# Inverse Optimal Planning for Air Traffic Control

Ekaterina Tolstaya[1†], Alejandro Ribeiro[1], Vijay Kumar[1], Ashish Kapoor[2]

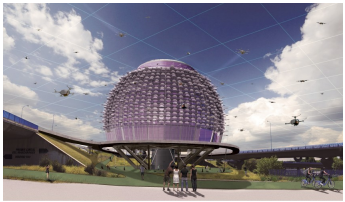
[1] University of Pennsylvania
[2] Microsoft Corporation
[†] eig@seas.upenn.edu

International Conference on Intelligent Robots and Systems
November 7, 2019

- Dense airspaces with:
  - $\Rightarrow$ Human-controlled and autonomous aerial vehicles that co-exist safely
  - $\Rightarrow$ Infrastructure for recreational, commercial and emergency aerial vehicles
- Autonomous air traffic control that imitates human air traffic control


(a) "Reckless drone flies ABOVE passenger jet" YouTube


(b) US Forest Service



(c) Architect Magazine

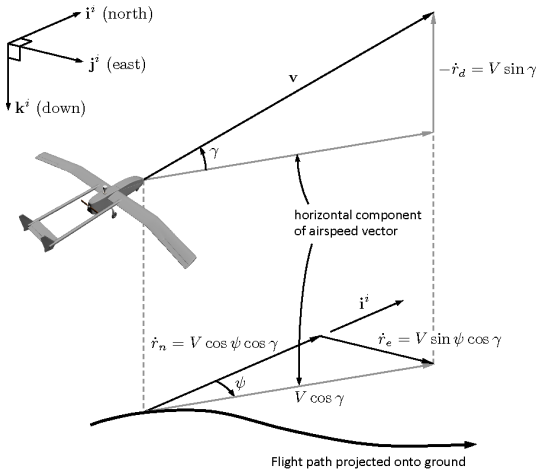
# Outline

- Dubins airplane model, $\mathbf{s} = (x, y, z, \phi)$

$$\dot{\mathbf{s}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} v\cos\phi \\ v\sin\phi \\ u_z \\ u_\phi \end{bmatrix}$$

- Goal: plan a feasible trajectory
  - $\Rightarrow$ from an initial state $\mathbf{s}_0 = (x_0, y_0, z_0, \phi_0)$
  - $\Rightarrow$ to a goal state $\mathbf{s}_g = (x_g, y_g, z_g, \phi_g)$.
  - $\Rightarrow$ that minimizes path length

$$\underset{\mathbf{s}(t)}{\operatorname{argmin}} \int_{t_0}^{T} \|\dot{\mathbf{s}}(t)\|\, dt$$




[OBM15]

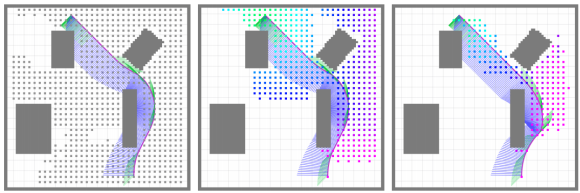
- Search-based planning (versus sampling-based planning)
  - $\Rightarrow$ Result is resolution-optimal (discretize space and time)
  - $\Rightarrow$ Ability to deal with dynamic obstacles
  - $\Rightarrow$ Ability to deal with non-holonomic systems and higher-order dynamics [LAMK17]
  - $\Rightarrow$ Finite-time sub-optimality using the ARA* search [LGT04]
- States are discretized with a resolution of $\boldsymbol{\rho} = [\rho_x, \rho_y, \rho_z, \rho_\phi]$:

$$\bar{\mathbf{s}} = \left[ \left\lfloor \frac{\mathbf{s}_x}{\boldsymbol{\rho}_x} \right\rfloor, \left\lfloor \frac{\mathbf{s}_y}{\boldsymbol{\rho}_y} \right\rfloor, \left\lfloor \frac{\mathbf{s}_z}{\boldsymbol{\rho}_z} \right\rfloor, \left\lfloor \frac{\mathbf{s}_\phi}{\boldsymbol{\rho}_\phi} \right\rfloor \right]$$

