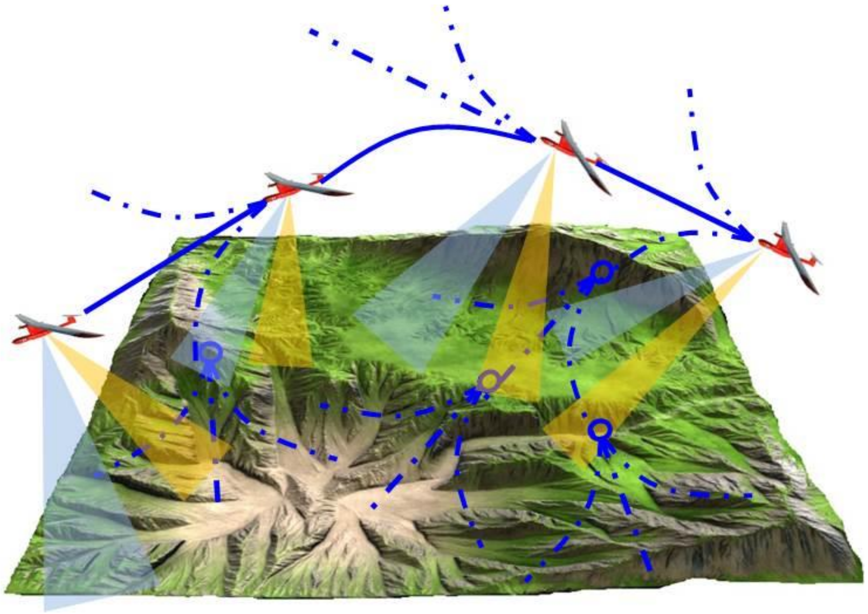
Probabilistic Robust Motion Planning for UAVs

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.



<u>Challenges</u>

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

Problem Formulation

Consider the following stochastic system

$$\boxed{\begin{aligned} x_{t+1} &= f(x_t, u_t) + w_t \\ x_0 &\sim \mathcal{N}(\hat{x}_0, \Sigma_{x_0}) \\ w_t &\sim \mathcal{N}(0, \Sigma_{w_t}) \end{aligned}}$$

Nonlinear Gaussian system

subject to

$$\boxed{\begin{aligned} u_t &\in \mathcal{U} \\ Pr(x_t \notin \mathcal{X}_{free}) &\leq \Delta \end{aligned}}$$

(e.g. turn radii constraint)

Probability of leaving configuration space

$$\boxed{\mathcal{X}_{free} \triangleq \mathcal{X} \setminus \{\mathcal{X}_1 \cup \mathcal{X}_2 \cdots \cup \mathcal{X}_B\}}$$

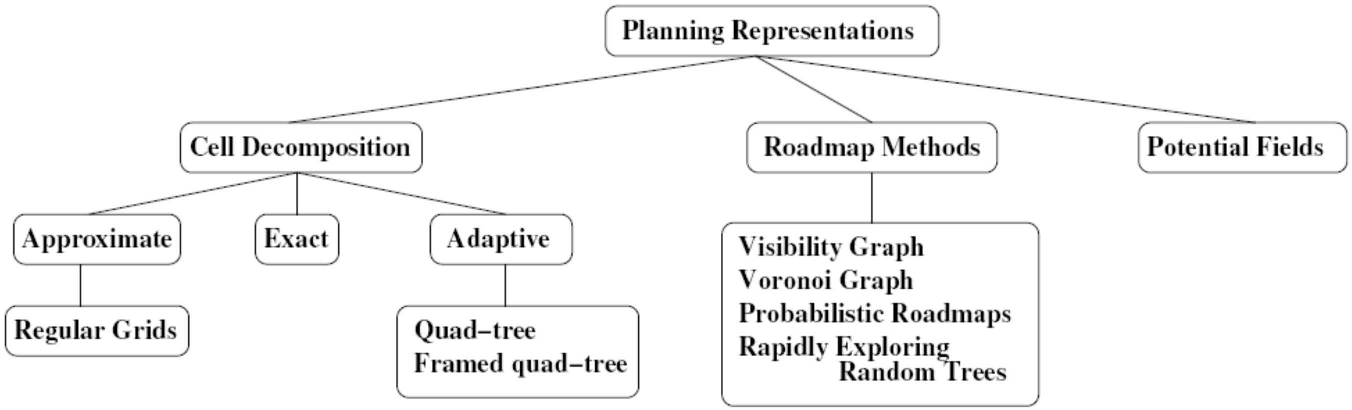
Goal

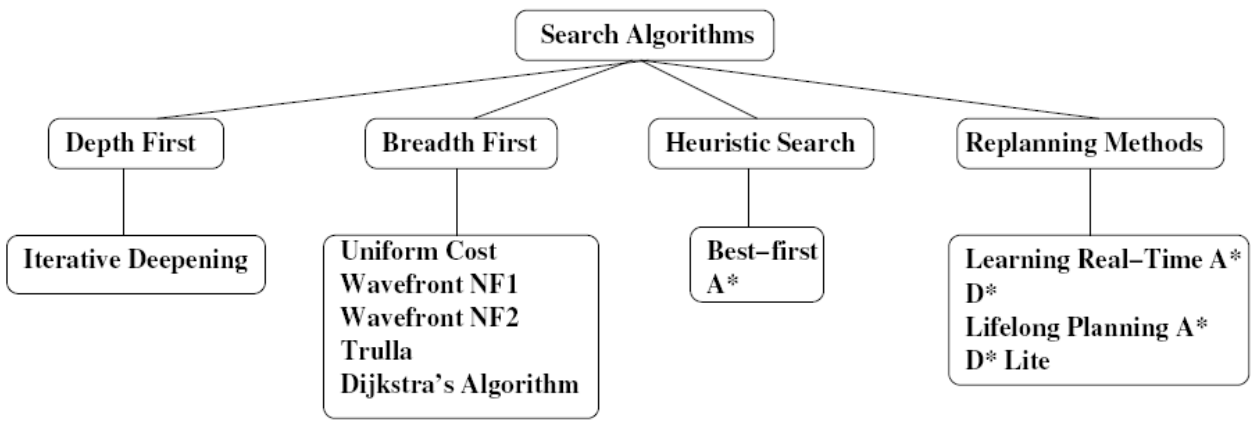
$$\boxed{x_t \subset \mathcal{X}_{\text{goal}}}$$

Minimum time problem

$$\boxed{t_{\text{goal}} = \inf\{t \in \mathbb{Z}_{0,t_f} \mid x_t \in \mathcal{X}_{\text{goal}}\}}$$

