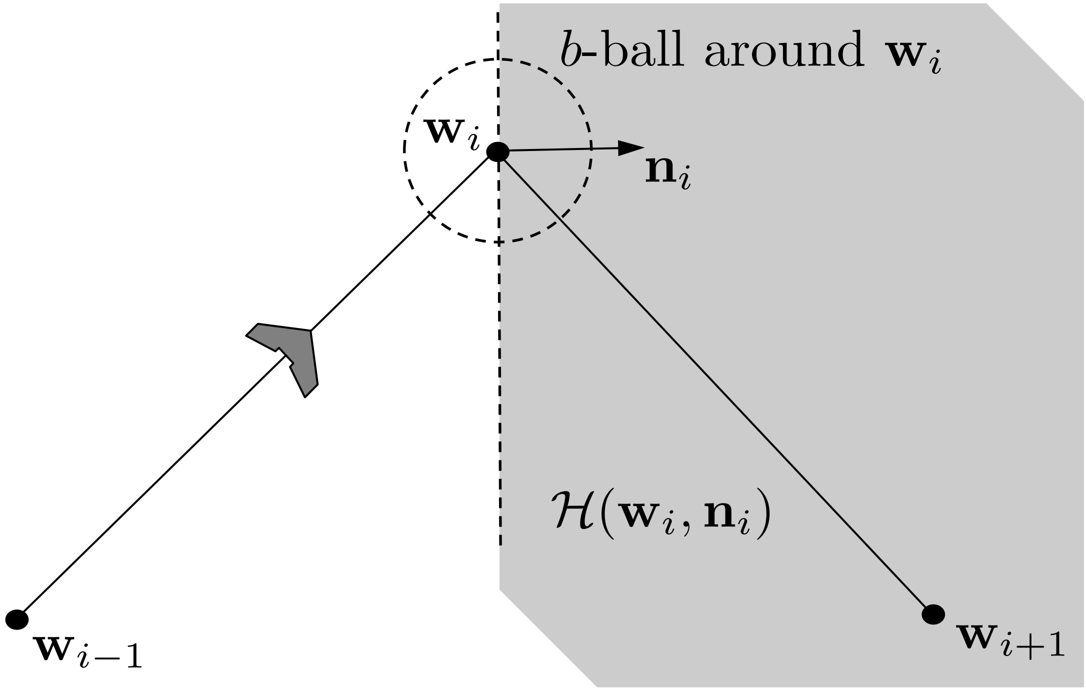
MAV tracks straight-line path from $\mathbf{w}_{i-1}$ to $\mathbf{w}_i$ until it enters $\mathcal{H}(\mathbf{w}_i, \mathbf{n}_i)$, at which point it will track straight-line path from $\mathbf{w}_i$ to $\mathbf{w}_{i+1}$

Waypoint Following

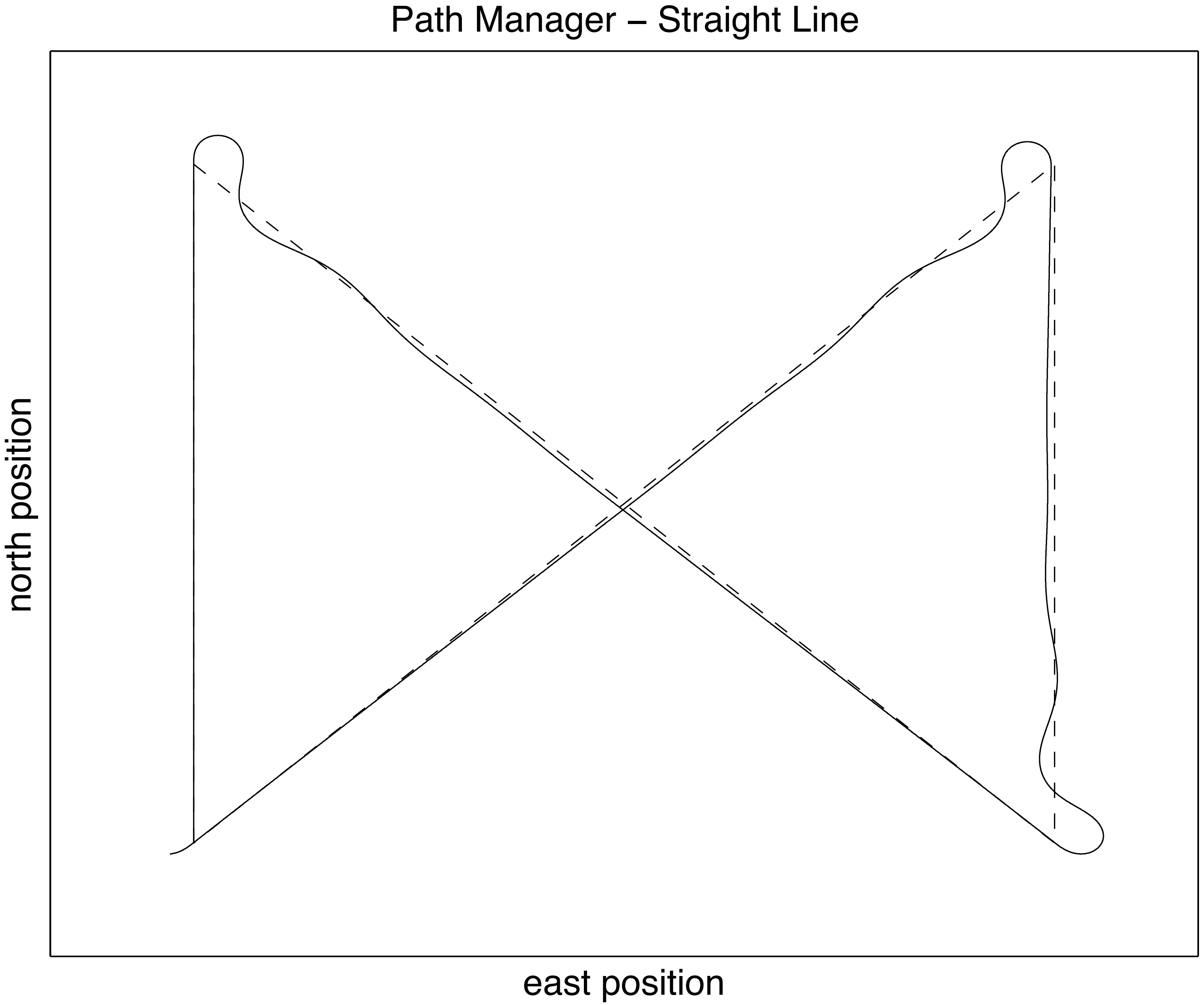
Algorithm 5 Follow Waypoints: $(\mathbf{r}, \mathbf{q}) = \text{followWpp}(\mathcal{W}, \mathbf{p})$

Input: Waypoint path $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$, MAV position $\mathbf{p} = (p_n, p_e, p_d)^\top$.
Require: $N \geq 3$

if New waypoint path $\mathcal{W}$ is received **then**
  Initialize waypoint index: $i \leftarrow 2$
end if
$\mathbf{r} \leftarrow \mathbf{w}_{i-1}$
$\mathbf{q}_{i-1} \leftarrow \frac{\mathbf{w}_i - \mathbf{w}_{i-1}}{\|\mathbf{w}_i - \mathbf{w}_{i-1}\|}$
$\mathbf{q}_i \leftarrow \frac{\mathbf{w}_{i+1} - \mathbf{w}_i}{\|\mathbf{w}_{i+1} - \mathbf{w}_i\|}$
$\mathbf{n}_i \leftarrow \frac{\mathbf{q}_{i-1} + \mathbf{q}_i}{\|\mathbf{q}_{i-1} + \mathbf{q}_i\|}$
if $\mathbf{p} \in \mathcal{H}(\mathbf{w}_i, \mathbf{n}_i)$ **then**
  Increment $i \leftarrow (i+1)$ until $i = N - 1$
end if
return $\mathbf{r}, \mathbf{q} = \mathbf{q}_{i-1}$ at each time step

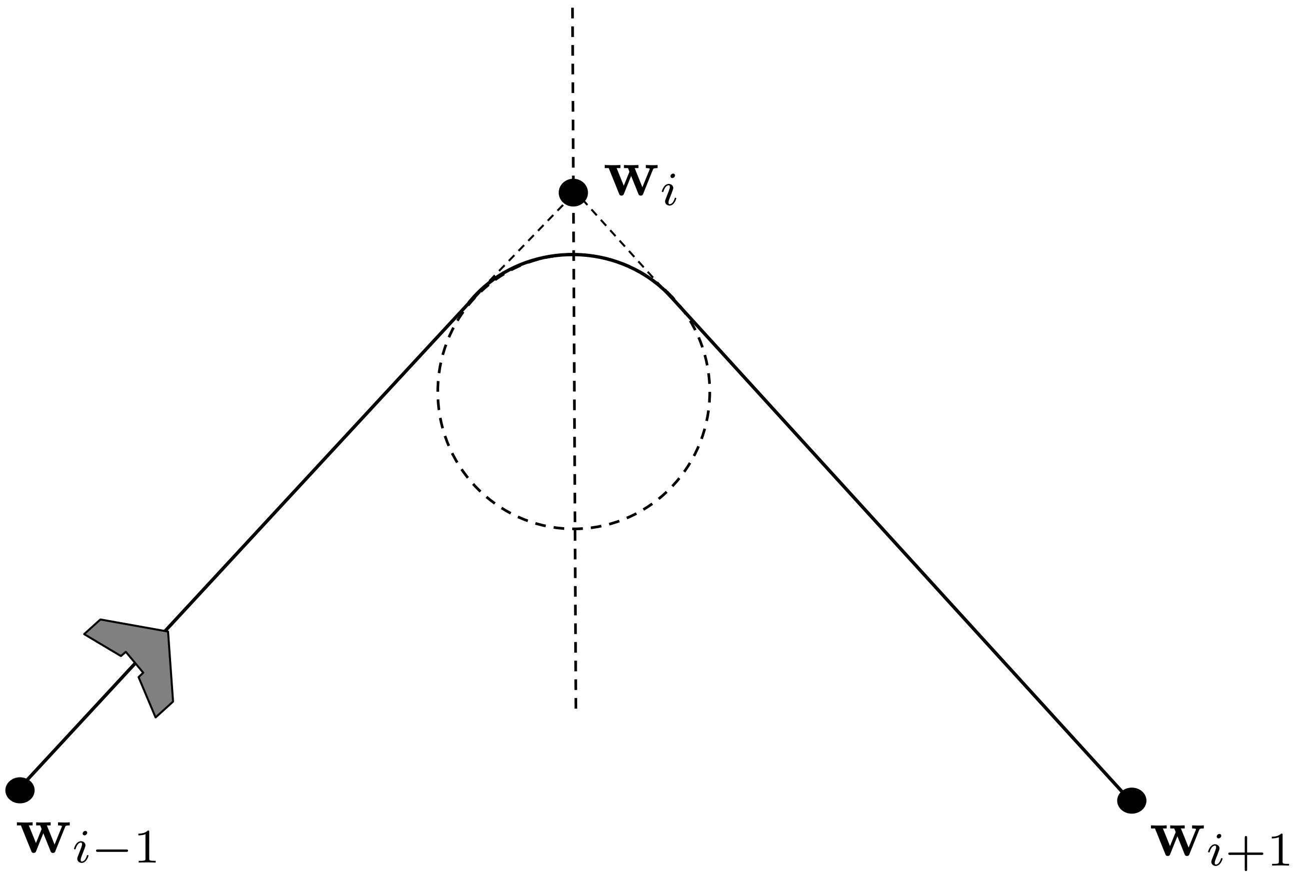


Waypoint Following Results

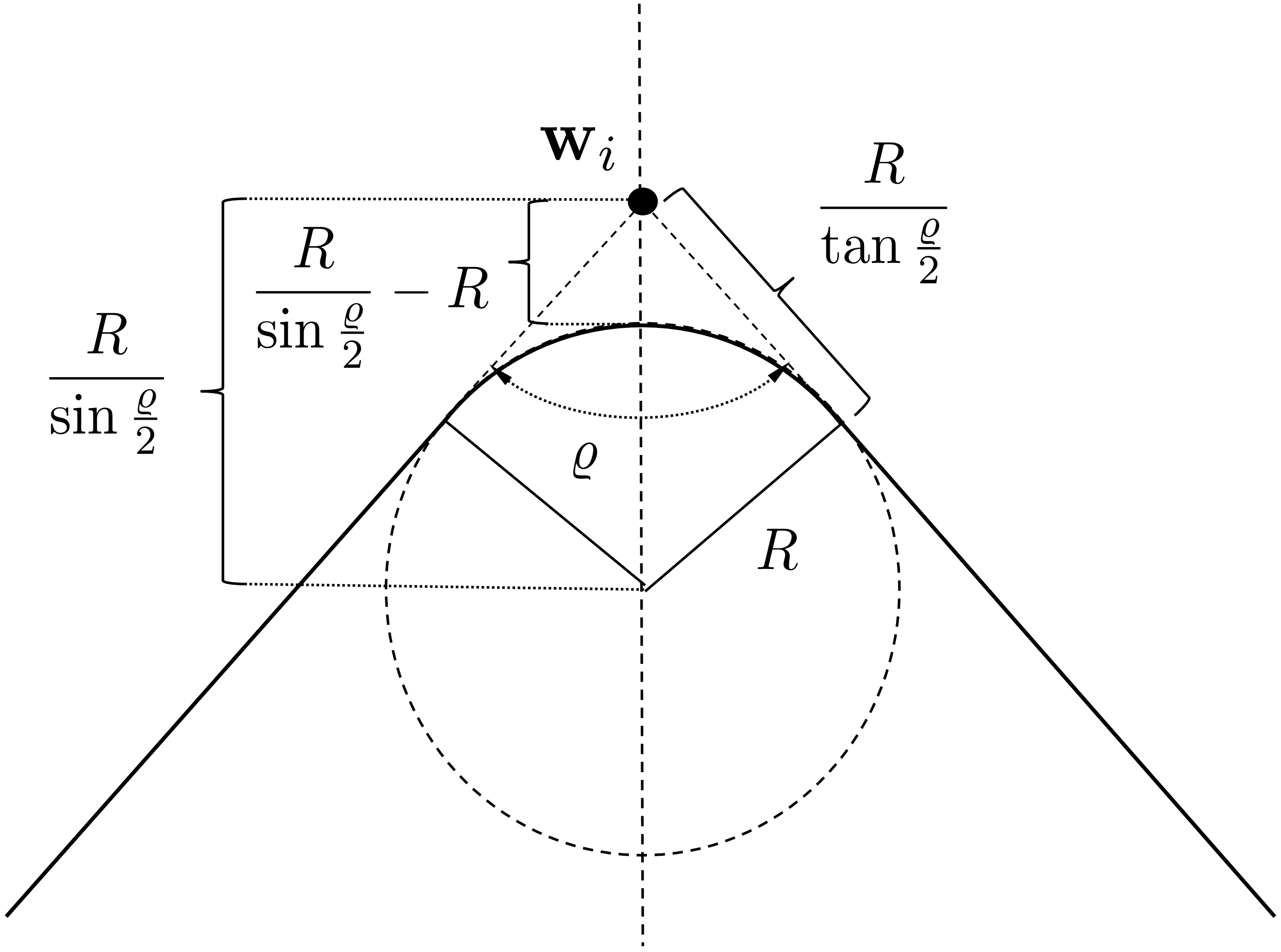


Fillet Transition

Transition between path segments can be smoothed by adding a fillet



Fillet Geometry



Fillet Smoothing Half Planes

Fillet center defined as

$$\mathbf{c} = \mathbf{w}_i - \left(\frac{R}{\sin\frac{\varrho}{2}}\right)\frac{\mathbf{q}_{i-1} - \mathbf{q}_i}{\|\mathbf{q}_{i-1} - \mathbf{q}_i\|}$$