- Goal: find a path from $\mathbf{s}_0$ to $\mathbf{s}_g$ in graph $\mathcal{G}$
- First try: Breadth-first search over all nodes in $\mathcal{G}$ (Djikstra's algorithm)
- Better: Prioritize nodes using a heuristic (A*)
- Best: Anytime heuristic search (ARA*) [LGT04]

  $\Rightarrow$ Over-inflates the heuristic to provide a sub-optimal solution in finite time

  $\Rightarrow$ Continually improves trajectory and reduces the sub-optimality bound



(a) Dijkstra. $T_p = 0.16s, N_p = 2707$ (b) $A^*$ with $h_1$. $T_p = 0.064s, N_p = 1282$ (c) $A^*$ with $h_2$. $T_p = 0.016s, N_p = 376$

Figure. A* and ARA* require an admissible heuristic! [LAMK17]

**Algorithm 1** Dubins Airplane Heuristic

**Input:** Start $\mathbf{s}_0$, goal $\mathbf{s}_g$, forward speed $v$, max rate of climb $\Delta z$, turning rate $\Delta\phi$

**Output:** Minimum path length from start to goal, $d_{min}$

1: Compute Dubins car distance $d_{xy}$ using the LSL, RSR, LSR, RSL, RLR, LRL paths with curvature $\kappa = \Delta\phi / v$
2: Compute the minimum time from start to goal in the $xy$ plane: $t_{min} = d_{xy}/v$
3: Compute the minimum time for ascent/descent:
4: $t_z = |z_g - z_0|/\Delta z$
5: **while** $t_z > t_{min}$ **do**
6: Add a helical ascent/descent to the trajectory:
7: $t_{min} = t_{min} + \frac{2\pi}{\Delta\phi}$
8: **end while**
9: Compute the minimum path length from start to goal: $d_{min} = \sqrt{(v \cdot t_{min})^2 + (z_g - z_0)^2}$

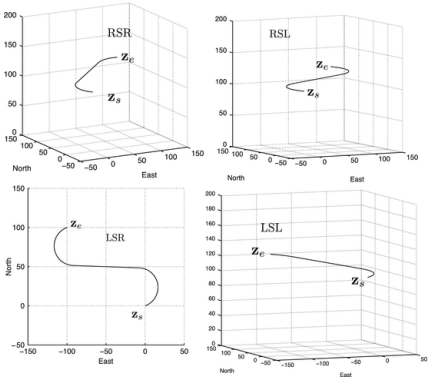



Figure. [OBM15]

- Safety requirements result in a constrained optimization problem:

$$\begin{aligned} &\underset{\mathbf{s}(t)}{\operatorname{argmin}} \int_{t_0}^{T} \|\dot{\mathbf{s}}(t)\|\, dt \\ \text{s.t. } &\dot{\mathbf{s}}(t) = \big(\cos\phi, \sin\phi, u_z(t), u_\phi(t)\big) \\ &\mathbf{s}(t_0) = \mathbf{s}_0, \mathbf{s}(T) = \mathbf{s}_g \\ &\mathbf{s}(t) \in \mathcal{S}^{safe}, \mathbf{u}(t) \in \mathcal{U} \end{aligned} \tag{1}$$

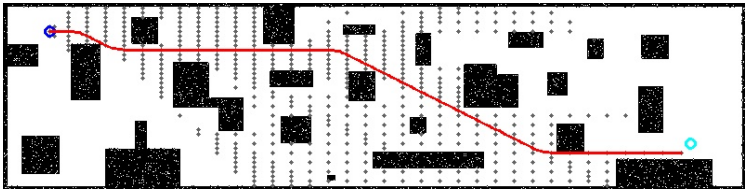


Figure. [LMAK18]

- Potential function $J$ to encode requirements and preferences of air traffic control
- The optimization problem is formulated as:

$$\begin{aligned} &\underset{\mathbf{s}(t)}{\operatorname{argmin}} \int_{t_0}^{T} \Big(1 + J\big(\mathbf{s}(t)\big)\Big) \, \|\dot{\mathbf{s}}(t)\| \, dt \\ \text{s.t. } &\dot{\mathbf{s}}(t) = \big(\cos\phi, \sin\phi, u_z(t), u_\phi(t)\big) \\ &\mathbf{s}(t_0) = \mathbf{s}_0, \mathbf{s}(T) = \mathbf{s}_g \\ &\mathbf{s}(t) \in \mathcal{S}^{safe}, \mathbf{u}(t) \in \mathcal{U} \end{aligned} \tag{2}$$