Path Planning Algorithms





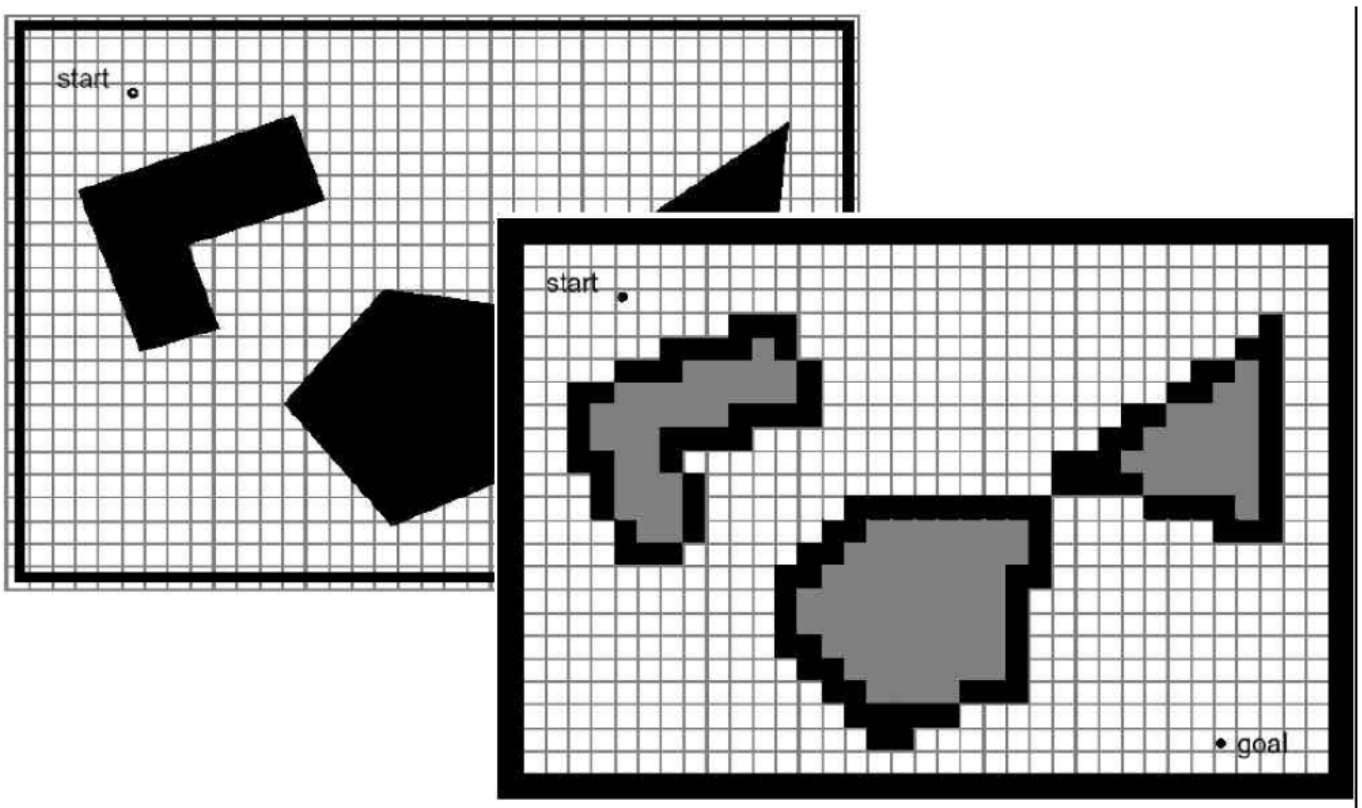
Overview

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

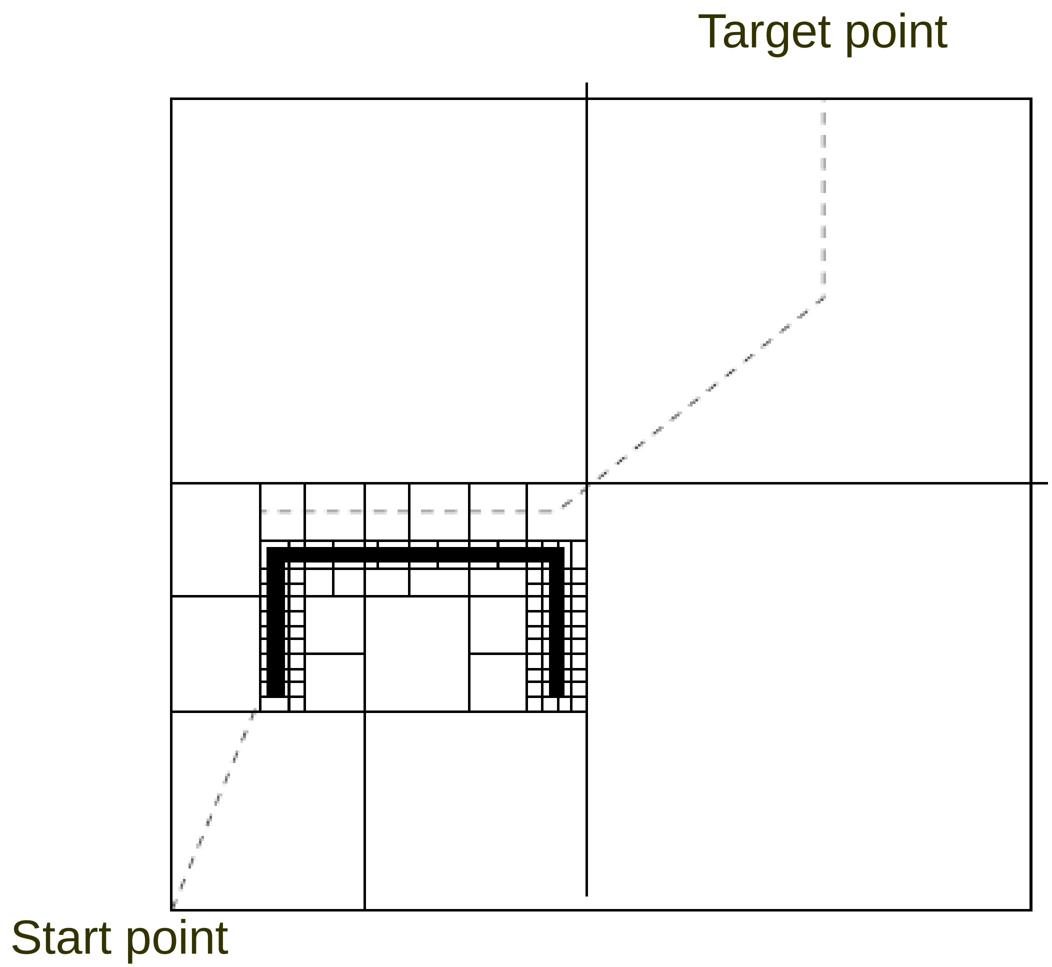
Cell Decomposition

- Approximate cell decomposition
- Adaptive cell decomposition
- Exact cell decomposition