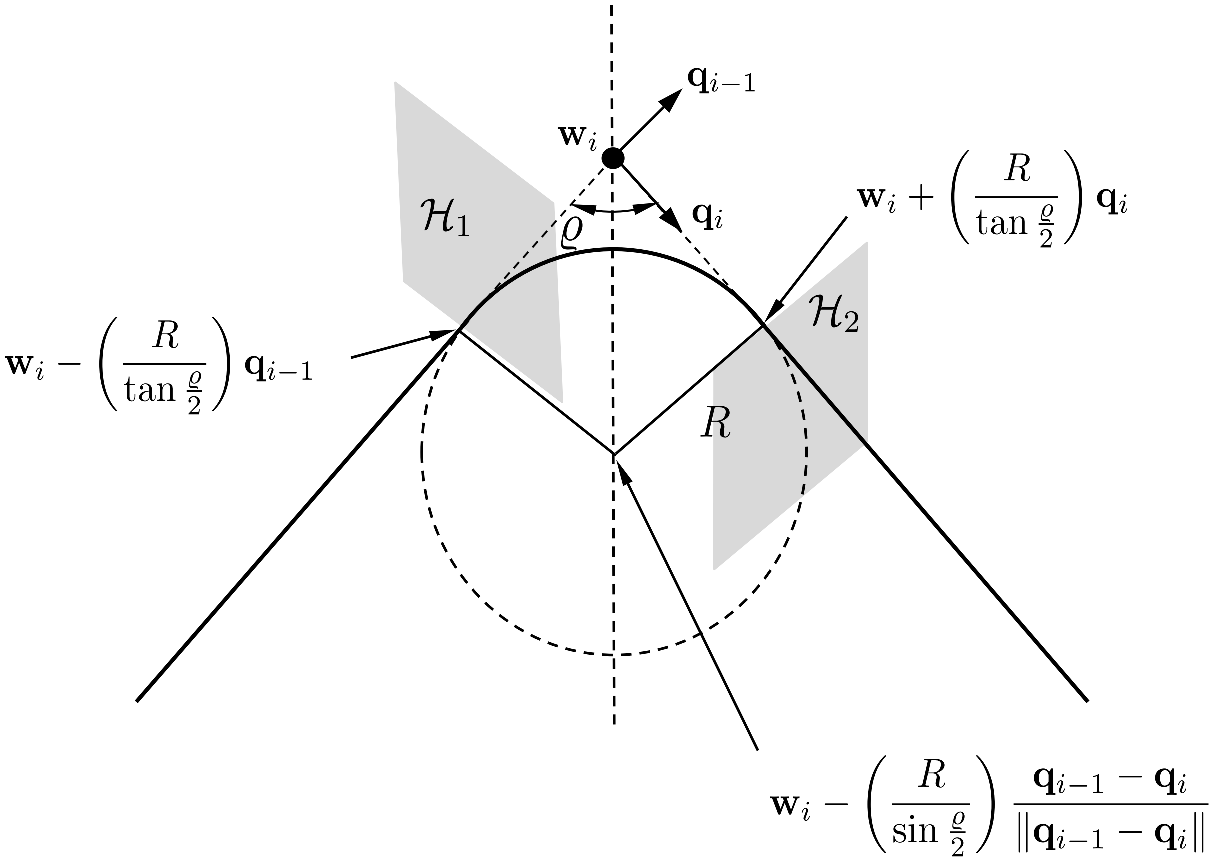
Half plane $\mathcal{H}_1$ is defined by location

$$\mathbf{r}_1 = \mathbf{w}_i - \left(\frac{R}{\tan\frac{\varrho}{2}}\right)\mathbf{q}_{i-1}$$

and normal vector $\mathbf{q}_{i-1}$
Half plane $\mathcal{H}_2$ is defined by location

$$\mathbf{r}_2 = \mathbf{w}_i + \left(\frac{R}{\tan\frac{\varrho}{2}}\right)\mathbf{q}_i$$

and normal vector $\mathbf{q}_i$



Waypoint Following with Fillets

Algorithm 1 Follow Waypoints with Fillets: $(\text{flag}, \mathbf{r}, \mathbf{q}, \mathbf{c}, \rho, \lambda) = \mathsf{followWppFillet}(\mathcal{W}, \mathbf{p}, R)$

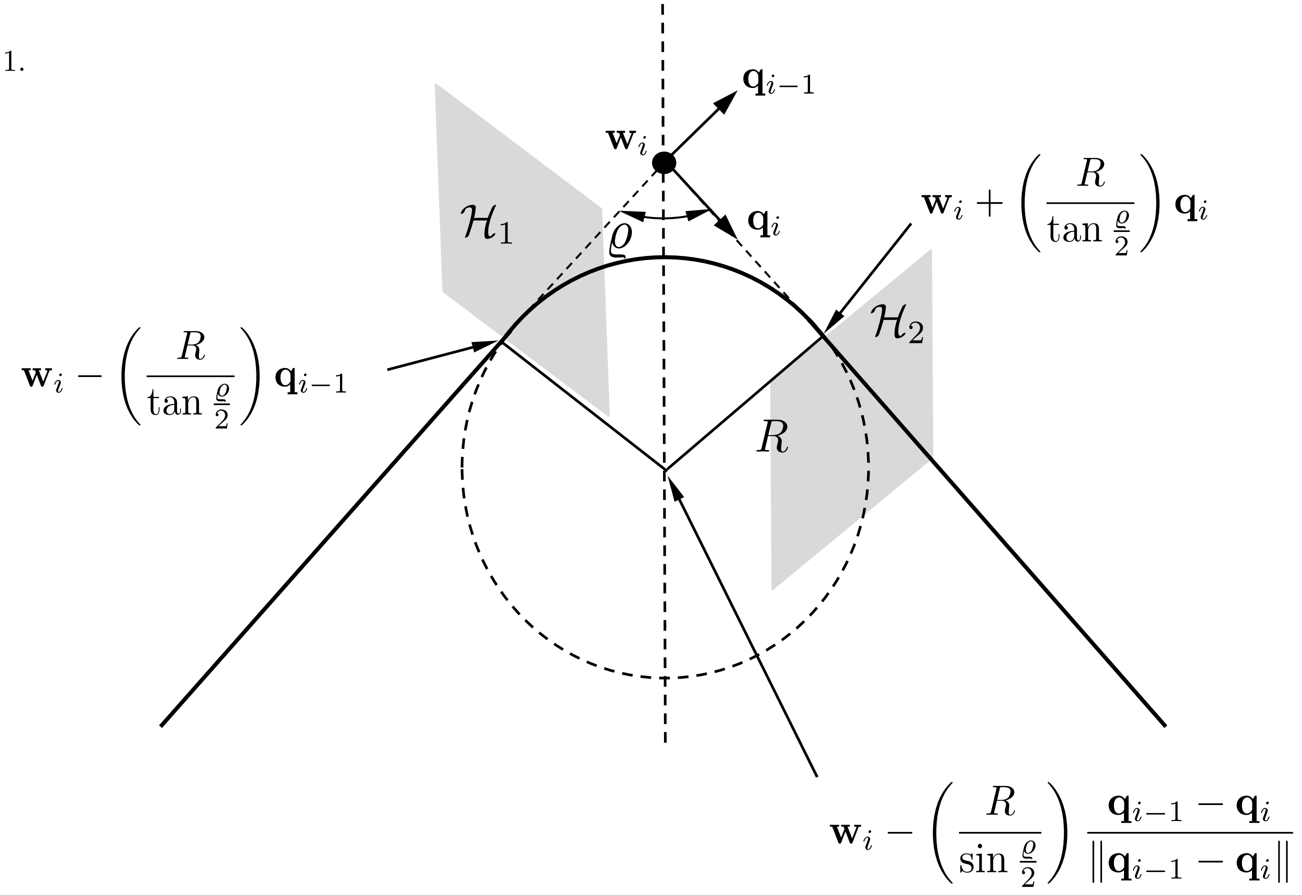
Ensure: Waypoint path $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$, MAV position $\mathbf{p} = (p_n, p_e, p_d)^\top$, fillet radius R.
Require: $N \geq 3$.

```
if New waypoint path W is received then
   Initialize waypoint index: i ← 2, and state machine: state ← 1.
end if
q_{i-1} ← (w_i − w_{i-1}) / ‖w_i − w_{i-1}‖.
q_i ← (w_{i+1} − w_i) / ‖w_{i+1} − w_i‖.
ϱ ← cos^{-1}(−q_{i-1}^⊤ q_i).
if state = 1 then
   flag ← 1
   r ← w_{i-1}
   q ← q_{i-1}
   z ← w_i − (R / tan(ϱ/2)) q_{i-1}
   if p ∈ H(z, q_{i-1}) then
      state ← 2
   end if
else if state = 2 then
   flag ← 2
   c ← w_i − (R / sin(ϱ/2)) (q_{i-1} − q_i) / ‖q_{i-1} − q_i‖
   ρ ← R
   λ ← sign(q_{i-1,n} q_{i,e} − q_{i-1,e} q_{i,n}).
   z ← w_i + (R / tan(ϱ/2)) q_i
   if p ∈ H(z, q_i) then
      i ← (i + 1) until i = N − 1.
      state ← 1
   end if
end if
return flag, r, q, c, ρ, λ.
```

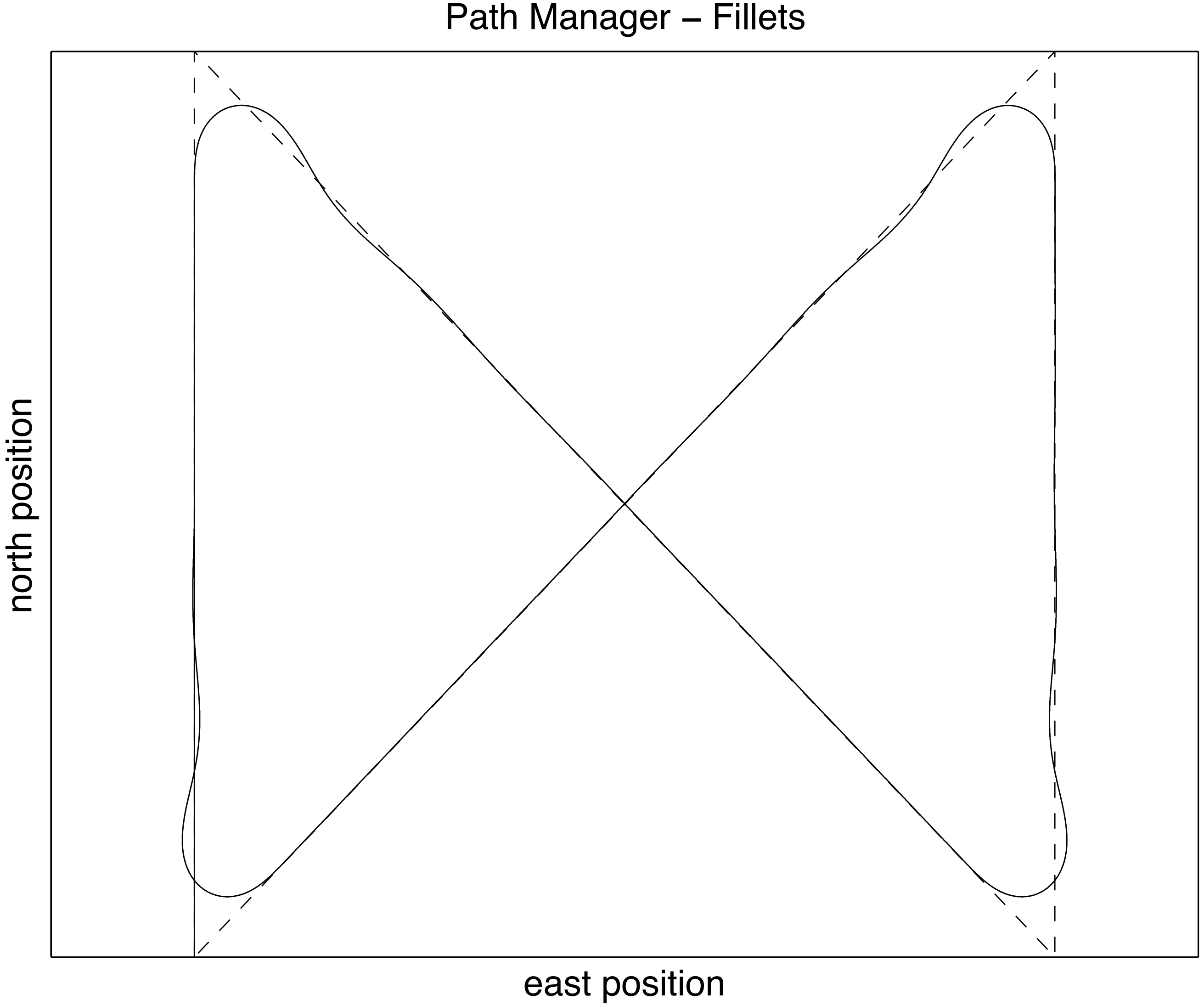
Where:

$$\mathbf{q}_{i-1} \leftarrow \frac{\mathbf{w}_i - \mathbf{w}_{i-1}}{\|\mathbf{w}_i - \mathbf{w}_{i-1}\|}, \quad \mathbf{q}_i \leftarrow \frac{\mathbf{w}_{i+1} - \mathbf{w}_i}{\|\mathbf{w}_{i+1} - \mathbf{w}_i\|}, \quad \varrho \leftarrow \cos^{-1}\left(-\mathbf{q}_{i-1}^\top \mathbf{q}_i\right)$$

$$\mathbf{z} \leftarrow \mathbf{w}_i - \left(\frac{R}{\tan(\varrho/2)}\right)\mathbf{q}_{i-1}, \quad \mathbf{c} \leftarrow \mathbf{w}_i - \left(\frac{R}{\sin(\varrho/2)}\right)\frac{\mathbf{q}_{i-1} - \mathbf{q}_i}{\|\mathbf{q}_{i-1} - \mathbf{q}_i\|}, \quad \mathbf{z} \leftarrow \mathbf{w}_i + \left(\frac{R}{\tan(\varrho/2)}\right)\mathbf{q}_i$$



Waypoint Following with Fillets



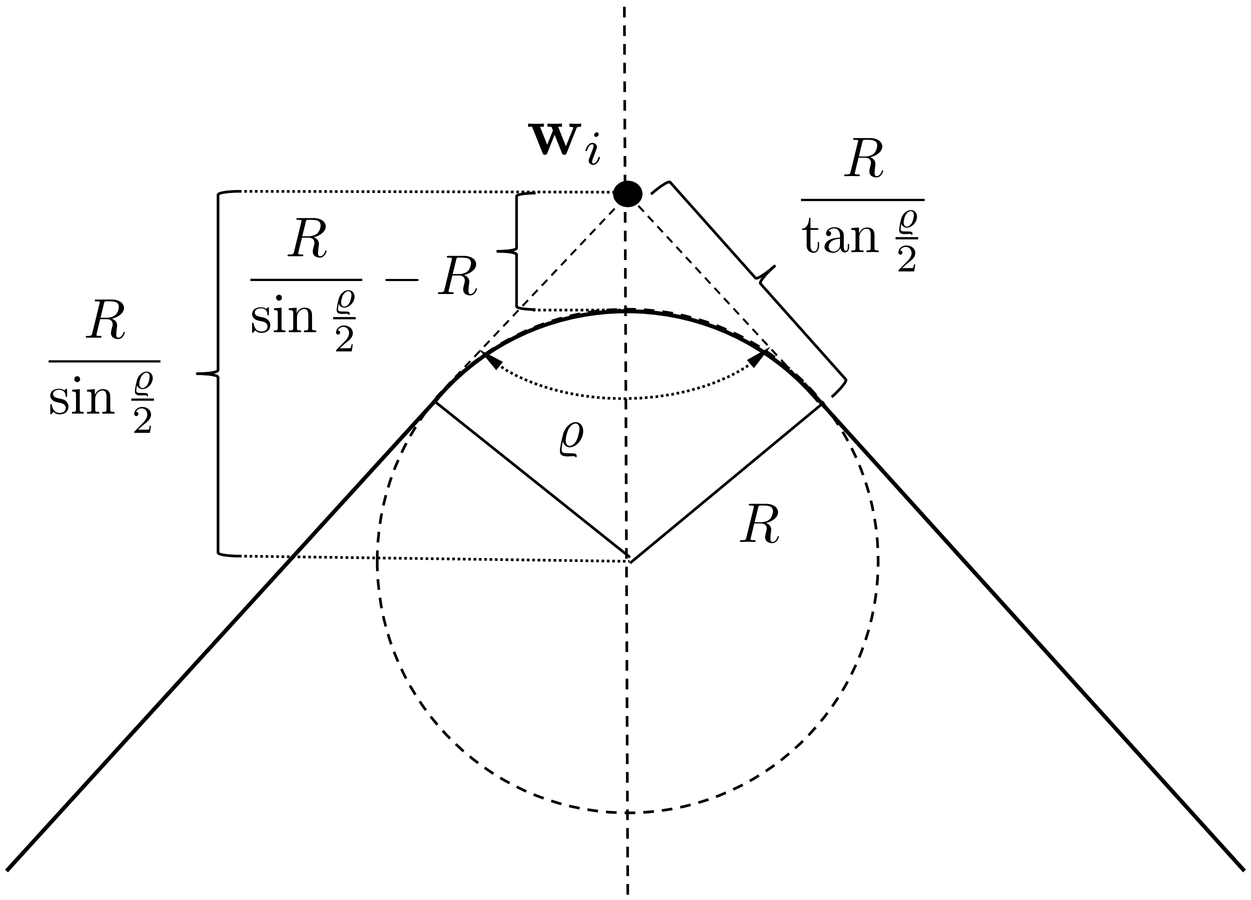
Fillet Path Length

Straight-line path length (no fillets):