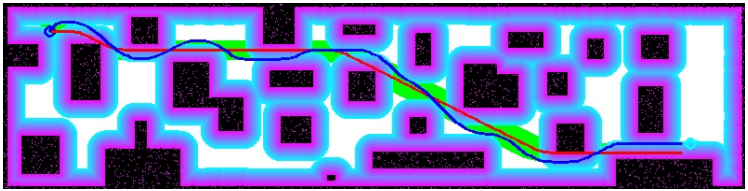


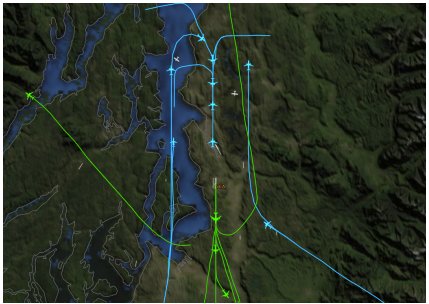
Figure. [LMAK18]

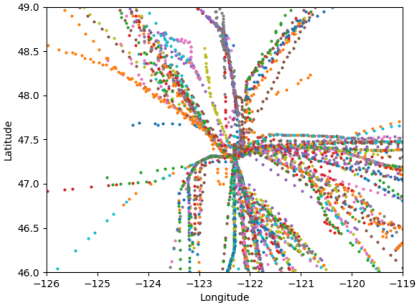
- Seattle-Tacoma airport data provided by FlightAware [Fli19]

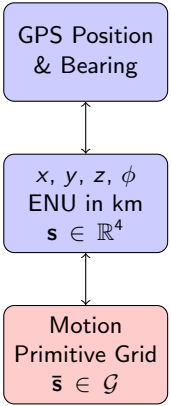
  $\Rightarrow$ Local geography, air traffic density requirements

- Three parametrizations are used for expressing airplane states:

  $\Rightarrow$ Continuous state in blue, discrete - in red





- We are given a data set of expert demonstrations, $\mathcal{D} = \left\{ s_i^e(t) \right\}_{i=1,\dots,M}$
- Assumption: The expert demonstrations follow the principal of maximum entropy: [Jay57]:

  $\Rightarrow$ An expert is exponentially more likely to select a trajectory $\tau$ with a lower cost:

$$\mathbb{P}(\tau) \propto e^{-C(\tau)} \tag{3}$$

- We are given a data set of expert demonstrations, $\mathcal{D} = \left\{ s_i^e(t) \right\}_{i=1,\dots,M}$
- Assumption: The expert demonstrations follow the principal of maximum entropy: [Jay57]:

  $\Rightarrow$ An expert is exponentially more likely to select a trajectory $\tau$ with a lower cost:

$$\mathbb{P}(\tau) \propto e^{-C(\tau)} \tag{3}$$

- Hypothesis about the parametrization of the potential function

$$J_\theta(\mathbf{s}) = \theta^T \mathbf{f}(\mathbf{s}). \tag{4}$$

- Now, we seek to maximize the log likelihood of the expert trajectories:

$$\mathcal{L}(\theta) = \log \mathbb{P}(\mathcal{D}, \theta | J_\theta) \tag{5}$$

- Following the approach of [ZMBD08, WOP15]:

  $\Rightarrow$ Expectation of the learner's state visitation counts:

$$\mathbb{E}[\mathbf{f}_\theta] = \sum_{\mathbf{s}\in\mathcal{S}} \mathbf{f}(\mathbf{s})\mathbb{P}(\mathbf{s}|J_\theta) \tag{6}$$

  $\Rightarrow$ Expert's empirical feature counts:

$$\mathbf{f}_\mathcal{D} = \sum_{\tau\in\mathcal{D}} \sum_{\mathbf{s}\in\tau} \mathbb{P}(\tau)\mathbf{f}(\mathbf{s}) \tag{7}$$

- $\nabla_\theta \mathcal{L}$ is the difference in feature counts of the trajectories of the learner and the expert

$$\nabla_\theta \mathcal{L} = \mathbb{E}[\mathbf{f}_\theta] - \mathbf{f}_\mathcal{D} \tag{8}$$

