Probabilistic Robust Motion Planning for UAVs

Mangal Kothari
Assistant Professor
Department of Aerospace Engineering
Indian Institute of Technology Kanpur
Kanpur 208016
mangal@iitk.ac.in
Finding a path is not enough !!!
Successful execution requires robust planning.

Challenges

- Localization uncertainty
- Modelling uncertainty
- Perception uncertainty
- Wind disturbance
- Situation awareness
- Motion constraints (e.g. turn radius)
Problem Formulation

Consider the following stochastic system

\[ x_{t+1} = f(x_t, u_t) + w_t \]
\[ x_0 \sim \mathcal{N}(\hat{x}_0, \Sigma_{x_0}) \]
\[ w_t \sim \mathcal{N}(0, \Sigma_{w_t}) \]

subject to

\[ u_t \in \mathcal{U} \]
\[ \Pr(x_t \notin \mathcal{X}_{\text{free}}) \leq \Delta \]

Goal

\[ x_t \subset \mathcal{X}_{\text{goal}} \]

Minimum time problem

\[ t_{\text{goal}} = \inf\{ t \in \mathbb{Z}_{0,t_f} \mid x_t \in \mathcal{X}_{\text{goal}} \} \]
Path Planning Algorithms

Planning Representations

Cell Decomposition
- Approximate
  - Regular Grids
- Exact
- Adaptive
  - Quad–tree
  - Framed quad–tree

Roadmap Methods
- Visibility Graph
- Voronoi Graph
- Probabilistic Roadmaps
- Rapidly Exploring Random Trees

Potential Fields

Search Algorithms

Depth First
- Iterative Deepening

Breadth First
- Uniform Cost
  - Wavefront NF1
  - Wavefront NF2
  - Trulla
  - Dijkstra’s Algorithm

Heuristic Search
- Best–first
  - A* 

Replanning Methods
- Learning Real–Time A* 
- D*
- Lifelong Planning A*
- D* Lite
Overview

• Cell decomposition method
• Potential Field
• Voronoi Diagram
• Visibility Line (VL)
• Probabilistic Roadmap (PRM)
• Rapidly-exploring Random Tree (RRT)
Cell Decomposition

• Approximate cell decomposition

• Adaptive cell decomposition

• Exact cell decomposition
Approximate Cell Decomposition
Adaptive Cell Decomposition
Exact Cell Decomposition

=> Connectivity graph
Potential Field

(c) Potential field for the goal

(d) Sum of potential fields from Obs1, Obs2 and Goal

Local minima here
Voronoi Diagram

Start point

Target point
Visibility Line

Start point

Target point

X-Range (km)

Y-Range (km)
Probabilistic roadmap (PRM)
Rapidly-exploring Random Tree (RRT)
Search Algorithms

1. Breadth-First Search (BFS)

2. Depth-First Search (DFS)

3. Dijkstra’s Algorithm

4. Best-First Search

5. A star (A*)
Breadth-First Search (BFS)

Step 1: Explore paths \( A \rightarrow B \)
(Goal not found) \( A \rightarrow C \)
\( A \rightarrow D \)

Step 2: Explore paths \( A \rightarrow B \rightarrow E \)
(Goal not found) \( A \rightarrow B \rightarrow F \)
\( A \rightarrow C \rightarrow G \)
\( A \rightarrow D \rightarrow H \)

Step 3: Explore paths \( A \rightarrow C \rightarrow G \rightarrow \text{Goal} \)
(Goal found)

In the event of tie, the left node is chosen first.
Depth-First Search (DFS)

Step 1: Explore paths $A \rightarrow B$
(Goal not found)

Step 2: Explore paths $A \rightarrow B \rightarrow E$
(Goal not found) $A \rightarrow B \rightarrow F$

Step 3: Explore paths $A \rightarrow C$
(Goal not found)

Step 4: Explore paths $A \rightarrow C \rightarrow G$
(Goal not found)

Step 5: Explore paths $A \rightarrow C \rightarrow G \rightarrow$ Goal
(Goal found)

In the event of tie, the left node is chosen first.
Dijkstra Algorithm

- Dijkstra algorithm is used in graphs with varying costs of traversal.