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

Path length with fillets:

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

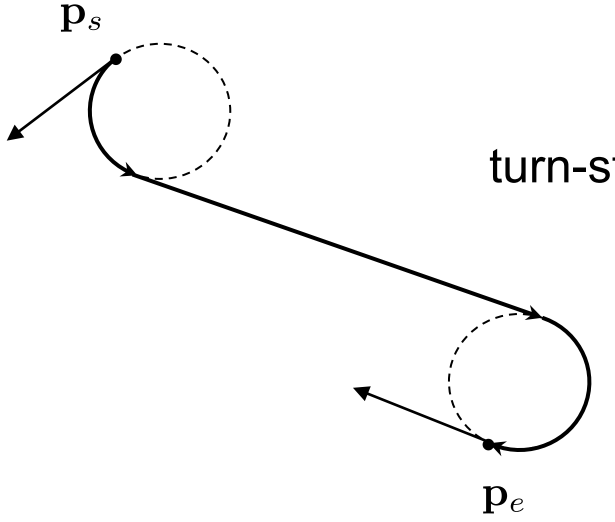


Dubins Paths

For vehicle with kinematics given by

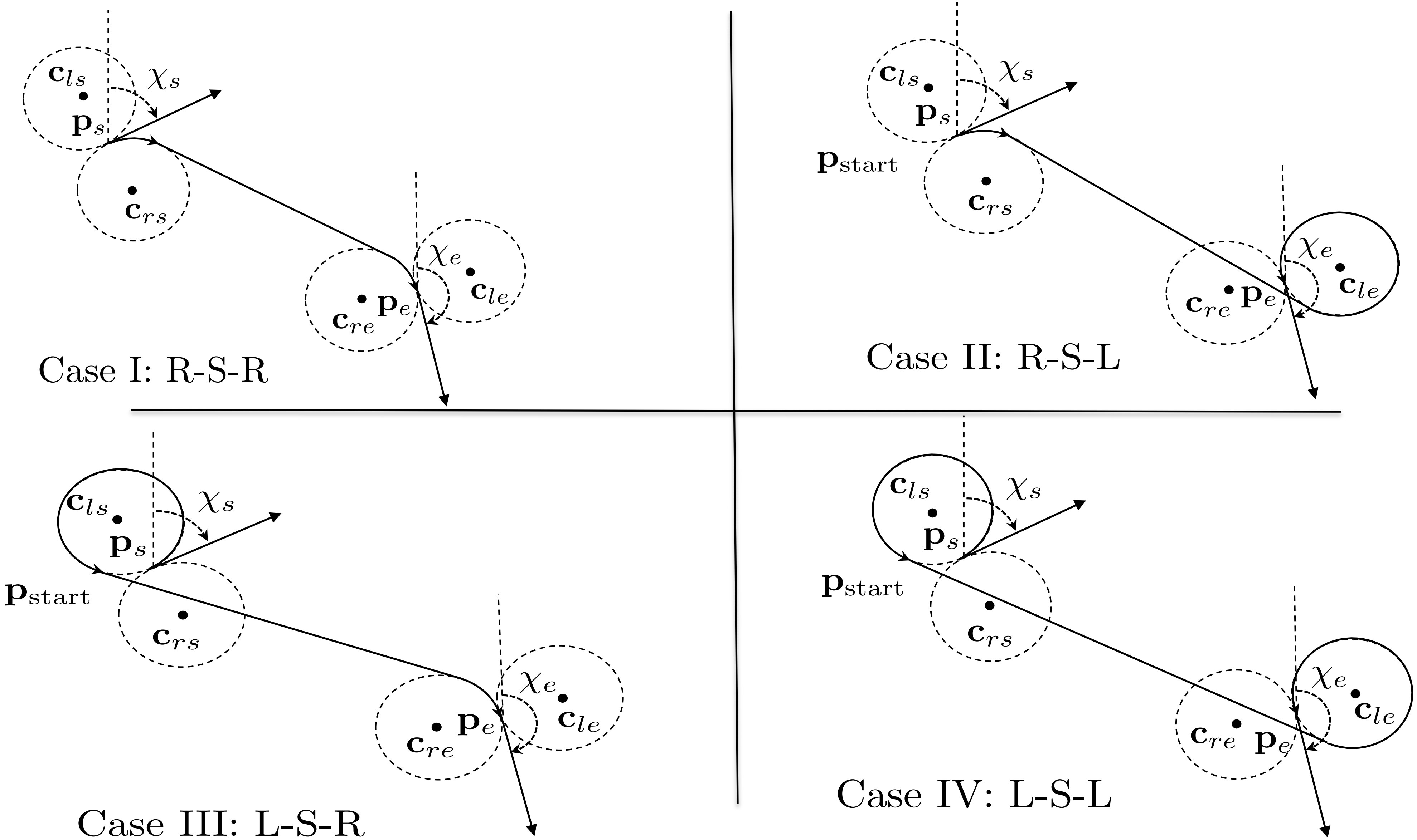
$$
\begin{aligned}
\dot{p}_n &= V \cos \vartheta \\
\dot{p}_e &= V \sin \vartheta \\
\dot{\vartheta} &= u,
\end{aligned}
$$

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 $V/\bar{u}$.



turn-straight-turn

Dubins path is defined as shortest of four cases

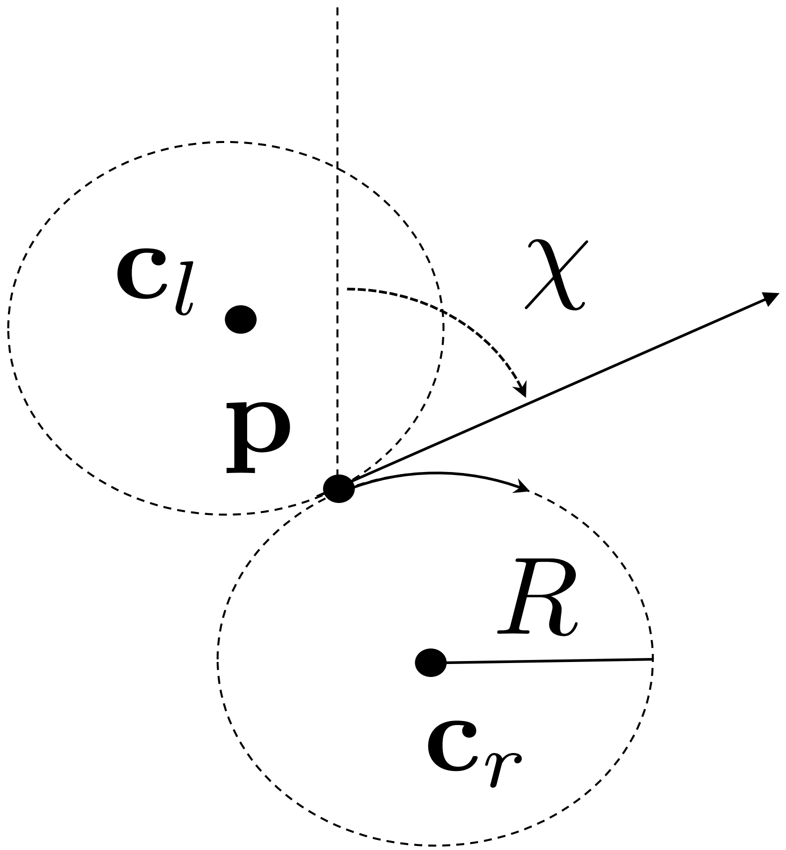


Path Length Preliminaries

Given position $\mathbf{p}$, course χ, and radius R, centers of right and left turning circles are given by

$$\mathbf{c}_r = \mathbf{p} + R\left(\cos(\chi + \tfrac{\pi}{2}), \quad \sin(\chi + \tfrac{\pi}{2}), \quad 0\right)^\top$$

$$\mathbf{c}_l = \mathbf{p} + R\left(\cos(\chi - \tfrac{\pi}{2}), \quad \sin(\chi - \tfrac{\pi}{2}), \quad 0\right)^\top$$



Path Length Preliminaries

For clockwise circles, angular distance between ϑ_1 and ϑ_2 given by

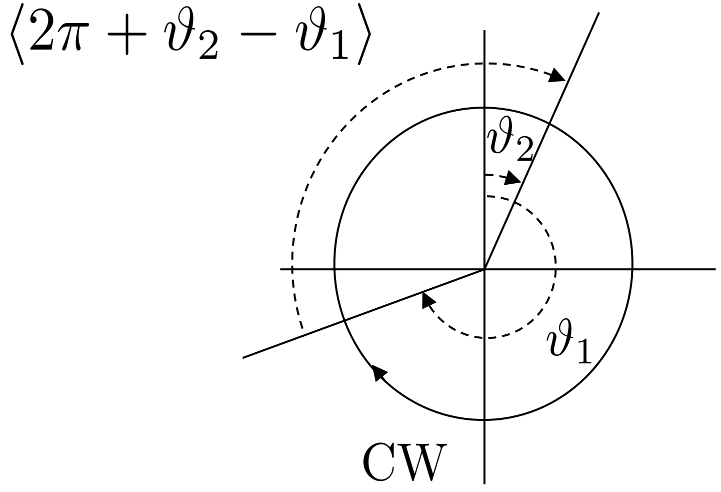
$$|\vartheta_2 - \vartheta_1|_{CW} \stackrel{\triangle}{=} \langle 2\pi + \vartheta_2 - \vartheta_1 \rangle,$$

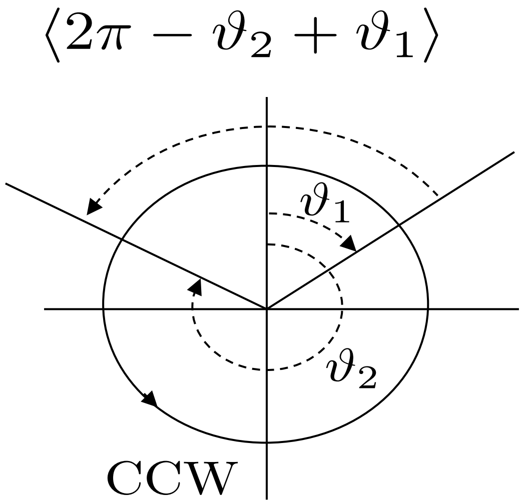
where

$$\langle \varphi \rangle \stackrel{\triangle}{=} \varphi \mod 2\pi$$

For counter clockwise circles,

$$|\vartheta_2 - \vartheta_1|_{CCW} \stackrel{\triangle}{=} \langle 2\pi - \vartheta_2 + \vartheta_1 \rangle$$





Alternative Viewpoint

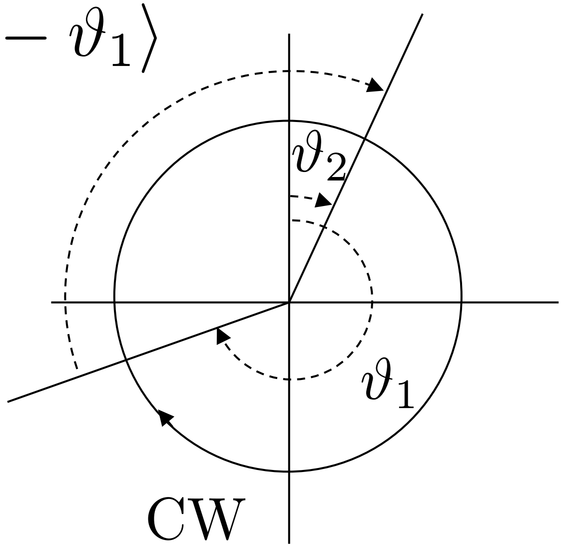
For clockwise circles, angular distance between ϑ_1 and ϑ_2 given by

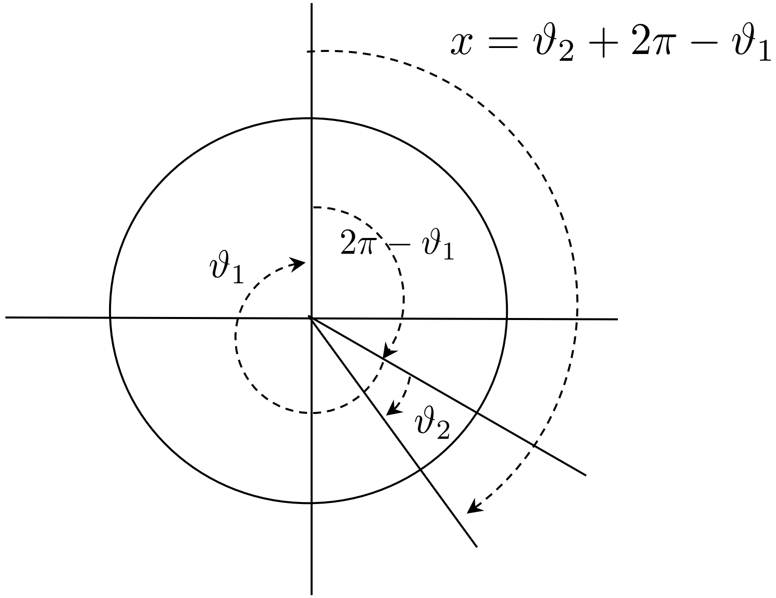
$$|\vartheta_2 - \vartheta_1|_{CW} \stackrel{\triangle}{=} \langle 2\pi + \vartheta_2 - \vartheta_1 \rangle,$$

where

$$\langle \varphi \rangle \stackrel{\triangle}{=} \varphi \mod 2\pi$$

$\langle 2\pi + \vartheta_2 - \vartheta_1 \rangle$

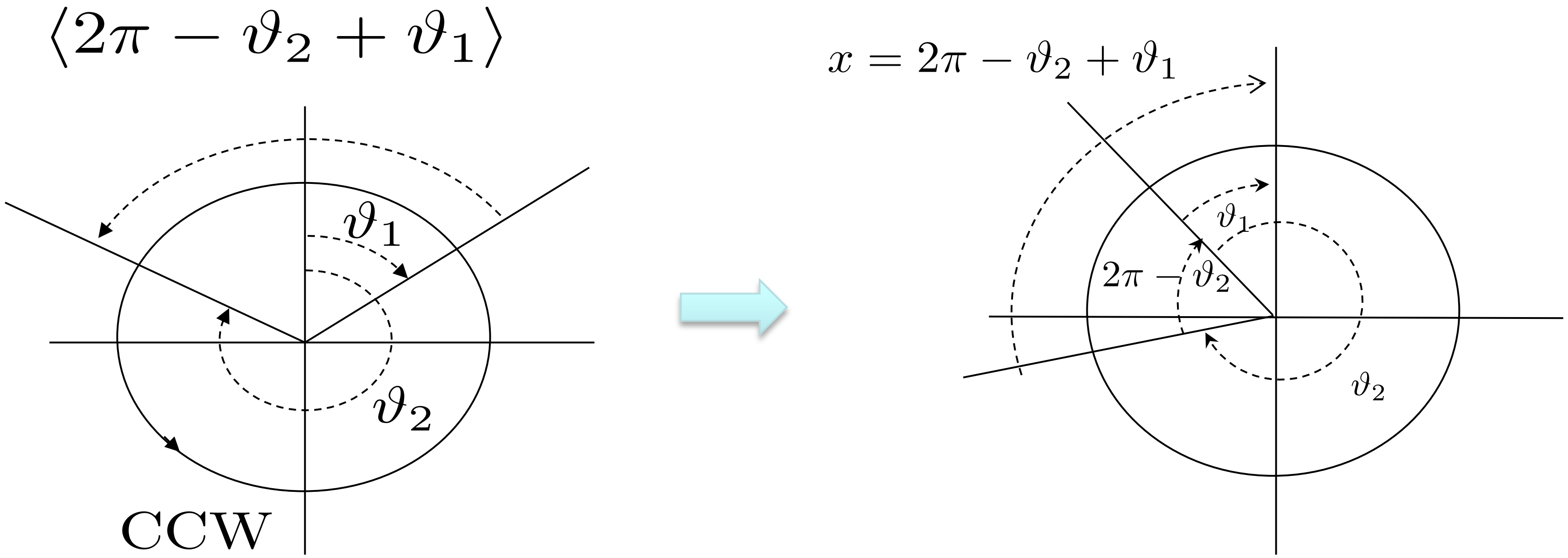




Alternative Viewpoint

For counter clockwise circles,

$$|\vartheta_2 - \vartheta_1|_{CCW} \stackrel{\triangle}{=} \langle 2\pi - \vartheta_2 + \vartheta_1 \rangle$$