- Instead, we use the stochastic gradient computed using trajectories from the learner: [KPRS13]:

$$\hat{\nabla}_\theta \mathcal{L} = \underbrace{\frac{1}{N}\sum_{i=1}^{N}\sum_{t=t_0}^{T_i} f\left(\mathbf{s}_i(t)\right)}_{\text{learner}} - \underbrace{\frac{1}{M}\sum_{i=1}^{M}\sum_{t=t_0}^{T_i} f\left(\mathbf{s}_i^e(t)\right)}_{\text{expert}} \tag{9}$$

**Algorithm 2** Inverse Optimal Control for Air Traffic

**loop** until convergence
  Receive expert trajectories ordered by arrival time $\mathcal{D} = \left\{\mathbf{s}_i^e(t)\right\}_{i=1,\ldots,M}$
  **for** $i = 0, 1, \ldots, M$ **do**
    Obtain start state, $\mathbf{s}_0$, and end state, $\mathbf{s}_g$, of $\mathbf{s}_i^e(t)$
    Obtain other airplanes' trajectories, $\mathbf{s}_k^o(t)$ for $k < i$
    Use ARA* planner to minimize $J_\theta$ over $\mathbf{s}(t)$
    Compute stochastic gradient:

$$\hat{\nabla}_\theta \mathcal{L} = \sum_{t=t_0}^{T} f\left(\mathbf{s}(t)\right) - \sum_{t=t_0}^{T_i} f\left(\mathbf{s}_i^e(t)\right)$$

    Update cost parameters $\theta$ with step size $\alpha$:

$$\theta_{t+1} = \theta_t + \alpha \hat{\nabla}_\theta \mathcal{L}$$

  **end for**
**end loop**

- The motion cost must be:
  - $\Rightarrow$ Non-negative
  - $\Rightarrow$ Piece-wise linear in features of state $\mathbf{f}(\mathbf{s})$, to satisfy the assumptions of [ZMBD08]

- The motion cost must be:
  - $\Rightarrow$ Non-negative
  - $\Rightarrow$ Piece-wise linear in features of state $\mathbf{f(s)}$, to satisfy the assumptions of [ZMBD08]
- The spatial potential field $\bar{J}_a$ is approximated for each state $\bar{\mathbf{s}} \in \mathcal{G}$
  - $\Rightarrow$ Using a fine discretization of the input space

$$\bar{J}_a(\bar{\mathbf{s}}(t)) = \sum_{\bar{\mathbf{s}}_i \in \mathcal{G}} \max\left\{w_i, 0\right\} \mathbb{1}_{\bar{\mathbf{s}}(t)=\bar{\mathbf{s}}_i}\left(\bar{\mathbf{s}}(t)\right), \tag{10}$$

- The motion cost must be:
  - $\Rightarrow$ Non-negative
  - $\Rightarrow$ Piece-wise linear in features of state $\mathbf{f(s)}$, to satisfy the assumptions of [ZMBD08]
- The spatial potential field $\bar{J}_a$ is approximated for each state $\bar{\mathbf{s}} \in \mathcal{G}$
  - $\Rightarrow$ Using a fine discretization of the input space

$$\bar{J}_a(\bar{\mathbf{s}}(t)) = \sum_{\bar{\mathbf{s}}_i \in \mathcal{G}} \max\{w_i, 0\} \mathbb{1}_{\bar{\mathbf{s}}(t)=\bar{\mathbf{s}}_i}(\bar{\mathbf{s}}(t)), \tag{10}$$

- Costs $w_i$ for each discrete state in the space are stored in a look-up table
  - $\Rightarrow$ Enables more efficient motion planning

# Collision Avoidance Potential

- To control the airspace around each airplane, we penalize pairwise distances between airplanes.
- Each prior arrival is treated as a moving cylinder with a known trajectory, $\bar{\mathbf{s}}_k^o(t)$

- To control the airspace around each airplane, we penalize pairwise distances between airplanes.
- Each prior arrival is treated as a moving cylinder with a known trajectory, $\bar{\mathbf{s}}_k^o(t)$
- We use a linear drop-off potential function [MMA00]:

$$\bar{J}_o\big(\bar{\mathbf{s}}(t), \bar{\mathbf{s}}_k^o(t)\big) = \mu \max\left\{\lambda_z - |\Delta_z(t)|, 0\right\} \cdot \max\left\{\lambda_{xy} - \sqrt{\Delta_x^2(t) + \Delta_y^2(t)}, 0\right\}, \tag{11}$$