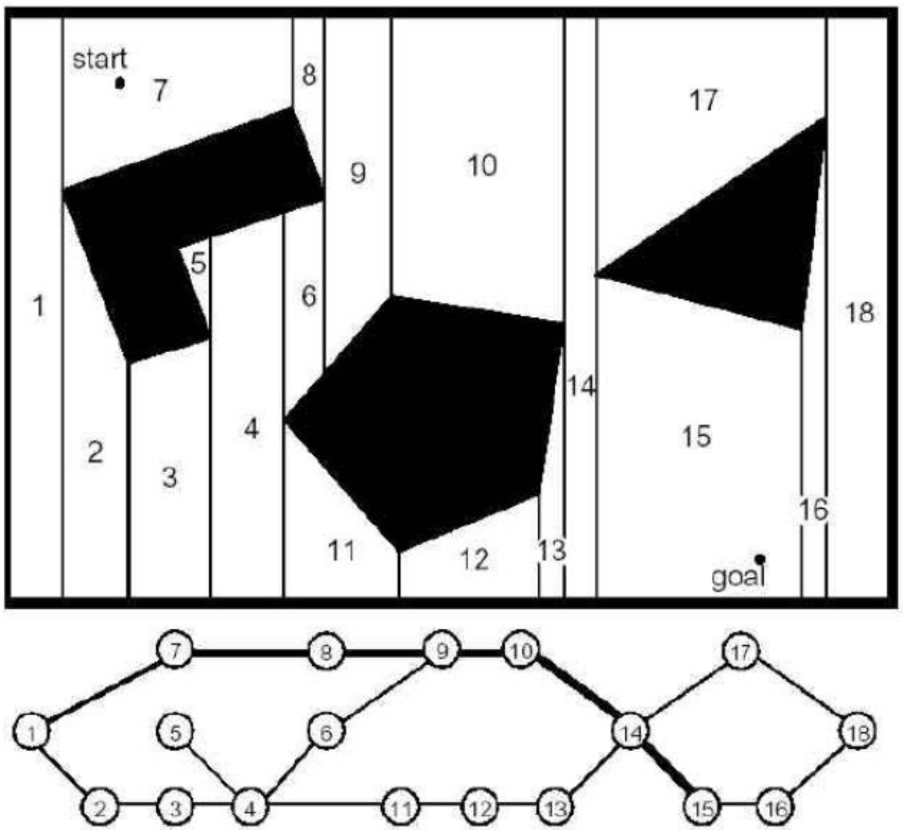
Approximate Cell Decomposition



Adaptive Cell Decomposition

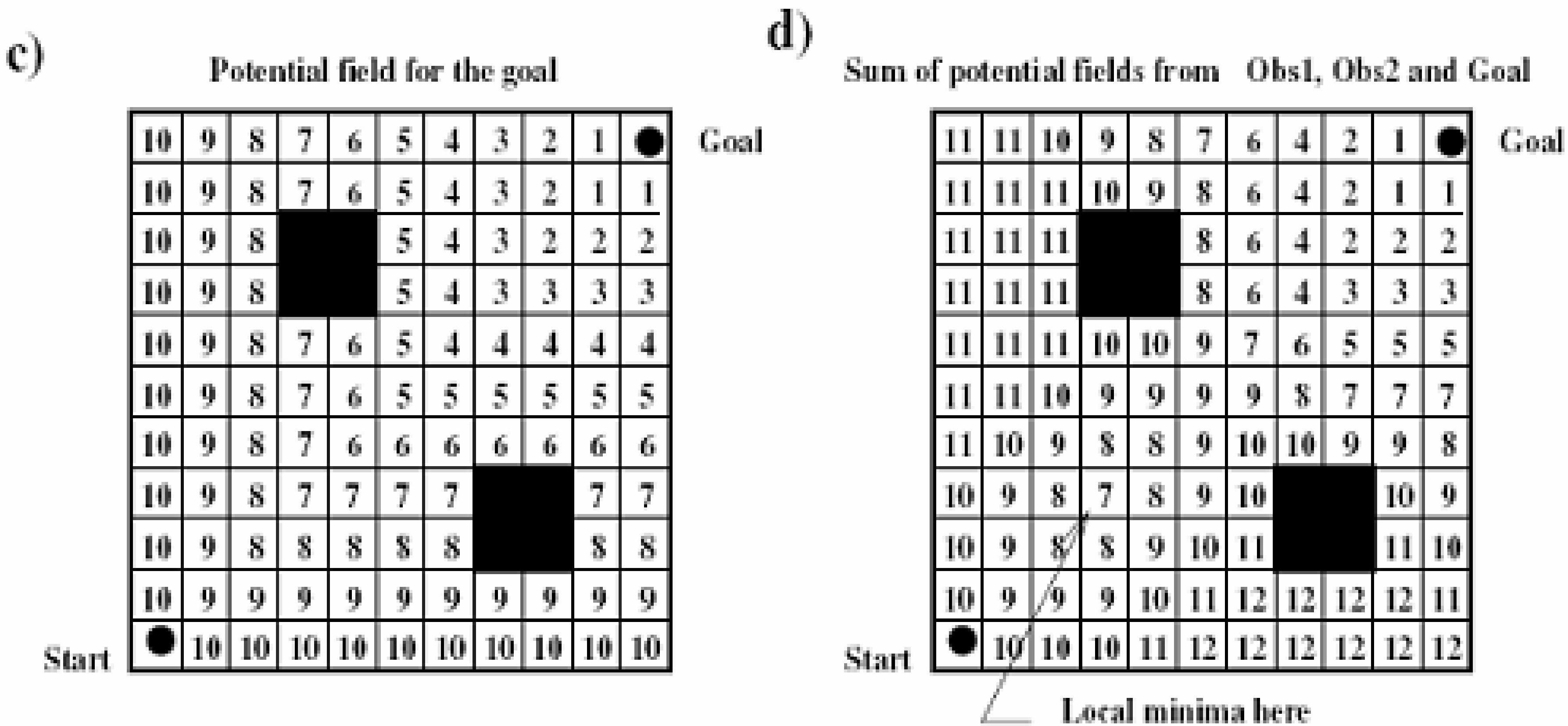