Dubins Case I: R-S-R

Distance traveled along $\mathbf{c}_{rs}$

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

Distance traveled along $\mathbf{c}_{re}$

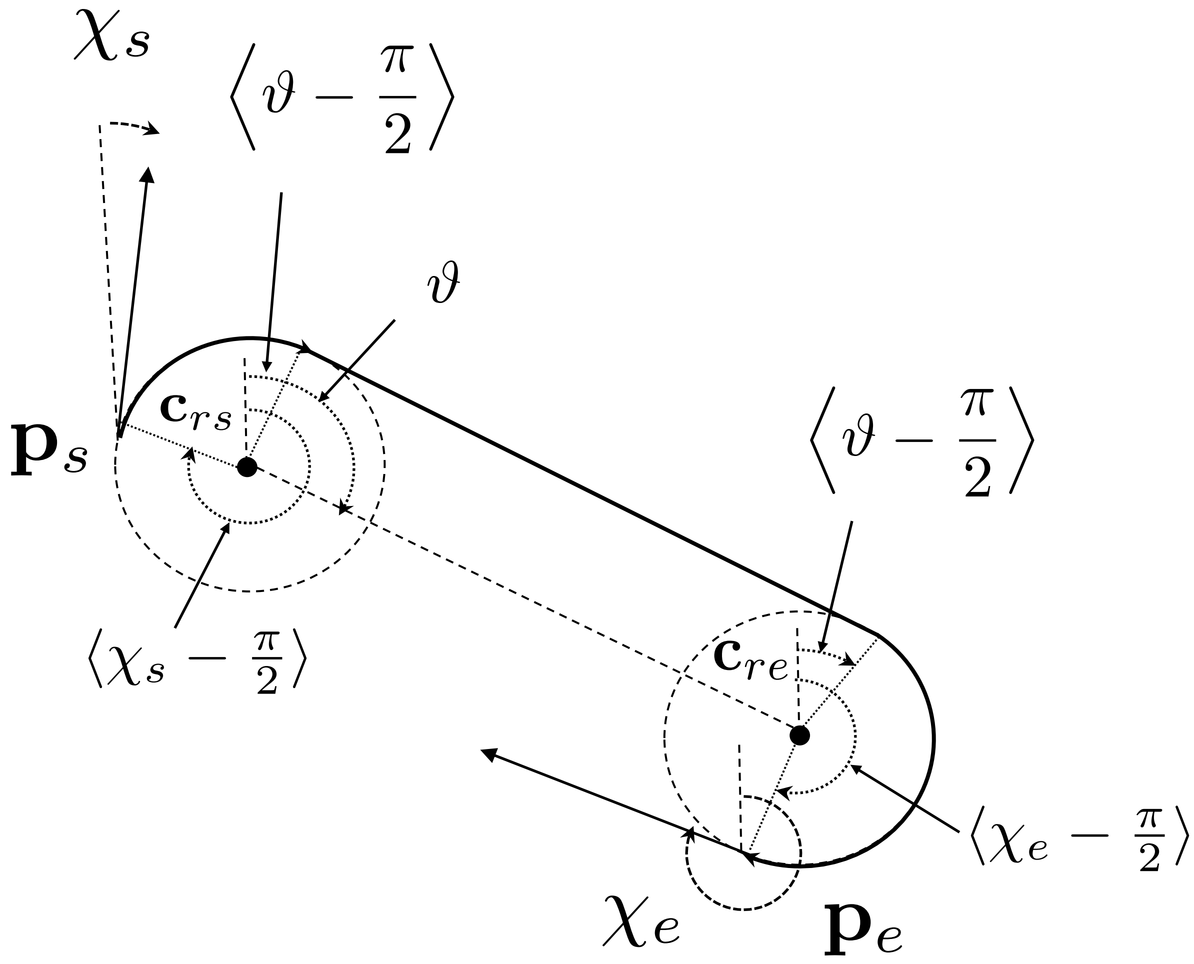
$$R\langle 2\pi + \langle \chi_e - \frac{\pi}{2}\rangle - \langle \vartheta - \frac{\pi}{2}\rangle\rangle$$

where

$$\vartheta = \text{atan2}(c_{ree} - c_{rse}, c_{ren} - c_{rsn})$$

Total path length:

$$L_1 = \|\mathbf{c}_{rs} - \mathbf{c}_{re}\| + R\langle 2\pi + \langle \vartheta - \frac{\pi}{2}\rangle - \langle \chi_s - \frac{\pi}{2}\rangle\rangle + R\langle 2\pi + \langle \chi_e - \frac{\pi}{2}\rangle - \langle \vartheta - \frac{\pi}{2}\rangle\rangle$$



Dubins Case II: R-S-L

Distance traveled along $\mathbf{c}_{rs}$

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

Distance traveled along $\mathbf{c}_{le}$

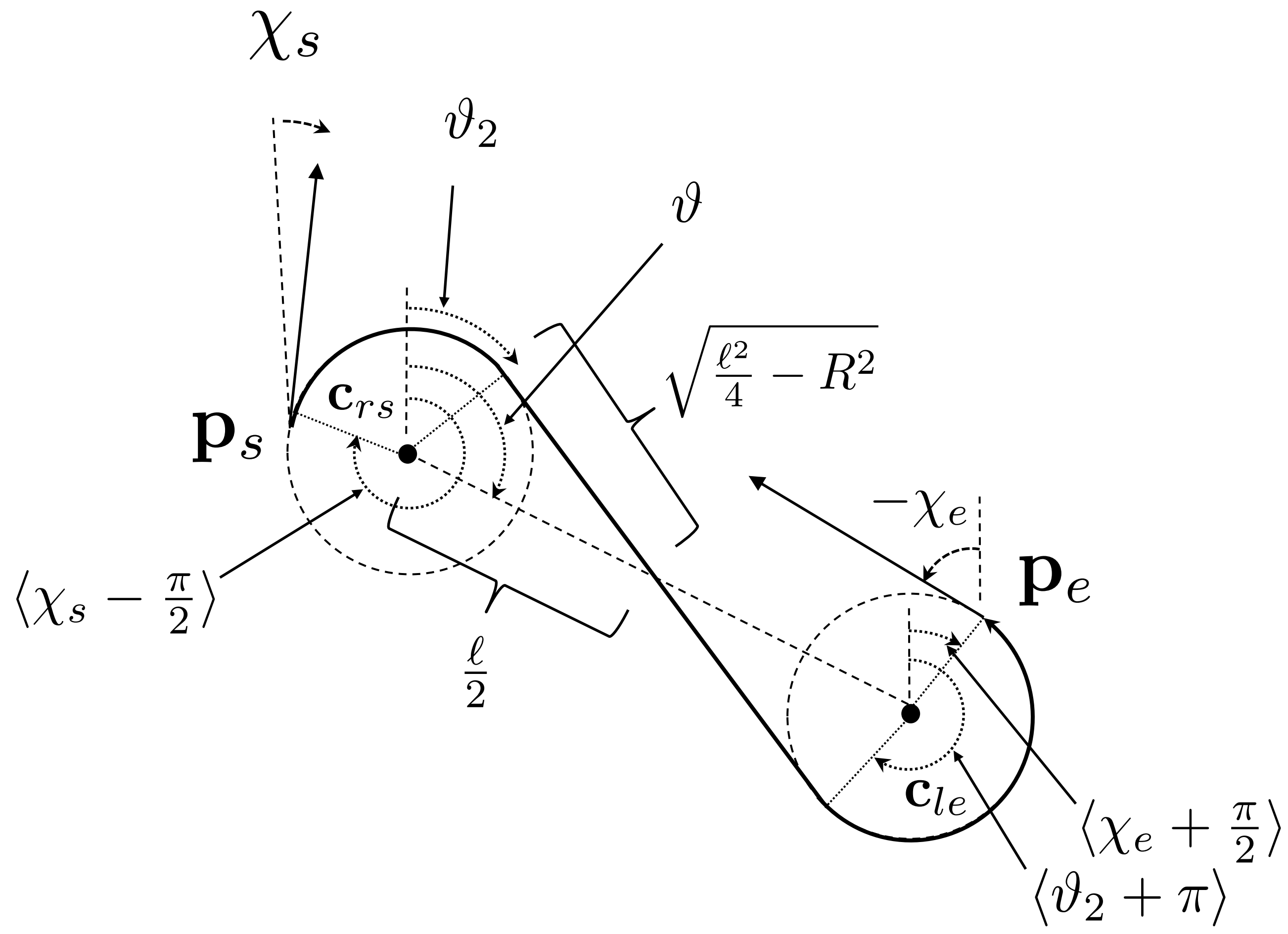
$$R\langle 2\pi + \langle \vartheta_2 + \pi \rangle - \langle \chi_e + \frac{\pi}{2} \rangle \rangle$$

where

$$\vartheta_2 = \vartheta - \frac{\pi}{2} + \sin^{-1}\left(\frac{2R}{\ell}\right)$$

Total path length:

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



Dubins Case III: L-S-R

Distance traveled along $\mathbf{c}_{ls}$

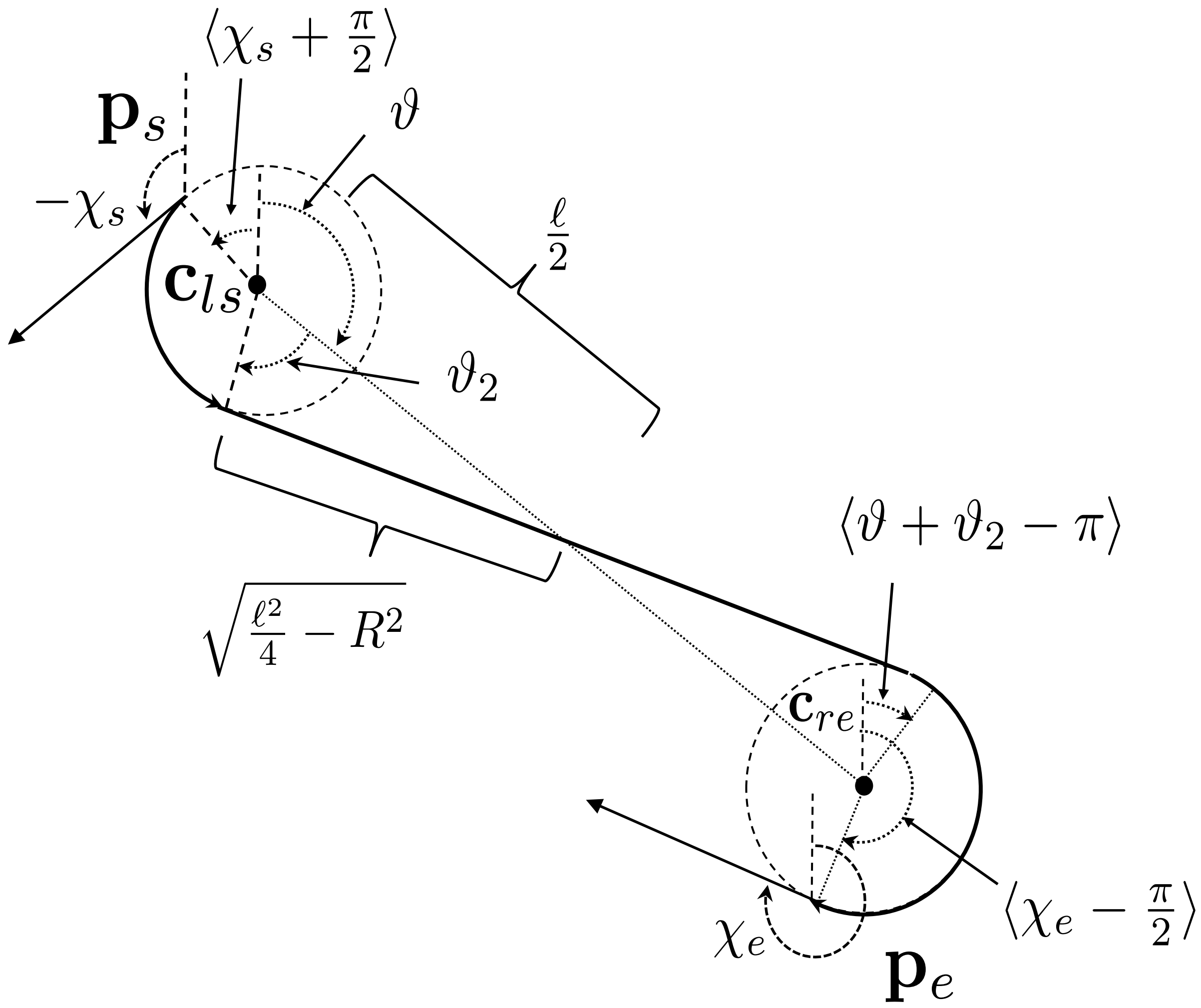
$$R\langle 2\pi + \langle \chi_s + \frac{\pi}{2}\rangle - \langle \vartheta + \vartheta_2\rangle\rangle$$

Distance traveled along $\mathbf{c}_{re}$

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

where

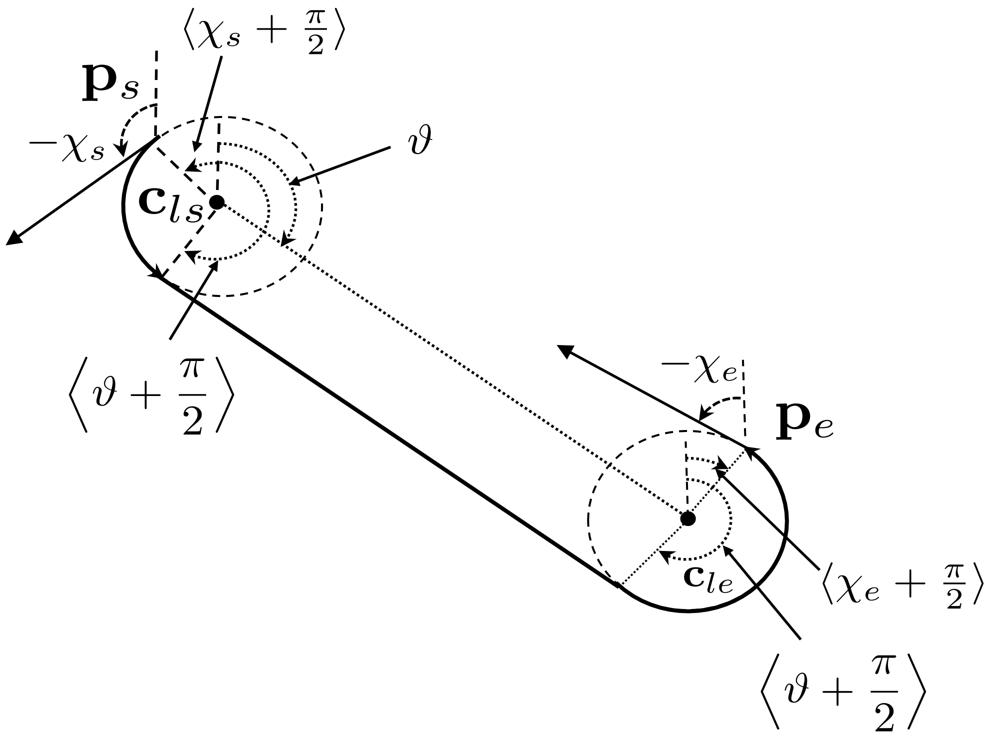
$$\vartheta_2 = \cos^{-1}\left(\frac{2R}{\ell}\right)$$



Total path length:

$$L_3 = \sqrt{\ell^2 - 4R^2} + R\langle 2\pi + \langle \chi_s + \frac{\pi}{2}\rangle - \langle \vartheta + \vartheta_2\rangle\rangle + R\langle 2\pi + \langle \chi_e - \frac{\pi}{2}\rangle - \langle \vartheta + \vartheta_2 - \pi\rangle\rangle$$