- $\Delta(t) = \bar{\mathbf{s}}(t) - \bar{\mathbf{s}}_k^o(t)$ is the distance between two airplanes at time $t$
  $\Rightarrow$ $\Delta_x(t)$, $\Delta_y(t)$ and $\Delta_z(t)$ are its components

- To control the airspace around each airplane, we penalize pairwise distances between airplanes.
- Each prior arrival is treated as a moving cylinder with a known trajectory, $\bar{\mathbf{s}}_k^o(t)$
- We use a linear drop-off potential function [MMA00]:

$$\bar{J}_o\big(\bar{\mathbf{s}}(t), \bar{\mathbf{s}}_k^o(t)\big) = \mu \max\big\{\lambda_z - |\Delta_z(t)|, 0\big\} \cdot \max\Big\{\lambda_{xy} - \sqrt{\Delta_x^2(t) + \Delta_y^2(t)}, 0\Big\}, \tag{11}$$

- $\Delta(t) = \bar{\mathbf{s}}(t) - \bar{\mathbf{s}}_k^o(t)$ is the distance between two airplanes at time $t$

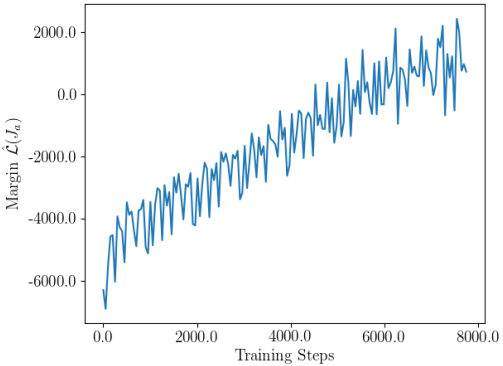
  $\Rightarrow$ $\Delta_x(t)$, $\Delta_y(t)$ and $\Delta_z(t)$ are its components

- Finally, the parameters for $\bar{J}_a$ and $\bar{J}_o$ can now be learned through stochastic gradient descent:

$$\theta = \Big[\{w_i\}_{\bar{\mathbf{s}}_i \in \mathcal{G}}, \lambda_{xy}, \lambda_z\Big] \tag{12}$$

- The margin between the expert's cost and the learner's cost increases during training

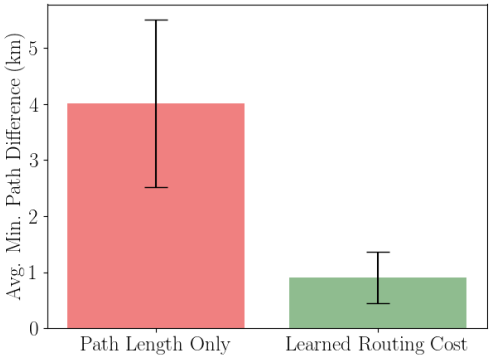
$$\hat{\mathcal{L}}(\bar{J}_a) = \sum_{t=t_0}^{T_i} \bar{J}_a\big(\bar{\mathbf{s}}(t)\big) - \sum_{t=t_0}^{T_i} \bar{J}_a\big(\bar{\mathbf{s}}_i^e(t)\big) \tag{13}$$



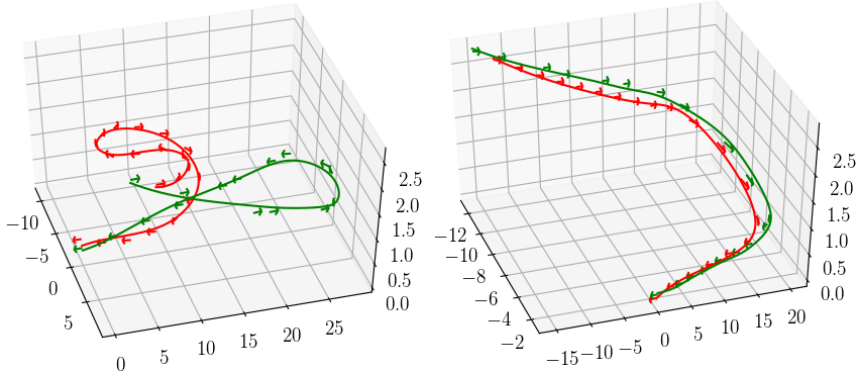
- The average minimum path difference

  $\Rightarrow$ How far the learner's trajectory strays from the expert's demonstration on average.