Exact Cell Decomposition

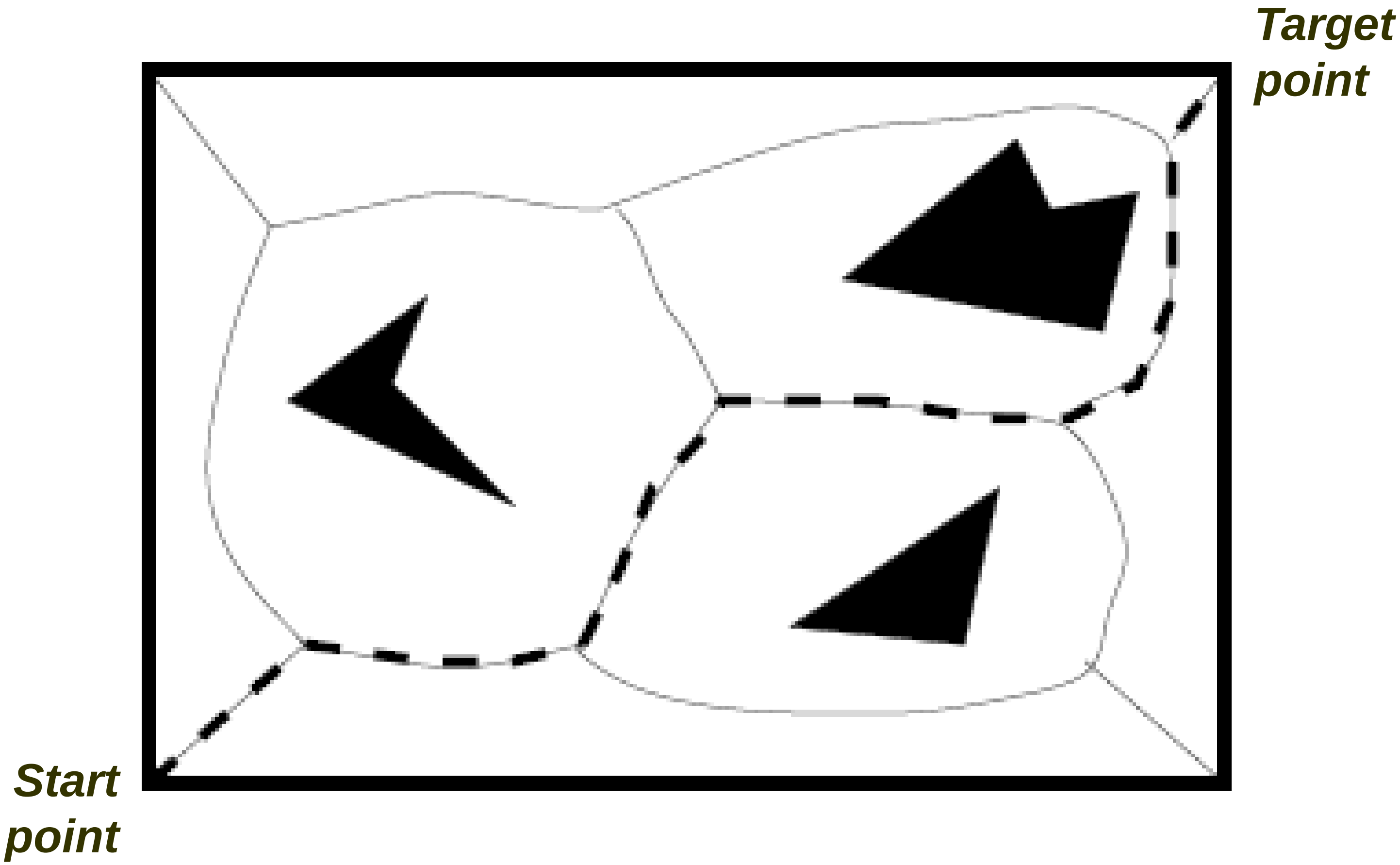


➔ Connectivity graph

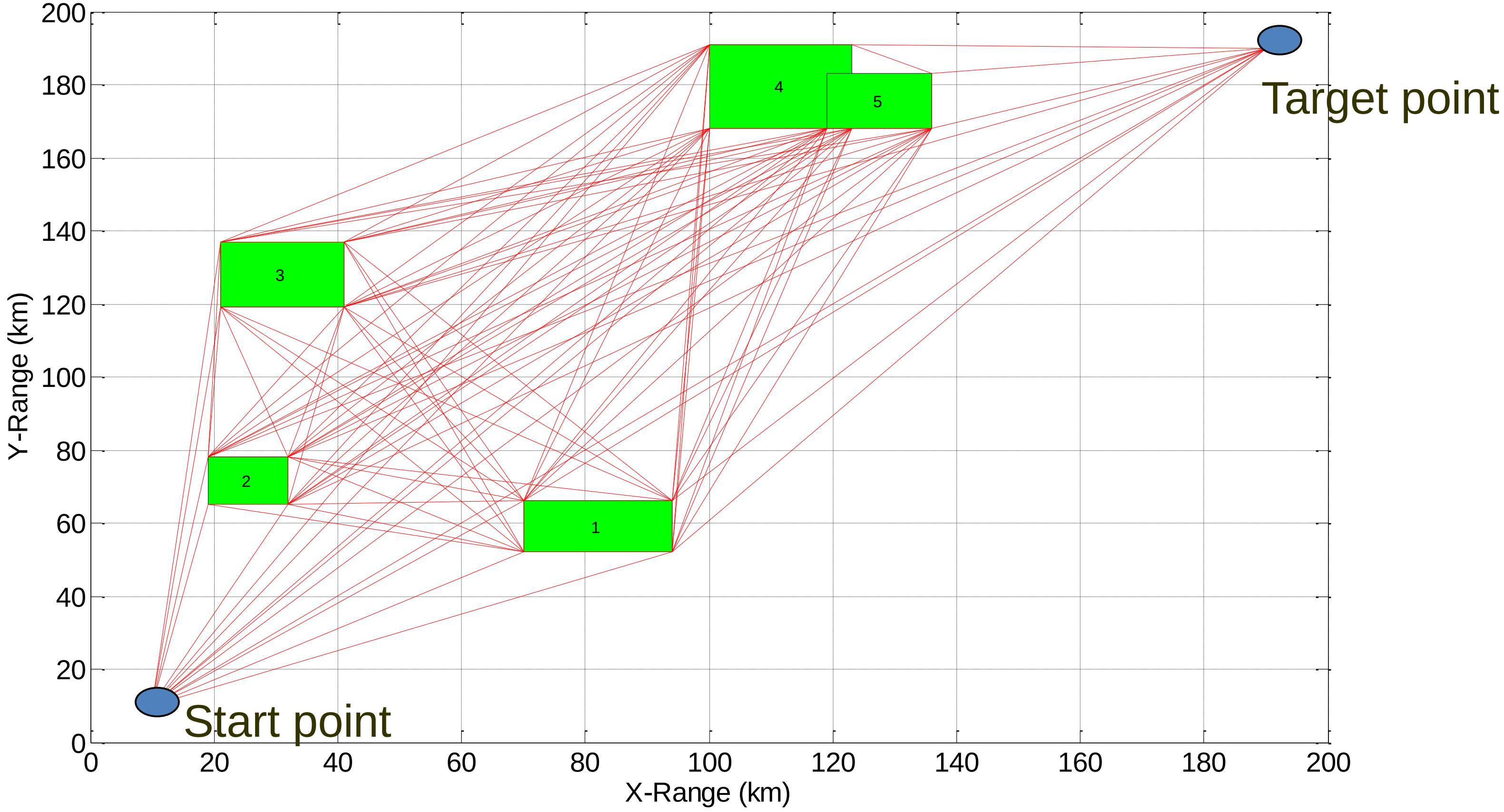
Potential Field