Dubins Case IV: L-S-L



Distance traveled along $\mathbf{c}_{ls}$

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

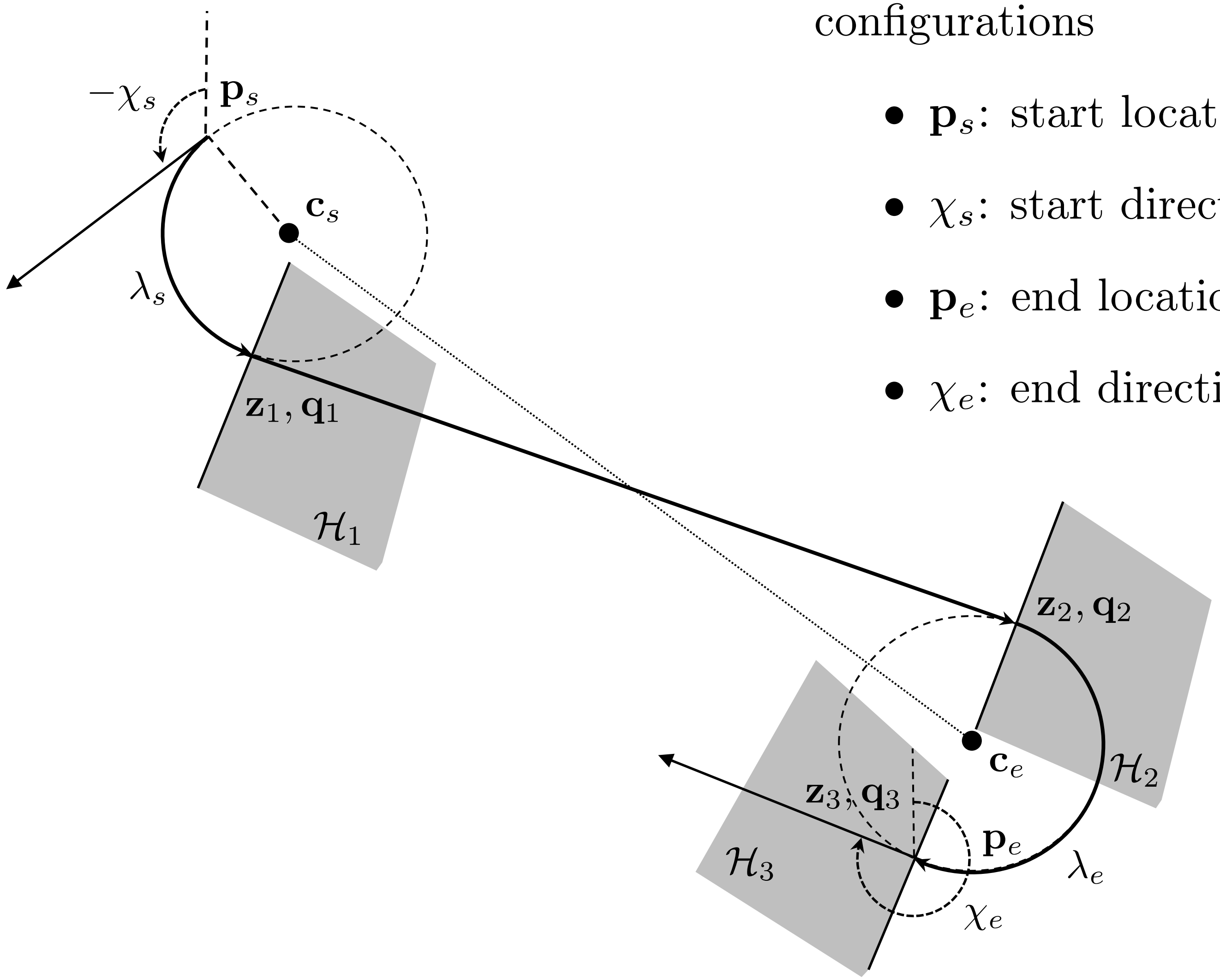
Distance traveled along $\mathbf{c}_{le}$

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

Total path length:

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

Dubins Path Half-plane Switching



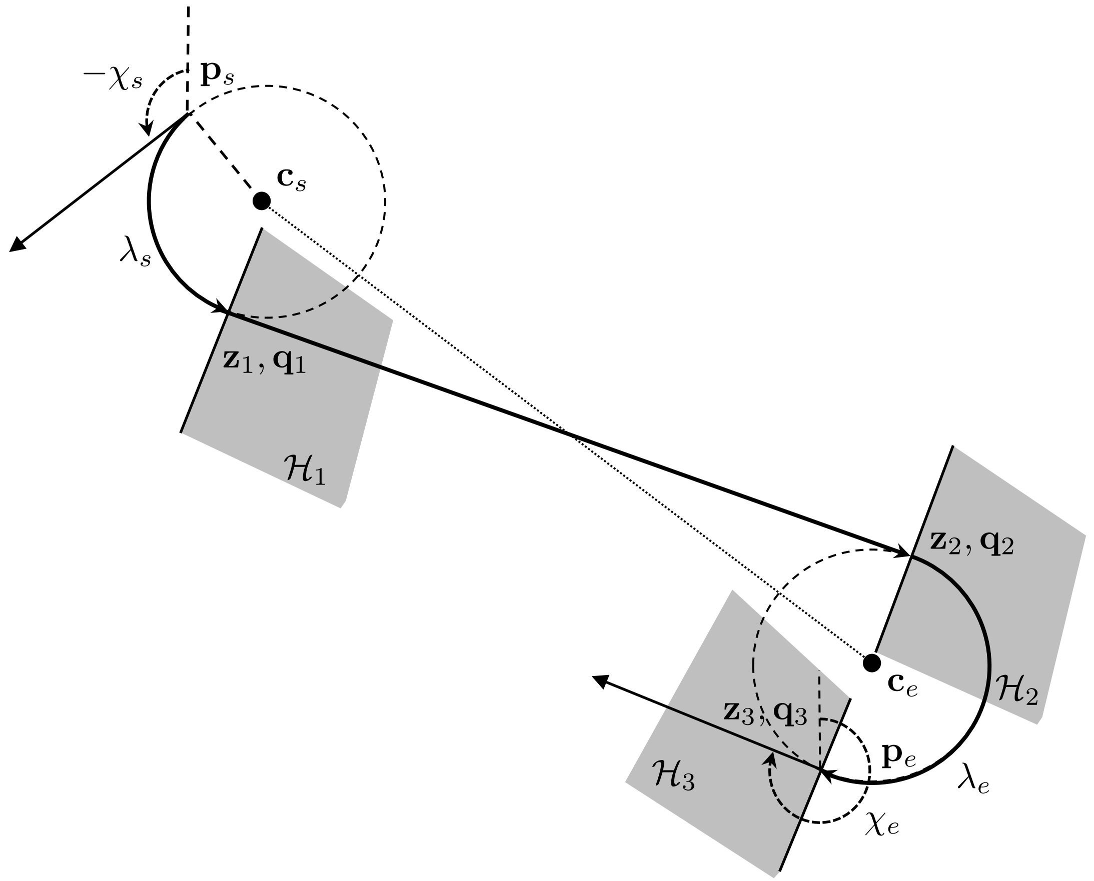
Given desired start and end configurations

- $\mathbf{p}_s$: start location
- χ_s: start direction
- $\mathbf{p}_e$: end location
- χ_e: end direction

Calculate Dubins path parameters

- $\mathbf{c}_s$: start circle location
- λ_s: start circle direction
- $\mathbf{c}_e$: end circle location
- λ_e: end circle direction
- $\mathbf{z}_1, \mathbf{q}_1$: half-plane $\mathcal{H}_1$ parameters
- $\mathbf{z}_2, \mathbf{q}_2$: half-plane $\mathcal{H}_2$ parameters
- $\mathbf{z}_3, \mathbf{q}_3$: half-plane $\mathcal{H}_3$ parameters

Dubins Path Parameters Algorithm

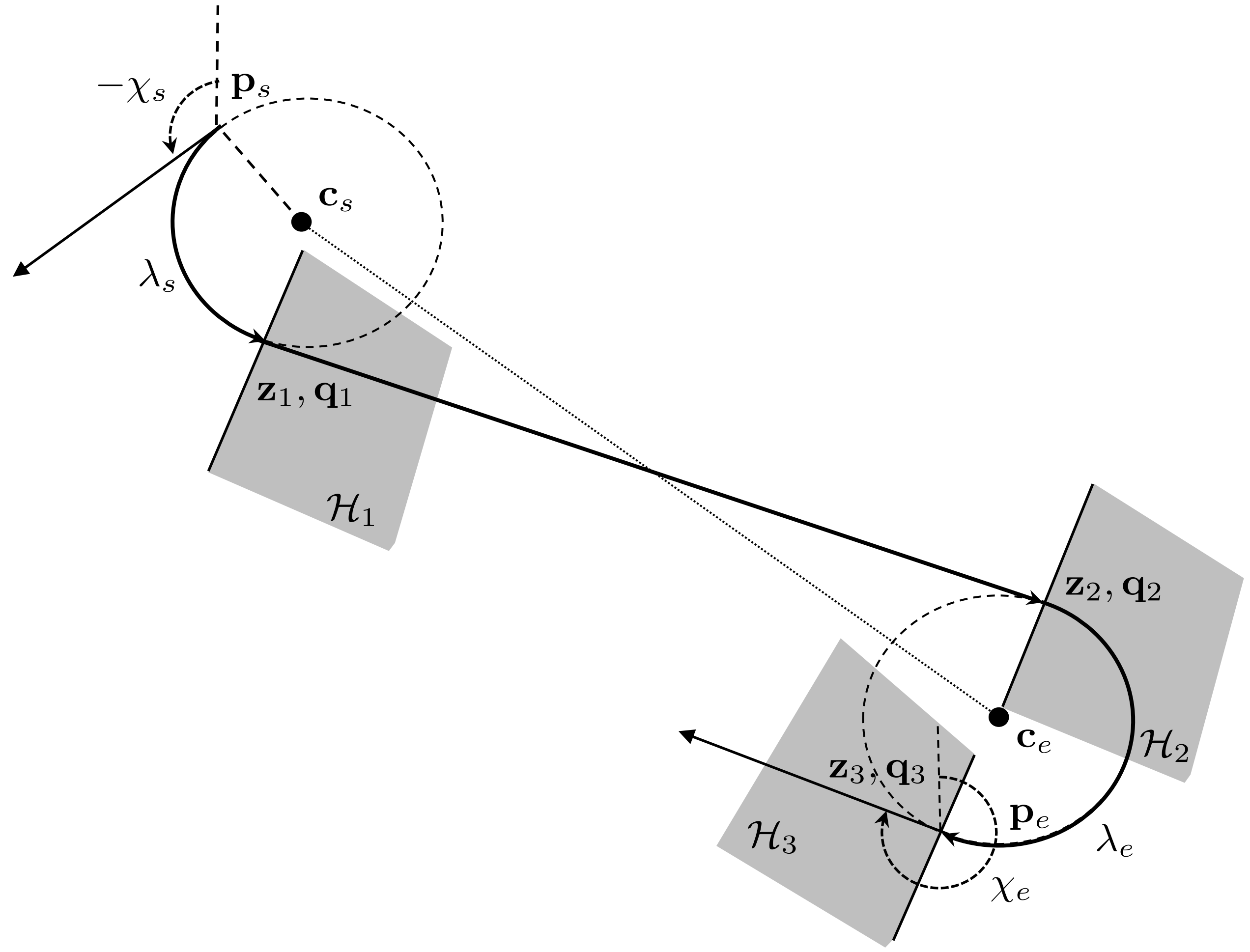


Algorithm 7 Find Dubins Parameters:
$(L, \mathbf{c}_s, \lambda_s, \mathbf{c}_e, \lambda_e, \mathbf{z}_1, \mathbf{q}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{q}_3) =$
findDubinsParameters($\mathbf{p}_s, \chi_s, \mathbf{p}_e, \chi_e, R$)

Input: Start configuration $(\mathbf{p}_s, \chi_s)$, End configuration $(\mathbf{p}_e, \chi_e)$, Radius R.
Require: $\|\mathbf{p}_s - \mathbf{p}_e\| \geq 3R$
Require: R is larger than minimum turn radius of MAV

$\mathbf{c}_{rs} \leftarrow \mathbf{p}_s + R\mathcal{R}_z\left(\frac{\pi}{2}\right)(\cos\chi_s, \sin\chi_s, 0)^\top$
$\mathbf{c}_{ls} \leftarrow \mathbf{p}_s + R\mathcal{R}_z\left(\frac{-\pi}{2}\right)(\cos\chi_s, \sin\chi_s, 0)^\top$
$\mathbf{c}_{re} \leftarrow \mathbf{p}_e + R\mathcal{R}_z\left(\frac{\pi}{2}\right)(\cos\chi_e, \sin\chi_e, 0)^\top$
$\mathbf{c}_{le} \leftarrow \mathbf{p}_e + R\mathcal{R}_z\left(-\frac{\pi}{2}\right)(\cos\chi_e, \sin\chi_e, 0)^\top$
Compute L_1, L_2, L_3, and L_3 using equations (11.9) through (11.12).
$L \leftarrow \min\{L_1, L_2, L_3, L_4\}$
if arg $\min\{L_1, L_2, L_3, L_4\} = 1$ **then**
  $\mathbf{c}_s \leftarrow \mathbf{c}_{rs}$, $\lambda_s \leftarrow +1$, $\mathbf{c}_e \leftarrow \mathbf{c}_{re}$, $\lambda_e \leftarrow +1$
  $\mathbf{q}_1 \leftarrow \frac{\mathbf{c}_e - \mathbf{c}_s}{\|\mathbf{c}_e - \mathbf{c}_s\|}$
  $\mathbf{z}_1 \leftarrow \mathbf{c}_s + R\mathcal{R}_z\left(-\frac{\pi}{2}\right)\mathbf{q}_1$
  $\mathbf{z}_2 \leftarrow \mathbf{c}_e + R\mathcal{R}_z\left(-\frac{\pi}{2}\right)\mathbf{q}_1$
else if arg $\min\{L_1, L_2, L_3, L_4\} = 2$ **then**
  $\mathbf{c}_s \leftarrow \mathbf{c}_{rs}$, $\lambda_s \leftarrow +1$, $\mathbf{c}_e \leftarrow \mathbf{c}_{le}$, $\lambda_e \leftarrow -1$
  $\ell \leftarrow \|\mathbf{c}_e - \mathbf{c}_s\|$
  $\vartheta \leftarrow \text{angle}(\mathbf{c}_e - \mathbf{c}_s)$
  $\vartheta_2 \leftarrow \vartheta - \frac{\pi}{2} + \sin^{-1}\frac{2R}{\ell}$
  $\mathbf{q}_1 \leftarrow \mathcal{R}_z\left(\vartheta_2 + \frac{\pi}{2}\right)\mathbf{e}_1$
  $\mathbf{z}_1 \leftarrow \mathbf{c}_s + R\mathcal{R}_z(\vartheta_2)\,\mathbf{e}_1$
  $\mathbf{z}_2 \leftarrow \mathbf{c}_e + R\mathcal{R}_z(\vartheta_2 + \pi)\,\mathbf{e}_1$
else if arg $\min\{L_1, L_2, L_3, L_4\} = 3$ **then**
  $\mathbf{c}_s \leftarrow \mathbf{c}_{ls}$, $\lambda_s \leftarrow -1$, $\mathbf{c}_e \leftarrow \mathbf{c}_{re}$, $\lambda_e \leftarrow +1$
  $\ell \leftarrow \|\mathbf{c}_e - \mathbf{c}_s\|$,
  $\vartheta \leftarrow \text{angle}(\mathbf{c}_e - \mathbf{c}_s)$,
  $\vartheta_2 \leftarrow \cos^{-1}\frac{2R}{\ell}$
  $\mathbf{q}_1 \leftarrow \mathcal{R}_z\left(\vartheta + \vartheta_2 - \frac{\pi}{2}\right)\mathbf{e}_1$,
  $\mathbf{z}_1 \leftarrow \mathbf{c}_s + R\mathcal{R}_z(\vartheta + \vartheta_2)\,\mathbf{e}_1$,
  $\mathbf{z}_2 \leftarrow \mathbf{c}_e + R\mathcal{R}_z(\vartheta + \vartheta_2 - \pi)\,\mathbf{e}_1$
else if arg $\min\{L_1, L_2, L_3, L_4\} = 4$ **then**
  $\mathbf{c}_s \leftarrow \mathbf{c}_{ls}$, $\lambda_s \leftarrow -1$, $\mathbf{c}_e \leftarrow \mathbf{c}_{le}$, $\lambda_e \leftarrow -1$
  $\mathbf{q}_1 \leftarrow \frac{\mathbf{c}_e - \mathbf{c}_s}{\|\mathbf{c}_e - \mathbf{c}_s\|}$,
  $\mathbf{z}_1 \leftarrow \mathbf{c}_s + R\mathcal{R}_z\left(\frac{\pi}{2}\right)\mathbf{q}_1$,
  $\mathbf{z}_2 \leftarrow \mathbf{c}_e + R\mathcal{R}_z\left(\frac{\pi}{2}\right)\mathbf{q}_2$
end if
$\mathbf{z}_3 \leftarrow \mathbf{p}_e$
$\mathbf{q}_3 \leftarrow \mathcal{R}_z(\chi_e)\mathbf{e}_1$