$$\frac{1}{N}\sum_{i=1}^{N}\frac{1}{T}\sum_{t=1}^{T}\min_{u}||\bar{\mathbf{s}}_i(t)-\bar{\mathbf{s}}_i^e(u)||_2$$



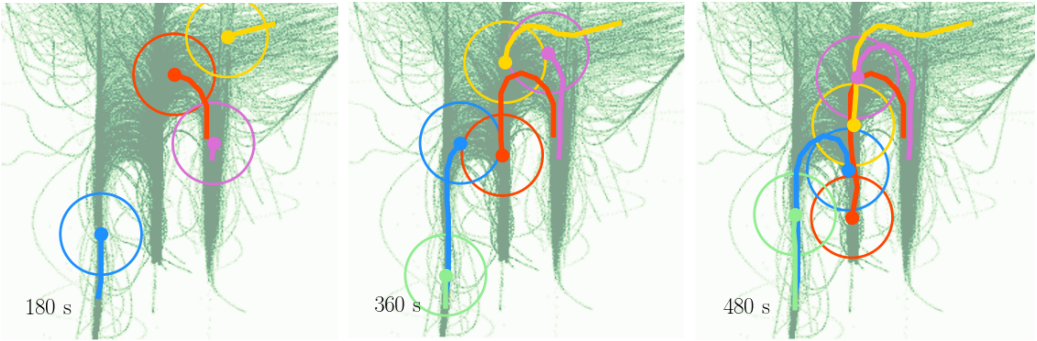
- A comparison of the expert's and learner's 3D trajectories for two different objective functions.
  - ⇒ The actual ATC trajectory is denoted with green arrows and a green spline.
  - ⇒ The planner's path is marked with red arrows and interpolated by the red curve.




(a) Minimum path length

(b) Using $\bar{J}_a$

- A top down view of five trajectories produced by the learner using both $\bar{J}_a$ and $\bar{J}_o$.
- The large unfilled circles denote the threshold of the $\bar{J}_o$ cost, maintaining separation
- The learned cost function $\bar{J}_a$ is in the background indicated by green shading, with dark green indicating low cost states, and white indicating high cost states.




(a) $t = 180$ s (b) $t = 360$ s (c) $t = 480$ s

# Inverse Optimal Planning for Air Traffic Control

Ekaterina Tolstaya[1†], Alejandro Ribeiro[1], Vijay Kumar[1], Ashish Kapoor[2]

[1] University of Pennsylvania
[2] Microsoft Corporation
[†] eig@seas.upenn.edu

International Conference on Intelligent Robots and Systems
November 7, 2019

# References


FlightAware, *Seattle-tacoma intl airport*, 2019.

Edwin T Jaynes, *Information theory and statistical mechanics*, Physical review **106** (1957), no. 4, 620.

Mrinal Kalakrishnan, Peter Pastor, Ludovic Righetti, and Stefan Schaal, *Learning objective functions for manipulation*, 2013 IEEE Int. Conf. on Robotics and Automation, IEEE, 2013, pp. 1331–1336.

Sikang Liu, Nikolay Atanasov, Kartik Mohta, and Vijay Kumar, *Search-based motion planning for quadrotors using linear quadratic minimum time control*, Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ Int. Conf. on, IEEE, 2017, pp. 2872–2879.

Maxim Likhachev, Geoffrey J Gordon, and Sebastian Thrun, *Ara*: Anytime a* with provable bounds on sub-optimality*, Advances in neural information processing systems, 2004, pp. 767–774.

Sikang Liu, Kartik Mohta, Nikolay Atanasov, and Vijay Kumar, *Towards search-based motion planning for micro aerial vehicles*, arXiv preprint arXiv:1810.03071 (2018).

Robin Murphy, Robin R Murphy, and Ronald C Arkin, *Introduction to ai robotics*, MIT press, 2000.

Mark Owen, Randal W Beard, and Timothy W McLain, *Implementing dubins airplane paths on*