Voronoi Diagram

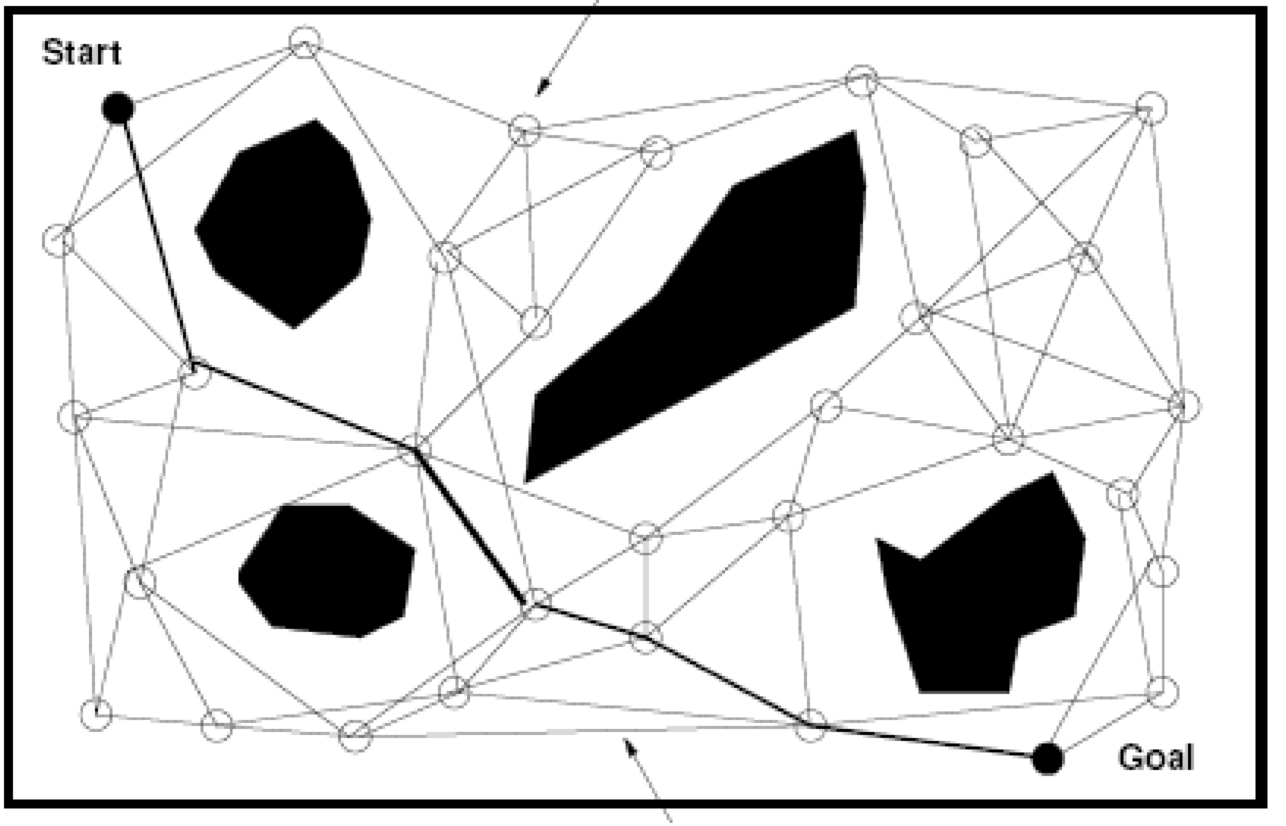


Visibility Line



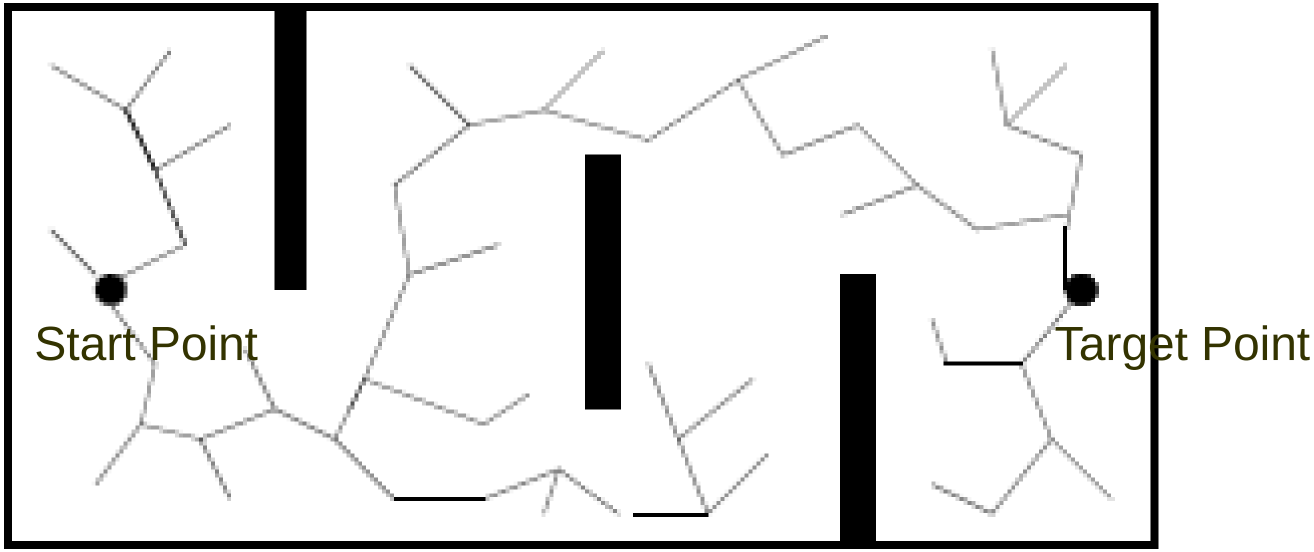
Probabilistic roadmap (PRM)