- The cost is usually the length of the edge.

- Using this algorithm, one can find the shortest paths from a start node to all points in a graph if the cost is minimum.

- Dijkstra algorithm is guaranteed to find the shortest path.
A Star (A*)

• Use Best-First search (similar to Depth-First algorithm along with a heuristic to determine the next move)
• Use Best-First search and Dijkstra algorithm to estimate the distance to goal and distance to start, respectively

• The cost function can be expressed as follows:

\[ f(n) = h(n) + g(n) \]

where \( f(n) \) is the total cost of the node, \( h(n) \) is the heuristic value of the node (from the goal), and \( g(n) \) is the cost from the start position to the node
Motion planning problem (MPP)

Consider a linear time-invariant system (LTI)

\[ x_{t+1} = Ax_t + Bu_t \]

Subject to

- Physical constraints
  \[ u_t \in \mathcal{U} \]
- Environmental constraints
  \[ x_t \in \mathcal{X}_{\text{free}} \]

\[ \mathcal{X}_{\text{free}} \equiv \mathcal{X} - \mathcal{X}_1 - \cdots - \mathcal{X}_B \]
Evaluating Constraint for Collision Avoidance

Collision with the $j^{th}$ obstacle (conjunction):

\[
\bigwedge_{i=1}^{n_j} a_{ij}^T x_i < b_{ij}
\]

In order to avoid collision (disjunctions):

\[
\bigvee_{i=1}^{n_j} a_{ij}^T x_i \geq b_{ij}
\]

- When the vehicle state is a random variable !!!
  - These constraints are not applicable 😞
  - Another approach need to be considered
Stochastic Environments

In the probabilistic framework, compute

$$Pr \left( \bigcap_{i=1}^{n_j} a^T_{ij} x_t < b_{ij} \right)$$

Enforce

$$Pr \left( \bigcap_{i=1}^{n_j} a^T_{ij} x_t < b_{ij} \right) \leq \Delta$$

Joint chance constraint
Evaluation of the Probabilistic Constraint

Simplify the joint chance constraint for the $j^{th}$ obstacle

$$\Pr \left( \bigwedge_{i=1}^{n_j} a_{ij}^T x_t < b_{ij} \right) \leq \Pr \left( a_{ij}^T x_t < b_{ij} \right), \quad \forall \ i \in \mathbb{Z}_{1,n_j}$$

If

$$\bigvee_{i=1}^{n_j} \Pr (a_{ij}^T x_t < b_{ij}) \leq \Delta$$

then

$$\Pr \left( \bigwedge_{i=1}^{n_j} a_{ij}^T x_t < b_{ij} \right) \leq \Delta$$

For probabilistic satisfaction, it is enough to show that one of surfaces satisfies the above inequality.
Converting into the Equivalent Deterministic Constraint

Chance constraint

\[ Pr(a_{ij}^T x_t < b_{ij}) \leq \Delta \]

Change of variable

\[ V_{ijt} = a_{ij}^T x_t - b_{ij} \]

 Converted constraint

\[ Pr(V_{ijt} < 0) \leq \Delta \]
Conversion: Constraint Tightening

The equivalent deterministic constraint

\[ Pr(V_{ijt} < 0) \leq \Delta \iff \hat{V}_{ijt} \geq \tilde{b}_{ijt} \]

\[ a_{ij}^T \hat{x}_t \geq b_{ij} + \tilde{b}_{ijt} \]

\[ \tilde{b}_{ijt} = \sqrt{2} \Sigma_v \text{erf}^{-1} (1 - 2\Delta) \]

The constraint tightening depends on uncertainty in the position and on the value of \( \Delta \).

\[ Pr \left( \bigwedge_{i=1}^{n_j} a_{ij}^T x_t < b_{ij} \right) \leq \Delta \]

\[ \bigvee_{i=1}^{n_j} a_{ij}^T \hat{x}_t \geq b_{ij} + \tilde{b}_{ijt} \]
Condition: Multiple Obstacles