Dubins Path Following Algorithm

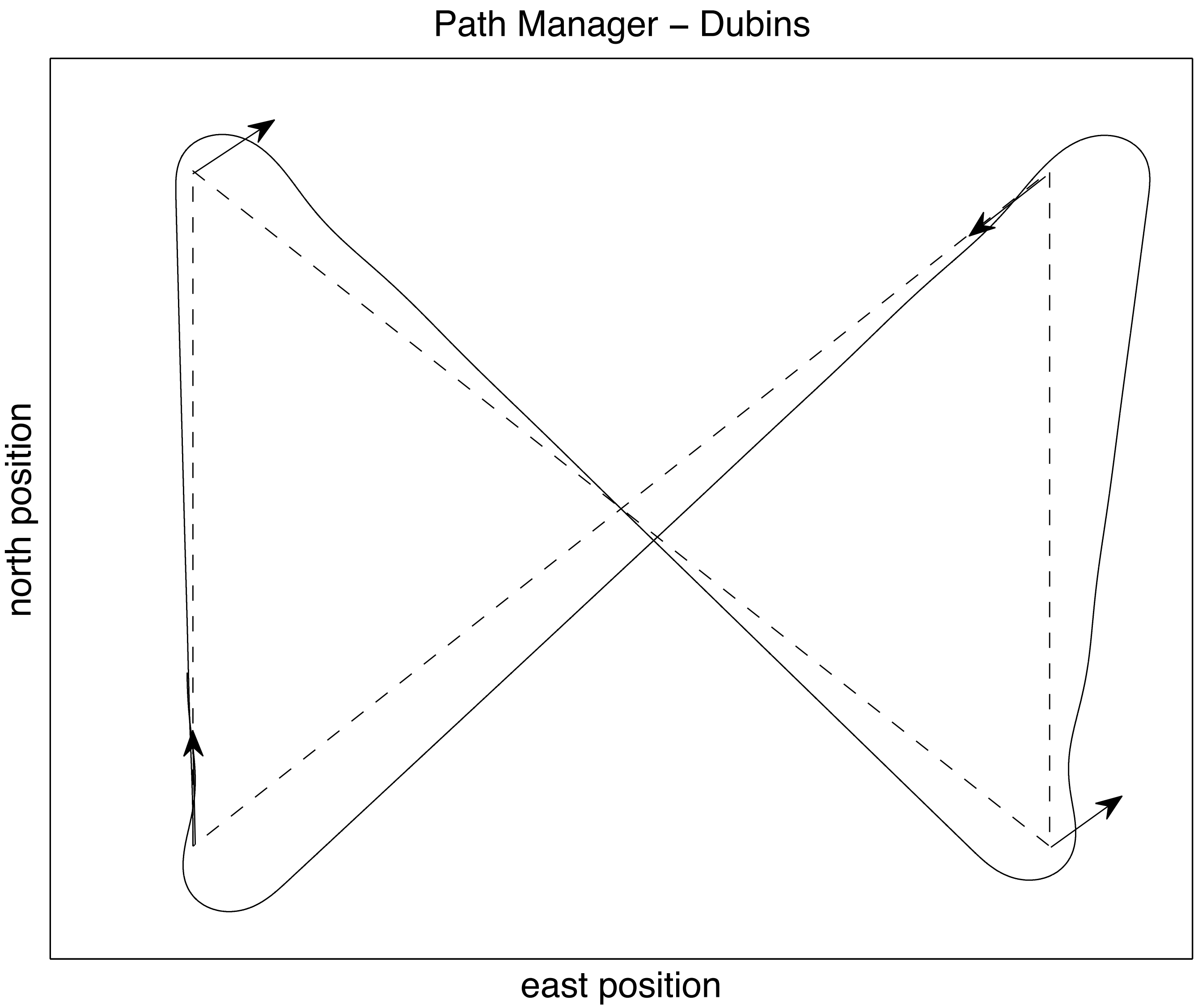


Algorithm 8 Follow Waypoints with Dubins: (flag, $\mathbf{r}$, $\mathbf{q}$, $\mathbf{c}$, ρ, λ) = followWppDubins($\mathcal{P}$, $\mathbf{p}$, R)

Input: Configuration path $\mathcal{P} = \{(\mathbf{w}_1, \chi_1), \ldots, (\mathbf{w}_N, \chi_N)\}$, MAV position $\mathbf{p} = (p_n, p_e, p_d)^\top$, fillet radius R.
Require: $N \geq 3$

```
if New configuration path P is received then
    Initialize waypoint pointer: i ← 2, and state machine: state ← 1.
end if
(L, c_s, λ_s, c_e, λ_e, z_1, q_1, z_2, z_3, q_3) ←
findDubinsParameters(w_{i−1}, χ_{i−1}, w_i, χ_i, R)
if state = 1 then
    flag ← 2, c ← c_s, ρ ← R, λ ← λ_s
    if p ∈ H(z_1, −q_1) then
        state ← 2
    end if
else if state = 2 then
    if p ∈ H(z_1, q_1) then
        state ← 3
    end if
else if state = 3 then
    flag ← 1, r ← z_1, q ← q_1
    if p ∈ H(z_2, q_1) then
        state ← 4
    end if
else if state = 4 then
    flag ← 2, c ← c_e, ρ ← R, λ ← λ_e
    if p ∈ H(z_3, −q_3) then
        state ← 5
    end if
else if state = 5 then
    if p ∈ H(z_3, q_3) then
        state ← 1
        i ← (i + 1) until i = N.
        (L, c_s, λ_s, c_e, λ_e, z_1, q_1, z_2, z_3, q_3) ←
            findDubinsParameters(w_{i−1}, χ_{i−1}, w_i, χ_i, R)
    end if
end if
return flag, r, q, c, ρ, λ.
```

Dubins Path Following Results



Need to incorporate in this next set of slides into book, and also slide deck.

C-C-C DUBINS PATHS

- Dubins RRT in dense obstacle fields works better with short segment lengths
- While methods exist to determine optimality without calculating paths [6], an exhaustive search was implemented as in the textbook with help from [7]
- An additional state is needed in the path manager (state 3.5)

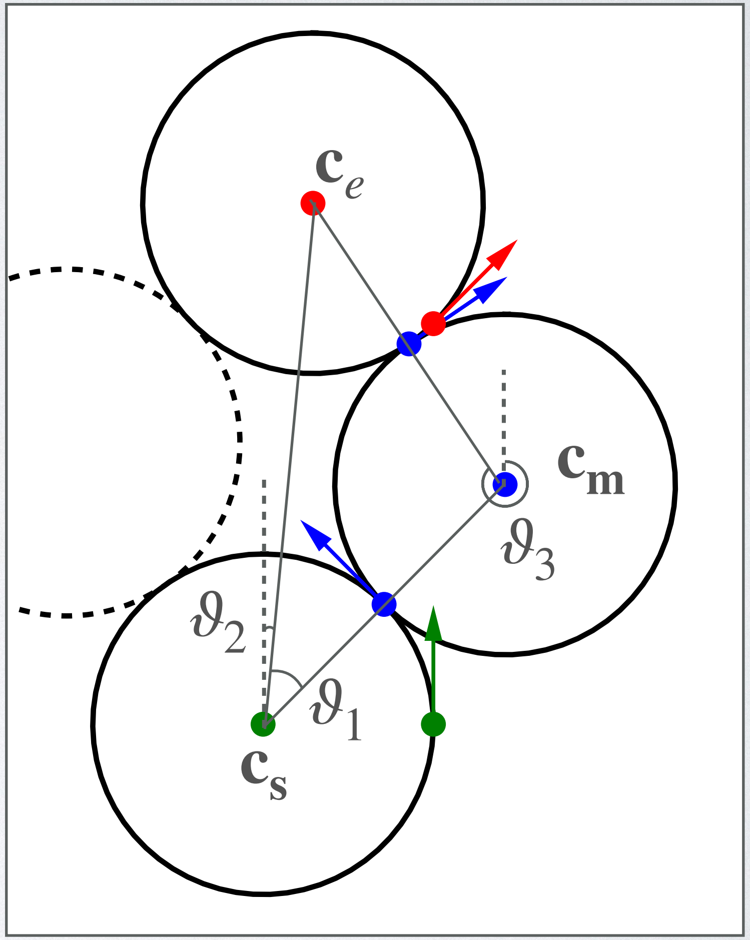
Case	Conditions for Validity
Case I: R-S-R	Always
Case II: R-S-L	$\|\mathbf{p}_e - \mathbf{p}_s\| \geq 4R$ and $\|\mathbf{c}_{le} - \mathbf{c}_{rs}\| \geq 4R$
Case III: L-S-R	$\|\mathbf{p}_e - \mathbf{p}_s\| \geq 4R$ and $\|\mathbf{c}_{re} - \mathbf{c}_{ls}\| \geq 4R$
Case IV: L-S-L	Always
Case V: R-L-R	$\|\mathbf{p}_e - \mathbf{p}_s\| \leq 4R$ and $\|\mathbf{c}_{re} - \mathbf{c}_{rs}\| \geq 4R$
Case VI: L-R-L	$\|\mathbf{p}_e - \mathbf{p}_s\| \leq 4R$ and $\|\mathbf{c}_{le} - \mathbf{c}_{ls}\| \geq 4R$

C-C-C DUBINS PATHS

- The angle between $\mathbf{c}_e - \mathbf{c}_s$ and the middle circle center $\mathbf{c}_m$ is given by

$$\vartheta_1 = \left\langle \pm\cos^{-1}\left(\frac{\ell}{4R}\right)\right\rangle$$

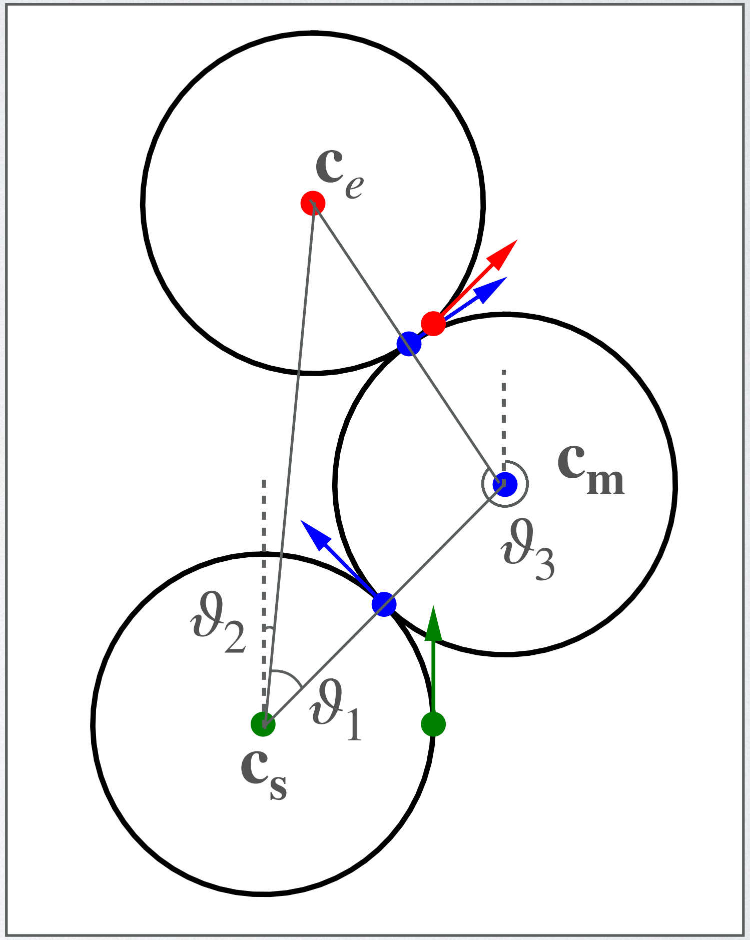
- The $\pm$ in the equation for ϑ_1 indicates that there are two cases for both R-L-R and L-R-L: one where the circle is to the right of $\mathbf{c}_e - \mathbf{c}_s$ (+) and one where it is to the left (−)



C-C-C DUBINS PATHS

- The center of the middle circle is given by $\mathbf{c}_m = \mathbf{c}_{rs} + 2R\mathscr{R}_z(\vartheta_1)\dfrac{\mathbf{c}_{re} - \mathbf{c}_{rs}}{\ell}$, where $\ell = \|c_{re} - c_{rs}\|$ for R-L-R and $\ell = \|c_{le} - c_{ls}\|$ for L-R-L

- ϑ_2 and ϑ_1 are defined as the angles of the vectors $\mathbf{c}_m - \mathbf{c}_s$ and $\mathbf{c}_e - \mathbf{c}_m$ respectively