Randomly chosen point



Path searched by algorithm

Rapidly-exploring Random Tree (RRT)



Search Algorithms

1. Breadth-First Search (BFS)

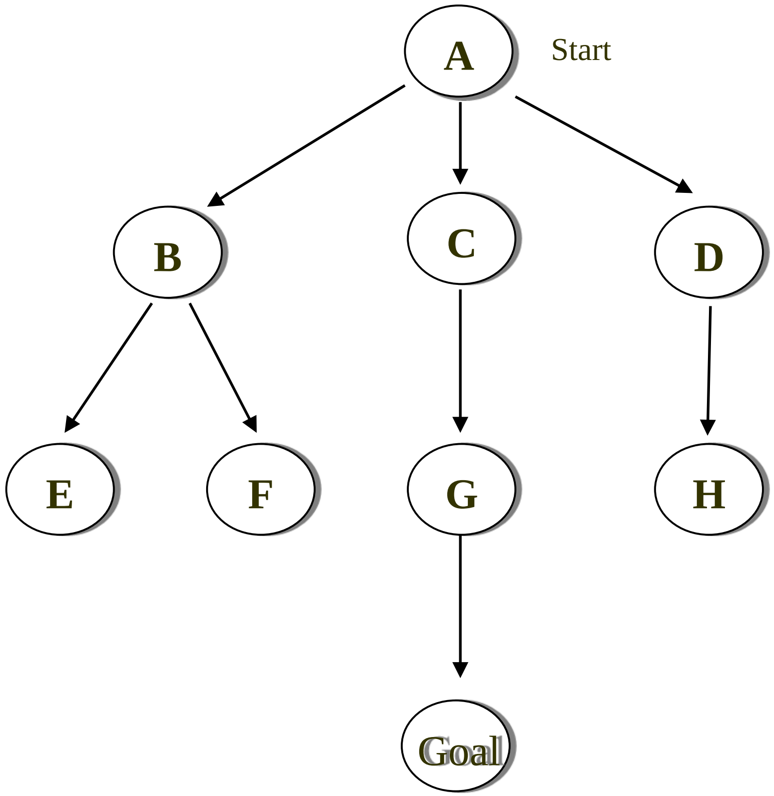
2. Depth-First Search (DFS)

3. Dijkstra's Algorithm

4. Best-First Search

5. A star (A*)

Breadth-First Search (BFS)



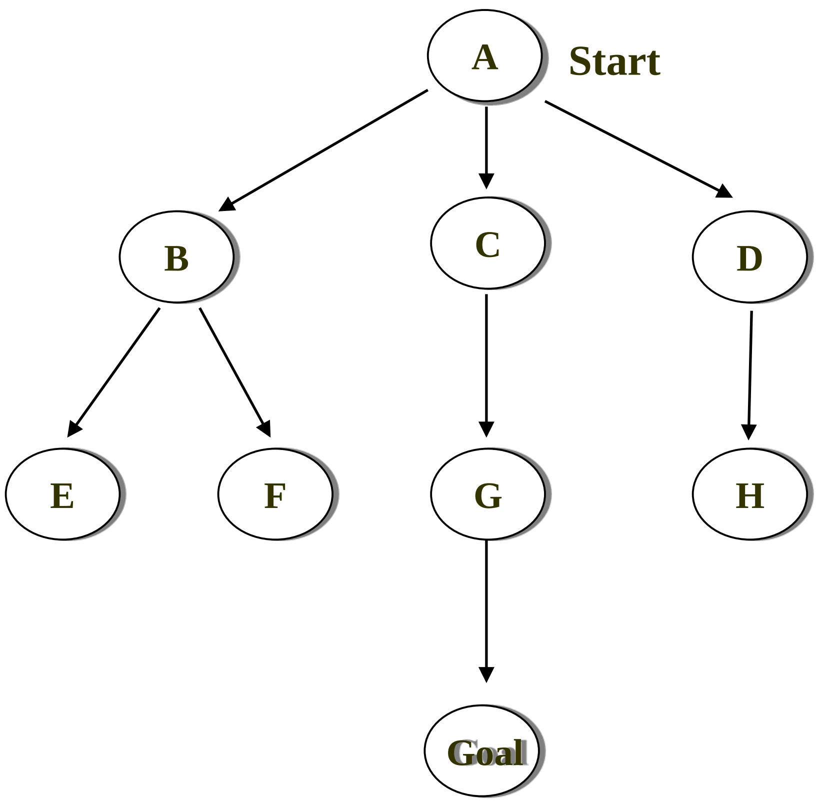
Step 1: Explore paths A → B
(Goal not found) A → C
A → D

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

Step 3 : Explore paths A→C→G→Goal
(Goal found)

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

Depth-First Search (DFS)



Step 1: Explore paths A → B
(Goal not found)

Step 2: Explore paths A → B → E
(Goal not found) A → B → F

Step 3: Explore paths A → C
(Goal not found)

Step 4 : Explore paths A → C → G
(Goal not found)

Step 5 : Explore paths A → C → G → 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*)

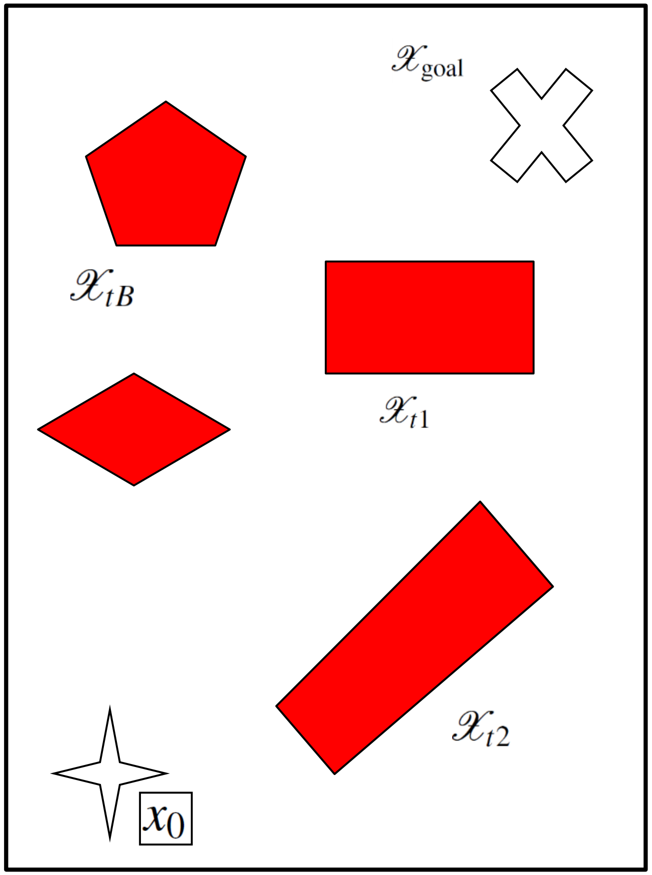
- Use Best-First search (similar to Depth-First algorithm along with a heuristic to determine the next move)
- Use Best-First search and Dijkstra algorithm to estimate the distance to goal and distance to start, respectively