Let \( F \triangleq x_t \not\in \mathcal{X}_{\text{free}} \) be an event

\[
Pr(F) = Pr\left( \bigcup_{i=1}^{B} x_t \in \mathcal{X}_i \right)
\]

From Boole’s bound, we can write

\[
Pr(F) \leq Pr\left( x_t \in \mathcal{X}_1 \right) + \ldots + Pr\left( x_t \in \mathcal{X}_B \right)
\]

By limiting

\[
Pr\left( x_t \in \mathcal{X}_i \right) \leq \frac{\Delta}{B}
\]

Risk allocation can be done!!!

It can be ensured

\[
Pr(F) \leq \sum_{i=1}^{B} \frac{\Delta}{B} = \Delta
\]
Problem Statement: Linear Systems

Find a sequence of control input that minimizes:

\[ J(u) = \inf \{ t \in \mathbb{Z}_0, \quad t_f \mid x_t \in X_{\text{goal}} \} + \sum_{t=0}^{t_f} \Phi(x_t) \]

subject to

Kalman Filter theory

\[
\dot{x}_t = A^t \hat{x}_0 + \sum_{k=0}^{t-1} A^{t-k-1} Bu_k, \quad \forall t \in \mathbb{Z}_{0,N},
\]

\[
\Sigma_{x_t} = A^t \Sigma_{x_0} (A^T)^t + \sum_{k=0}^{t-1} A^{t-k-1} \Sigma_{w} (A^T)^{t-k-1}
\]

\[
\sqrt[n_j]{a_{i,j}^T \hat{x}_t} \geq b_{i,j} + \bar{b}_{i,j}, \quad \forall \ j \in \mathbb{Z}_{1,B}
\]

\[
\bar{b}_{i,j} = \sqrt{2 a_{i,j}^T \Sigma_{x_t} a_{i,j} \text{erf}^{-1} \left( 1 - 2\frac{A}{B} \right)}
\]
Problem Statement: Nonlinear Systems

Find a sequence of control input that minimizes:

$$J(u) = \inf\{t \in \mathbb{Z}_0, t_f \mid x_t \in \mathcal{X}_{\text{goal}}\} + \sum_{t=0}^{t_f} \Phi(x_t)$$

subject to

\[ \hat{x}_{t+1} \triangleq x_{t+1|t} = f(x_{t|t}, u_t) \]
\[ \Sigma_{x_{t+1}} \triangleq \Sigma_{t+1|t} = F_t \Sigma_{t|t} F_t^T + Q \]

Prediction step of EKF: belief update (a priori distribution)

\[ \sqrt{\sum_{n_j} a_{ij}^T \hat{x}_{t}} \geq b_{ij} + \bar{b}_{ij}, \quad \forall \ j \in \mathbb{Z}_{1,B} \]
\[ \bar{b}_{ij} = \sqrt{2 a_{ij}^T \Sigma_{x_t} a_{ij}} \text{erf}^{-1} \left( 1 - 2 \frac{A}{B} \right) \]

MDP: consider every reachable belief state
Chance constrained RRT (CC-RRT) Algorithm

- Grow a tree of state distributions for a given time
  - sample reference path (similar to waypoint selection)
  - generate trajectory for the sampled path (use a control/guidance law to generate trajectory)
  - evaluate the feasibility of the generated trajectory (using chance constraint)
  - include the path in the existing tree if it is feasible

CC-RRT: Tree expansion

Closed-loop prediction (a priori distribution)
Trajectory Generation: Path Following

Fixed-wing UAV kinematic model

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\psi} \\
\dot{\phi}
\end{bmatrix} =
\begin{bmatrix}
v_a \cos \psi \\
v_a \sin \psi \\
g \tan \phi \\
-k(\phi - \phi^d)
\end{bmatrix}
\]

Path following law: pursuit and LOS components

\[
\phi^d = \tan^{-1} \left( \frac{k_1(\psi_r - \psi) + k_2d}{g} \right)
\]