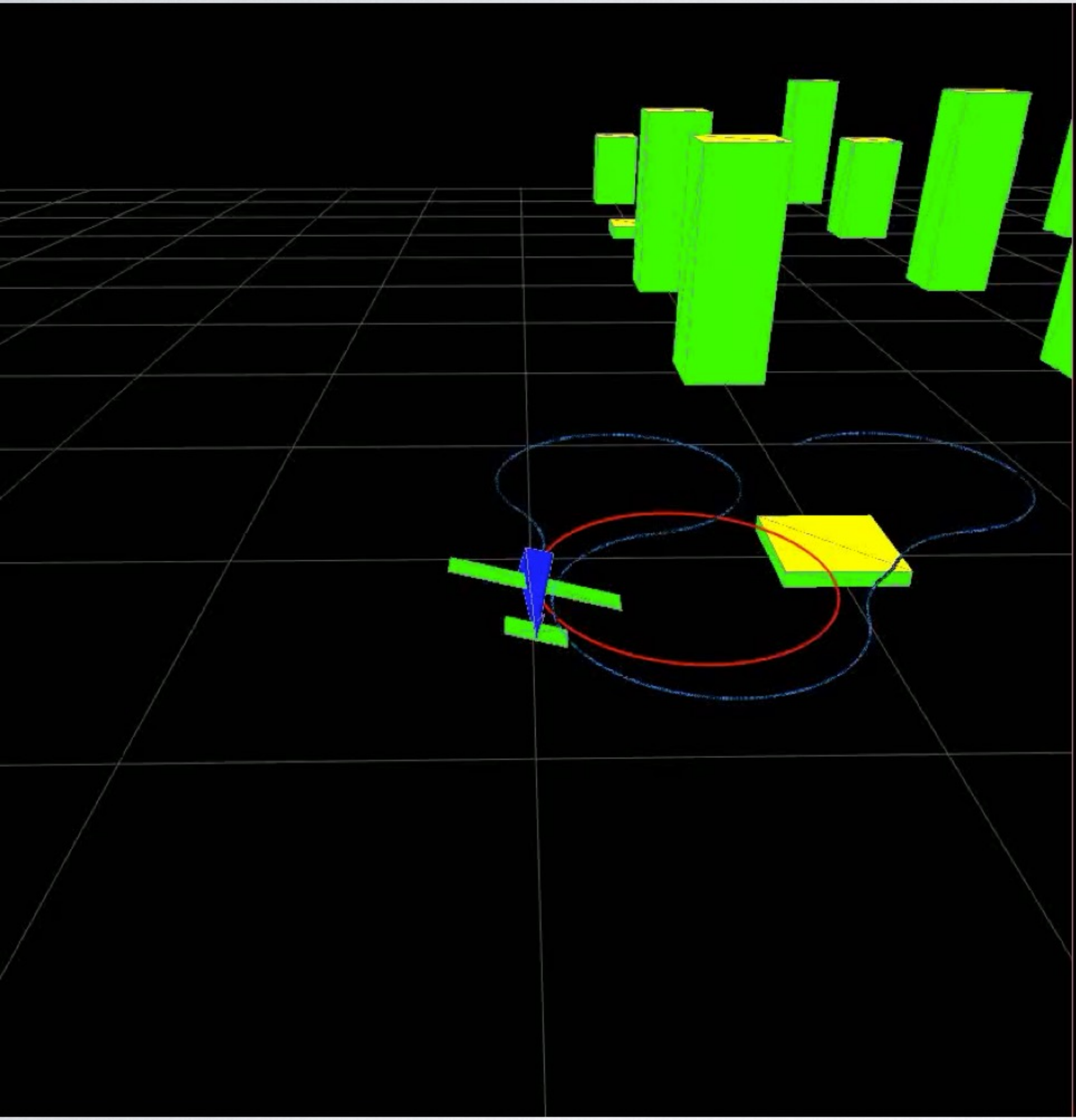
C-C-C DUBINS PATHS

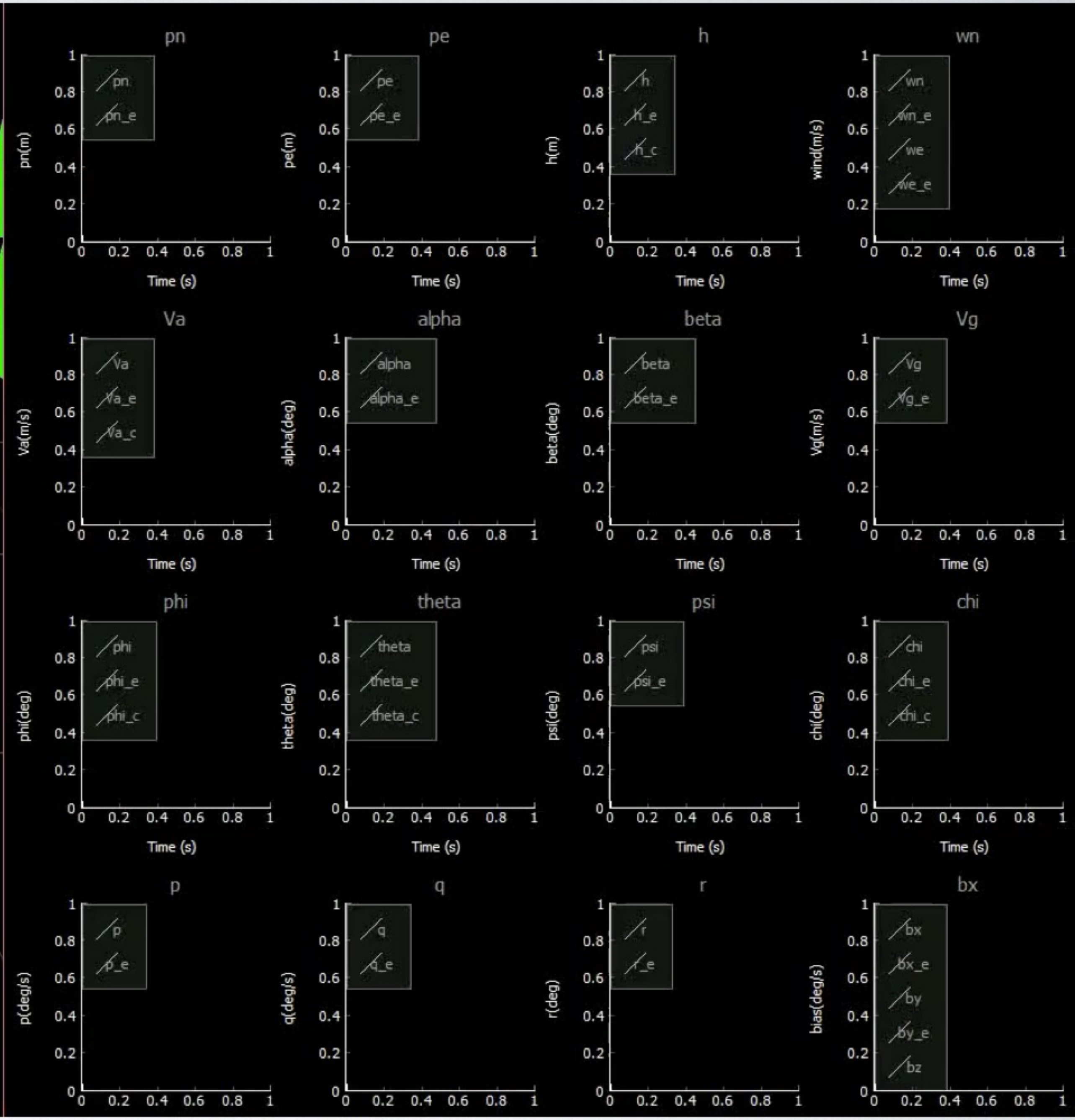
- R-L-R Length:

$$L_5 = R\left\langle 2\pi + \left\langle \frac{\pi}{2} - \vartheta_2 \right\rangle - \left\langle \chi_s - \frac{\pi}{2} \right\rangle \right\rangle$$
$$+R\left\langle 2\pi - \left\langle \frac{\pi}{2} - \vartheta_3 \right\rangle + \left\langle \frac{3\pi}{2} - \vartheta_2 \right\rangle \right\rangle$$
$$+R\left\langle 2\pi + \left\langle \chi_e - \frac{\pi}{2} \right\rangle - \left\langle \frac{3\pi}{2} - \vartheta_3 \right\rangle \right\rangle$$

- L-R-L Length:

$$L_6 = R\left\langle 2\pi - \left\langle \frac{\pi}{2} - \vartheta_2 \right\rangle + \left\langle \chi_s - \frac{\pi}{2} \right\rangle \right\rangle$$
$$+R\left\langle 2\pi + \left\langle \frac{\pi}{2} - \vartheta_3 \right\rangle - \left\langle \frac{3\pi}{2} - \vartheta_2 \right\rangle \right\rangle$$
$$+R\left\langle 2\pi - \left\langle \chi_e - \frac{\pi}{2} \right\rangle + \left\langle \frac{3\pi}{2} - \vartheta_3 \right\rangle \right\rangle$$

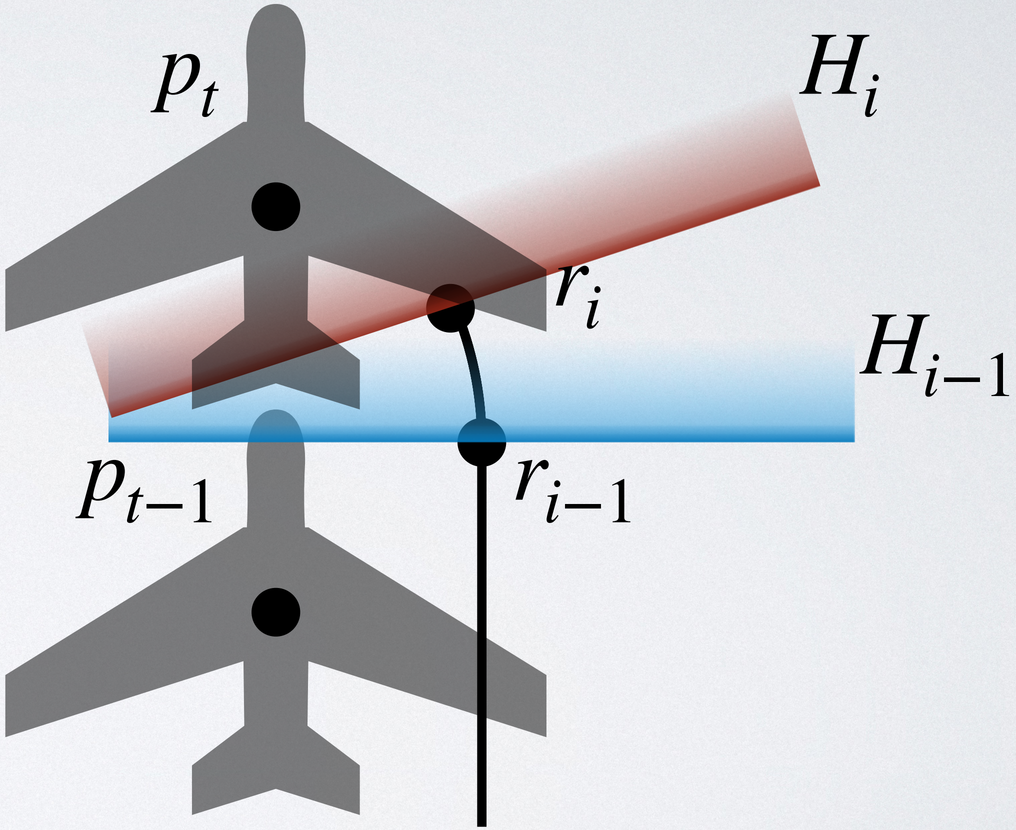




pn
pn(m)
pn
pn_e
Time (s)
pe
pe(m)
pe
pe_e
Time (s)
h
h(m)
h
h_e
h_c
Time (s)
wn
wind(m/s)
wn
wn_e
we
we_e
Time (s)
Va
Va(m/s)
Va
Va_e
Va_c
Time (s)
alpha
alpha(deg)
alpha
alpha_e
Time (s)
beta
beta(deg)
beta
beta_e
Time (s)
Vg
Vg(m/s)
Vg
Vg_e
Time (s)
phi
phi(deg)
phi
phi_e
phi_c
Time (s)
theta
theta(deg)
theta
theta_e
theta_c
Time (s)
psi
psi(deg)
psi
psi_e
Time (s)
chi
chi(deg)
chi
chi_e
chi_c
Time (s)
p
p(deg/s)
p
p_e
Time (s)
q
q(deg/s)
q
q_e
Time (s)
r
r(deg)
r
r_e
Time (s)
bx
bias(deg/s)
bx
bx_e
by
by_e
bz
Time (s)

ROBUST PATH MANAGEMENT

- The dual half-plane path manager occasionally flew a full orbit when paths included short arcs
- The negative half plane check should be skipped when points are close
  - $r_{i-1} \in -H_i$
  - $s < R\pi$
  - $\theta < \pi$



The best way to do this is check distance along circles and straight lines. If it is small, then just transition to the next segment.

REFERENCES

[1] T. Wallace, D. Watkins, and J. Schwartz, "A Map of Every Building in America," *The New York Times*, Oct 2018. Data available: https://github.com/microsoft/USBuildingFootprints.

[2] J. O'Rourke *et al.*, *Computational Geometry in C.* Cambridge University Press, 2 ed., 1998.

[3] S. Gillies *et al.*, "Shapely: Manipulation and Analysis of Geometric Objects," 2007.

[4] P. Virtanen *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[5] A. Hagberg, P. Swart, and D. S Chult, "Exploring Network Structure, Dynamics, and Function using NetworkX," Tech. Rep. LA-UR-08-05495, Los Alamos National Laboratory, 1 2008.

[6] A. M. Shkel and V. Lumelsky, "Classification of the Dubins set," *Robotics and Autonomous Systems*, vol. 34, no. 4, pp. 179–202, 2001.

[7] A. Giese, "A Comprehensive, Step-by-Step Tutorial on Computing Dubins Curves," 2014.

Dubins Airplane Model

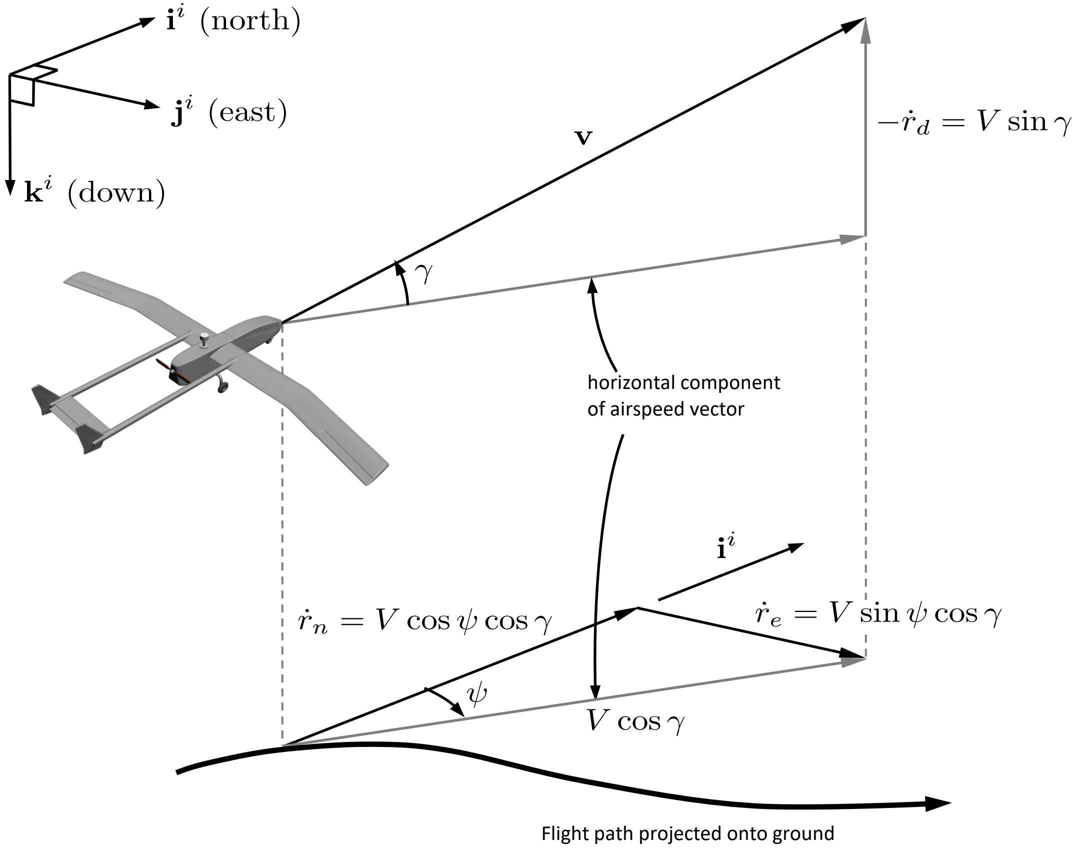
Adapted from: Mark Owen, Randal W. Beard, Timothy W. McLain, "Implementing Dubins Airplane Paths on Fixed-wing UAVs," *Handbook of Unmanned Aerial Vehicles,* ed. Kimon P. Valavanis, George J. Vachtsevanos, Springer Verlag, Section XII, Chapter 68, p. 1677-1702, 2014.

Dubins Airplane model:

$$\dot{r}_n = V\cos\psi\cos\gamma^c$$
$$\dot{r}_e = V\sin\psi\cos\gamma^c$$
$$\dot{r}_d = -V\sin\gamma^c$$
$$\dot{\psi} = \frac{g}{V}\tan\phi^c$$

Where the commanded flight path angle γ^c and the commanded roll angle ϕ^c are constrained by

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



3D Vector Field Path Following

Adapted from: V. M. Goncalves, L. C. A. Pimenta, C. A. Maia, B. C. O. Durtra, G. A. S. Pereira, B. C. O. Dutra, and G. A. S. Pereira, "Vector Fields for Robot Navigation Along Time-Varying Curves in n-Dimensions," IEEE Transactions on Robotics, vol. 26, pp. 647–659, Aug 2010.

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

$$\alpha_1(\mathbf{r}) = 0$$
$$\alpha_2(\mathbf{r}) = 0$$

$\mathbf{r} \in \mathbb{R}^3$. Define the composite function

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

Note that the gradient

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

points away from the path.

3D Vector Field Path Following

The desired velocity vector can be chosen as

$$\mathbf{u}' = \underbrace{-K_1 \frac{\partial W}{\partial \mathbf{r}}}_{\text{velocity directed toward the path}} + \underbrace{K_2 \frac{\partial \alpha_1}{\partial \mathbf{r}} \times \frac{\partial \alpha_2}{\partial \mathbf{r}}}_{\text{velocity directed along the path}}$$

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 $\mathbf{u}'$ may not equal V_a, normalize to get

$$\mathbf{u} = V_a \frac{\mathbf{u}'}{\|\mathbf{u}'\|}.$$

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

$$\begin{aligned} V\cos\psi^d\cos\gamma^c &= u_1 \\ V\sin\psi^d\cos\gamma^c &= u_2 \\ -V\sin\gamma^c &= u_3. \end{aligned}$$

Solving for γ^c, and ψ^d results in

$$\begin{aligned} \gamma^c &= -\mathrm{sat}_{\bar{\gamma}}\left[\sin^{-1}\left(\frac{u_3}{V}\right)\right] \\ \psi^d &= \mathrm{atan2}(u_2, u_1). \end{aligned}$$

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

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

where k_ϕ is a positive constant.