- 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

$$u_t \in \mathscr{U}$$ Physical constraints

$$x_t \in \mathscr{X}_{free}$$ Environmental constraints

$$\mathscr{X}_{free} \equiv \mathscr{X} - \mathscr{X}_1 - \cdots - \mathscr{X}_B$$

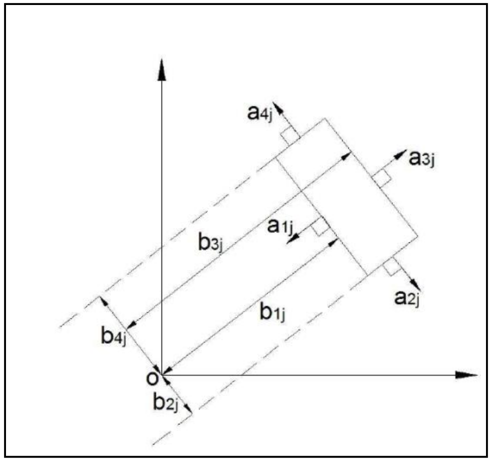
Evaluating Constraint for Collision Avoidance

Collision with the j[th] obstacle (conjunction):

$$\bigwedge_{i=1}^{n_j} a_{ij}^T x_t < b_{ij}$$

In order to avoid collision (disjunctions):

$$\bigvee_{i=1}^{n_j} a_{ij}^T x_t \geq b_{ij}$$

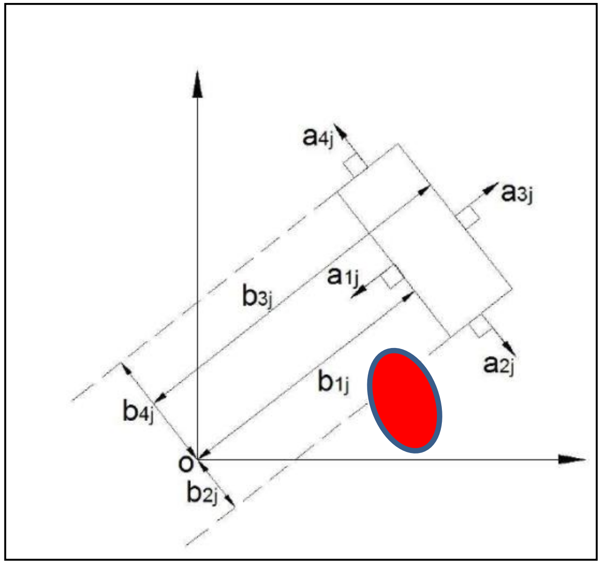


- When the vehicle state is a random variable !!!
  - These constraints are not applicable ☹
  - Another approach need to be considered

Stochastic Environments

In the probabilistic framework, compute

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



Enforce

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

Joint chance constraint

Evaluation of the Probabilistic Constraint

Simplify the joint chance constraint for the jth obstalce

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

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

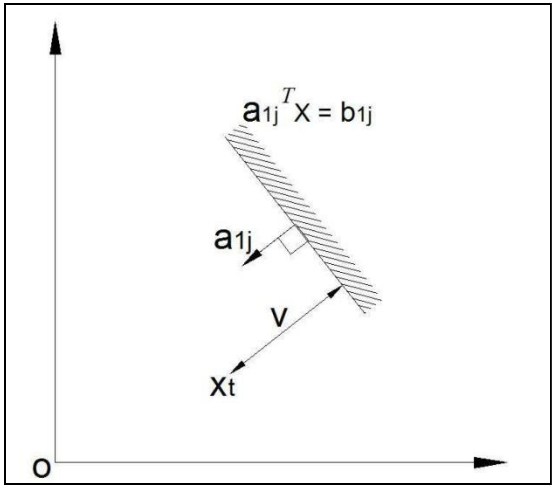
For probabilistic satisfaction, it is enough to show that one of surfaces satisfies the above inequality.

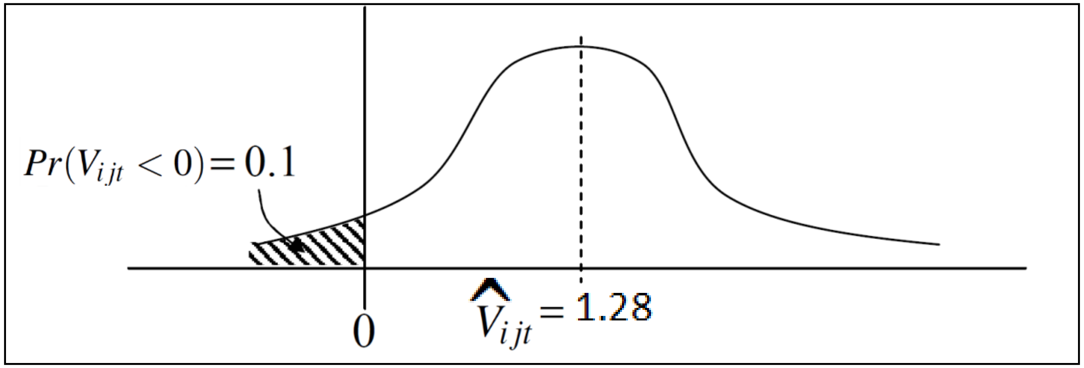
Converting into the Equivalent Deterministic Constraint

Chance constraint $\boxed{Pr(a_{ij}^T x_t < b_{ij}) \leq \Delta}$

Change of variable $\boxed{V_{ijt} = a_{ij}^T x_t - b_{ij}}$

Converted constraint $\boxed{Pr(V_{ijt} < 0) \leq \Delta}$





Conversion: Constraint Tightening

The equivalent deterministic constraint

$$Pr(V_{ijt} < 0) \leq \Delta \Leftrightarrow \hat{V}_{ijt} \geq \bar{b}_{ijt}$$

$$\Rightarrow$$

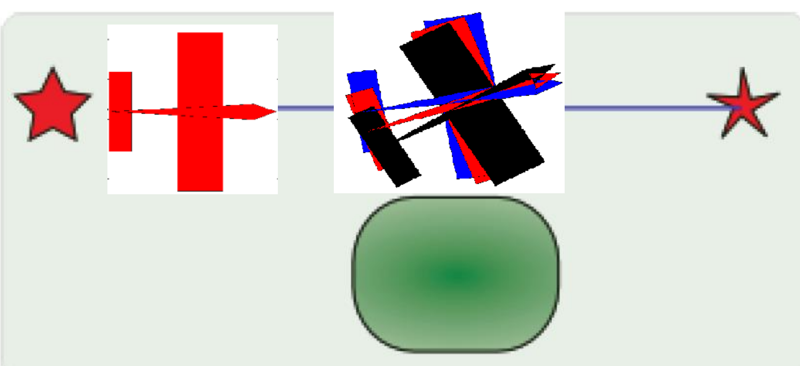
$$\begin{aligned} a_{ij}^T \hat{x}_t &\geq b_{ij} + \bar{b}_{ijt} \\ \bar{b}_{ijt} &= \sqrt{2}\Sigma_v \mathrm{erf}^{-1}(1 - 2\Delta) \end{aligned}$$