Error dynamics

\[
\begin{align*}
\dot{d} &= v_a \sin(\psi_r - \psi) \\
\dot{\psi} &= -\frac{k_1}{v_a}(\psi_r - \psi) - \frac{k_2}{v_a} d
\end{align*}
\]
Tree Expansion: Offline

Pr(Δ) = 0.5

Pr(Δ) = 0.1
Numerical Results: Offline

Paths with different $Pr(\Delta)$
Online Implementation

• Employ a look-ahead strategy
  • Expansion – grow the tree for the given time window
    • with emphasis on exploration and optimization heuristics
    • branch and bound method to keep only promising nodes
  • Execution – choose the best path for execution and update the information (pop-up and dynamic obstacles)
    • re-evaluate feasibility of the tree when new measurements are received
Numerical Results: Online

Sample tree after 1 sec

Sample tree after execution of the first segment

Complete tracked path
Incorporation of a Sensor Model

Consider the following nonlinear Gaussian system

\[ x_{t+1} = f(x_t, u_t) + w_t \]
\[ z_t = h(x_t) + v_t \]

Let \( x_t^e \triangleq x_t - x_t^* \) be the error between the nominal and actual systems. The linearized error dynamics is given as

\[ x_{t+1}^e = A_t x_t^e + B_t u_t^e + w_t \]
\[ z_t^e = H_t x_t^e + v_t \]

Compute \textit{a priori} closed-loop distribution

\[ \hat{x}_t \triangleq \mathbb{E}[x_t] = \mathbb{E}[x_t^*] + \mathbb{E}[x_t^e] \]
\[ \Sigma_{x_t} \triangleq \mathbb{E} \left[ (x_t - \mathbb{E}[x_t]) (x_t - \mathbb{E}[x_t])^T \right] \]
Closed-loop Distributions

The measurement step of Kalman Filter is given by

\[
\begin{align*}
    x_{t+1|t+1} &= x_{t|t} + L_{t+1} \left( z_{t+1} - H_t \, x_{t+1|t} \right) \\
    \Sigma_{t+1|t+1} &= (I - L_{t+1} \, H_t) \ A_t \ \Sigma_{t+1|t} 
\end{align*}
\]

where

\[
L_{t+1} = \Sigma_{t+1|t} H_t^T (H_t \ \Sigma_{t+1|t} \ H_t^T + \Sigma_v)^{-1}
\]

The augmented system

\[
\xi_{t+1} \triangleq \begin{bmatrix} x_{t+1}^e \\ x_{t+1|t+1} \end{bmatrix} = \begin{bmatrix} A_t & 0 \\ L_{t+1} H_t A_t & A_t - L_{t+1} H_t A_t \end{bmatrix} \xi_t + \begin{bmatrix} B_t \\ B_t \end{bmatrix} u_t + \begin{bmatrix} I & 0 \\ L_{t+1} H_t & L_{t+1} \end{bmatrix} \begin{bmatrix} w_t \\ v_{t+1} \end{bmatrix}
\]
Closed-loop Update: Most likely Measurements

A priori distribution of the augmented system

\[
\begin{align*}
\hat{\xi}_{t+1} &= F_t \hat{\xi}_t + \bar{B}u_t \\
M_{t+1} &= F_t M_t F_t^T + G_t \Sigma_s G_t^T
\end{align*}
\]

Closed-loop belief update

\[
\hat{x}_t^e = \mathbf{E}[\Lambda \hat{\xi}_t] = \Lambda \hat{\xi}_t
\]

\[
\mathbf{E} \left[ (x_t - \mathbf{E}[x_t]) (x_t - \mathbf{E}[x_t])^T \right] = \mathbf{E} \left[ (x_t^e - \mathbf{E}[x_t^e]) (x_t^e - \mathbf{E}[x_t^e])^T \right] = \Lambda M_t \Lambda^T
\]

Closed-loop prediction (with most likely future measurements)
Numerical Results: A Single Agent System
A Two Agent System

(a) Agents moving towards their goal positions while avoiding conflict

(b) Probability of collision
Thank You