3D Vector Field – Straight Line path

The straight line path is given by

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

where

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

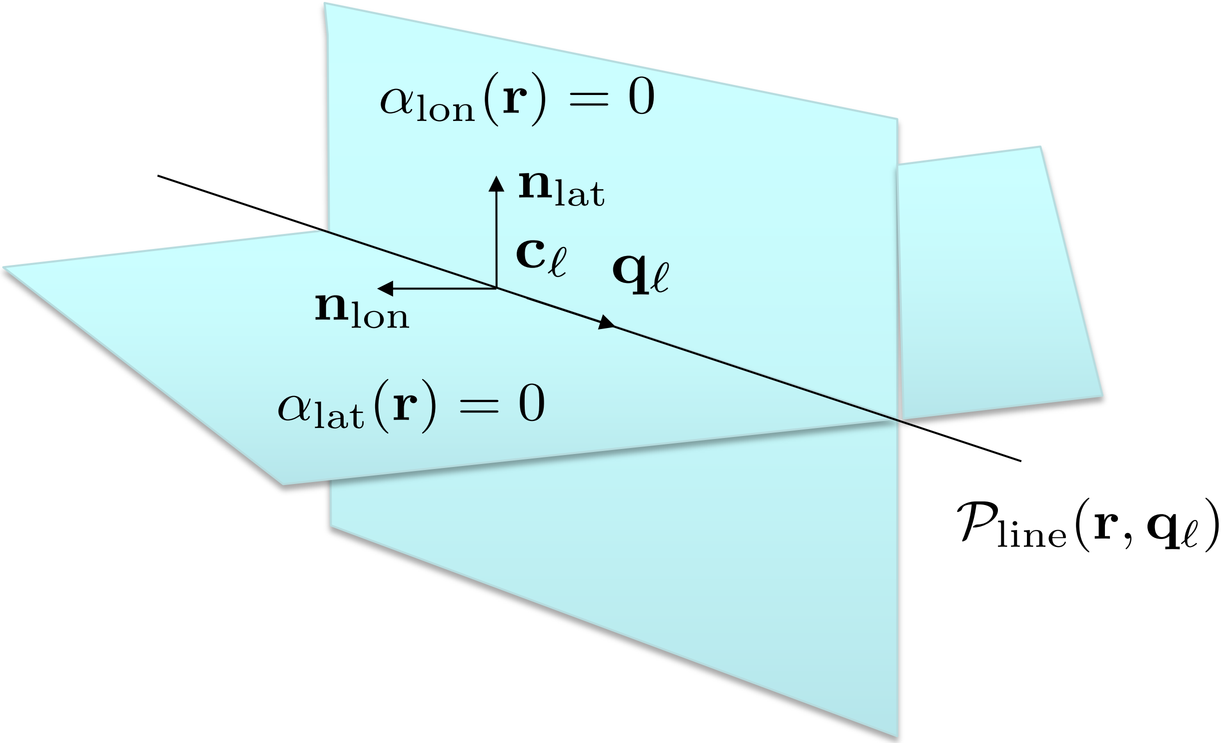
Define

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

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

to get

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



3D Vector Field – Helical Path

A helical path is then defined as

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

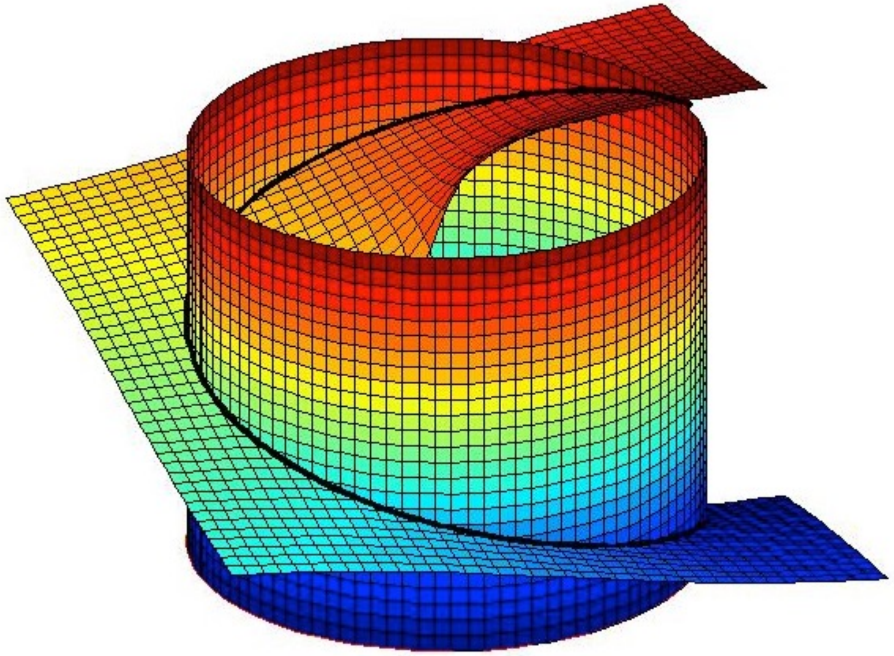
where

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

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

where the initial position along the helix is

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



$\mathbf{c}_h$ is the center of the helix, R_h is the radius, γ_h is the climb angle.

Dubins Airplane Paths

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

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

Low Altitude:

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

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:

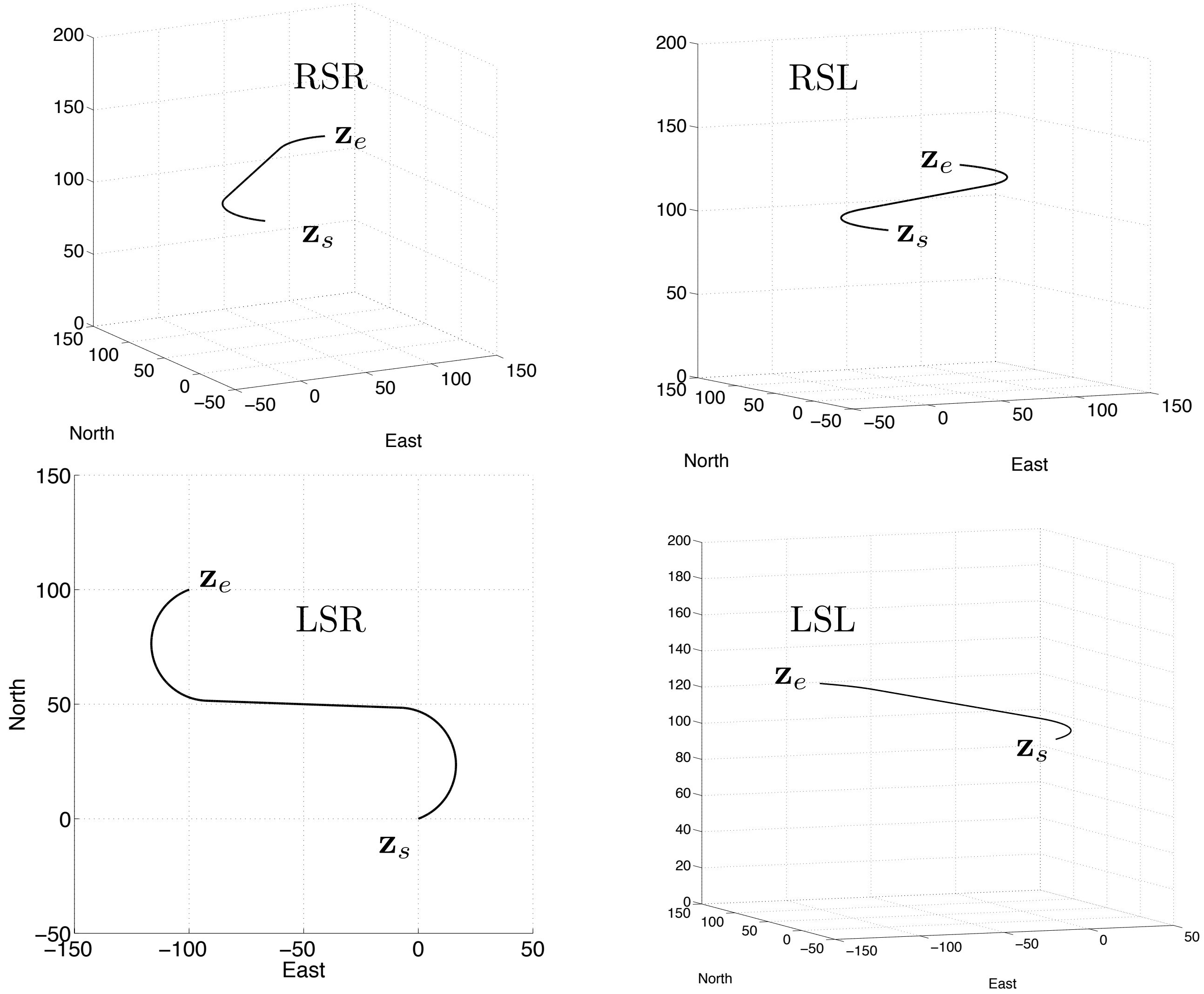
$$|z_{de} - z_{ds}| > [L_{\text{car}}(R_{\min}) + 2\pi R_{\min}] \tan \bar{\gamma}.$$

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

Medium Altitude:

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

Low Altitude Dubins Airplane Paths

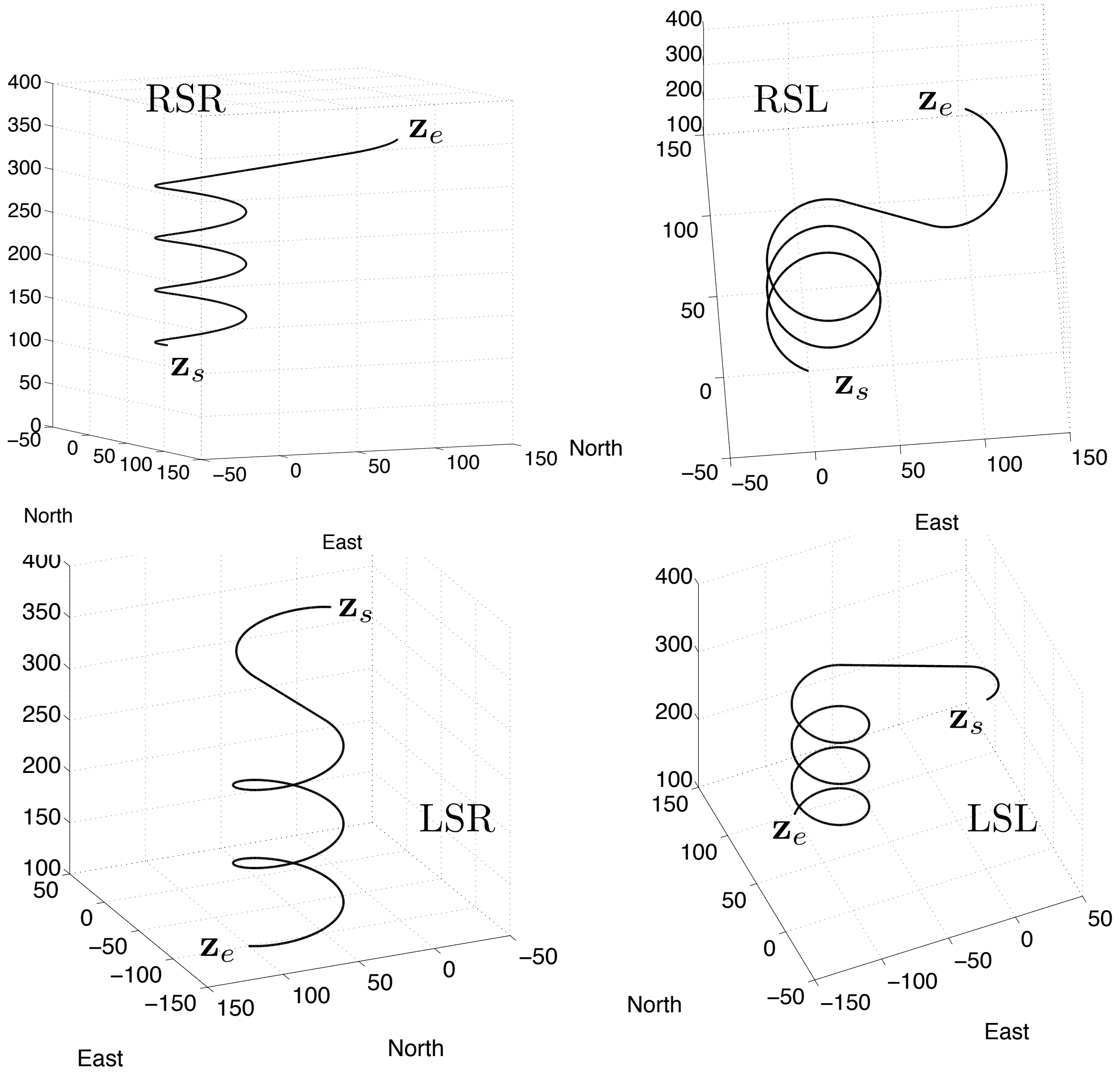


$$\gamma^* = \tan^{-1}\left(\frac{|z_{de} - z_{ds}|}{L_{\text{car}}(R_{\min})}\right)$$

$$R^* = R_{\min}$$

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

High Altitude Dubins Airplane Paths



Find smallest integer k such that

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

Increase the radius R^* so that

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

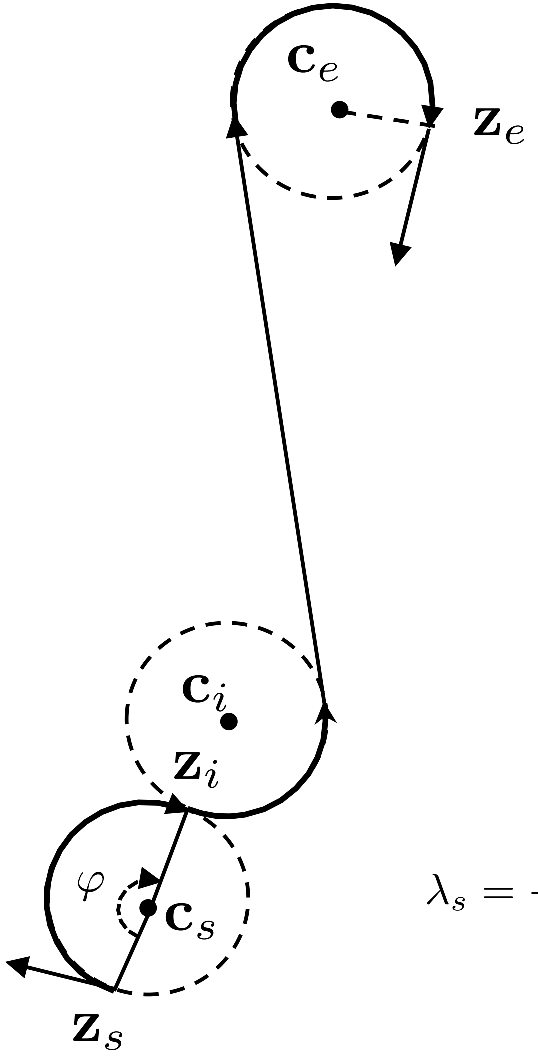
The resulting path is

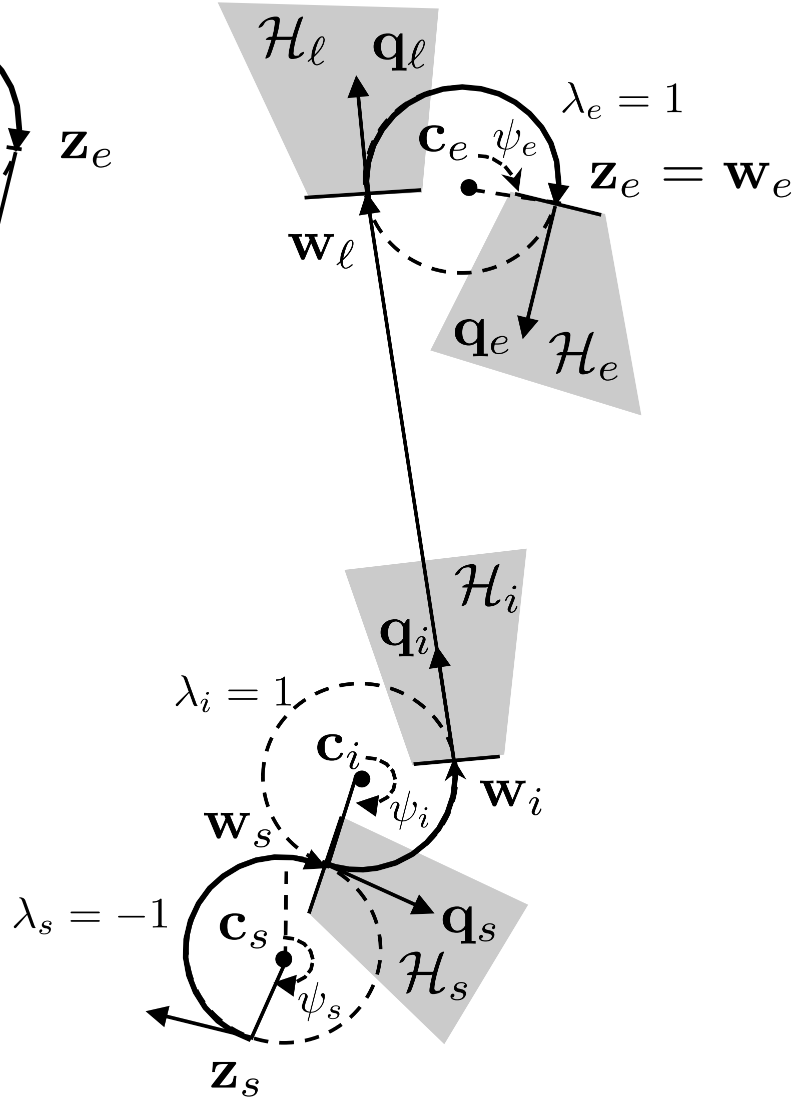
$$L_{\text{air}}(R^*, \bar{\gamma}) = \frac{L_{\text{car}}(R^*)}{\cos\bar{\gamma}}.$$

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