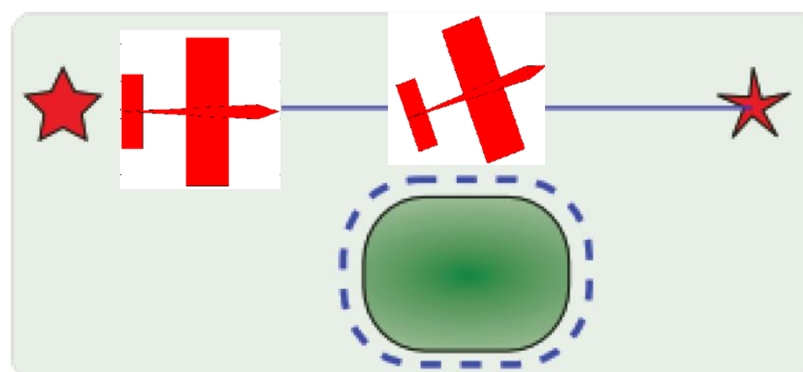
The constraint tightening depends on uncertainty in the position and on the value of Δ.

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

$$\Leftrightarrow$$

$$\bigvee_{i=1}^{n_j} a_{ij}^T \hat{x}_t \geq b_{ij} + \bar{b}_{ijt}$$





Condition: Multiple Obstacles

Let $F \triangleq x_t \notin \mathscr{X}_{free}$ be an event

$$\boxed{Pr(F) = Pr\left(\bigcup_{i=1}^{B} x_t \in \mathscr{X}_i\right)}$$

From Boole's bound, we can write

$$\boxed{Pr(F) \leq Pr\left(x_t \in \mathscr{X}_1\right) + \ldots + Pr\left(x_t \in \mathscr{X}_B\right)}$$

By limiting $\boxed{Pr\left(x_t \in \mathscr{X}_i\right) \leq \frac{\Delta}{B}}$ Risk allocation can be done!!!

It can be ensured $\boxed{Pr(F) \leq \sum_{i=1}^{B} \frac{\Delta}{B} = \Delta}$

Problem Statement: Linear Systems

Find a sequence of control input that minimizes:

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

subject to

$$\begin{aligned} \hat{x}_t &= A^t \hat{x}_0 + \sum_{k=0}^{t-1} A^{t-k-1} B u_k, \quad \forall\, t \in \mathbb{Z}_{0,N}, \\ \Sigma_{x_t} &= A^t \Sigma_{x_0} (A^T)^t + \sum_{k=0}^{t-1} A^{t-k-1} \Sigma_w (A^T)^{t-k-1} \end{aligned}$$

Kalman Filter theory

$$\begin{aligned} \bigvee^{n_j} a_{ij}^T \hat{x}_t &\geq b_{ij} + \bar{b}_{ijt}, \quad \forall\, j \in \mathbb{Z}_{1,B} \\ \bar{b}_{ijt} &= \sqrt{2 a_{ij}^T \Sigma_{x_t} a_{ij}}\ \mathrm{erf}^{-1}\left(1 - 2\tfrac{\Delta}{B}\right) \end{aligned}$$

MDP/POMDP

Problem Statement: Nonlinear Systems

Find a sequence of control input that minimizes:

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

subject to

$$\boxed{\begin{aligned} \hat{x}_{t+1} &\triangleq x_{t+1|t} = f(x_{t|t}, u_t) \\ \Sigma_{x_{t+1}} &\triangleq \Sigma_{t+1|t} = F_t \Sigma_{t|t} F_t^T + Q \end{aligned}}$$

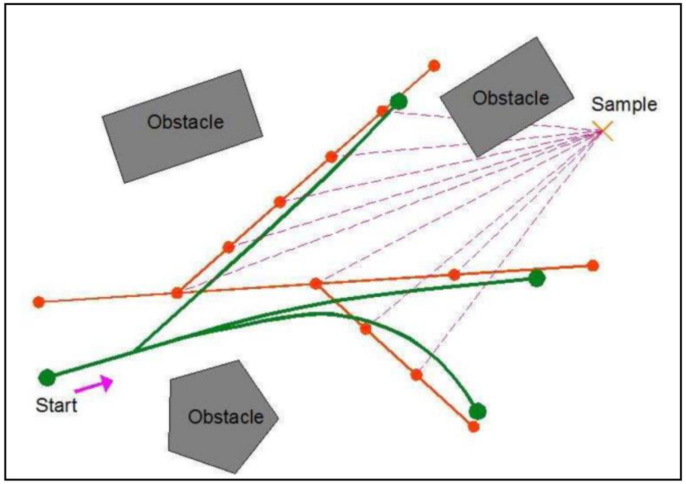
Prediction step of EKF: **belief update** (*a priori* distribution)

$$\boxed{\begin{aligned} \bigvee^{n_j} a_{ij}^T \hat{x}_t &\geq b_{ij} + \bar{b}_{ijt},\ \forall\, j \in \mathbb{Z}_{1,B} \\ \bar{b}_{ijt} &= \sqrt{2 a_{ij}^T \Sigma_{x_t} a_{ij}}\ \operatorname{erf}^{-1}\left(1 - 2\tfrac{\Delta}{B}\right) \end{aligned}}$$

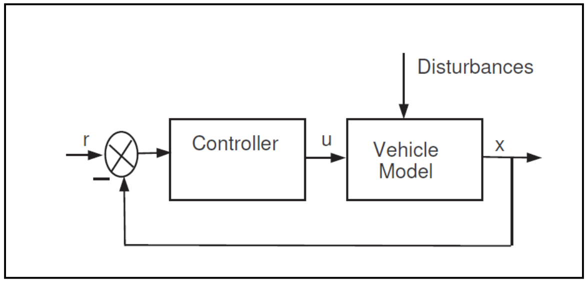
MDP: consider every reachable belief state

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

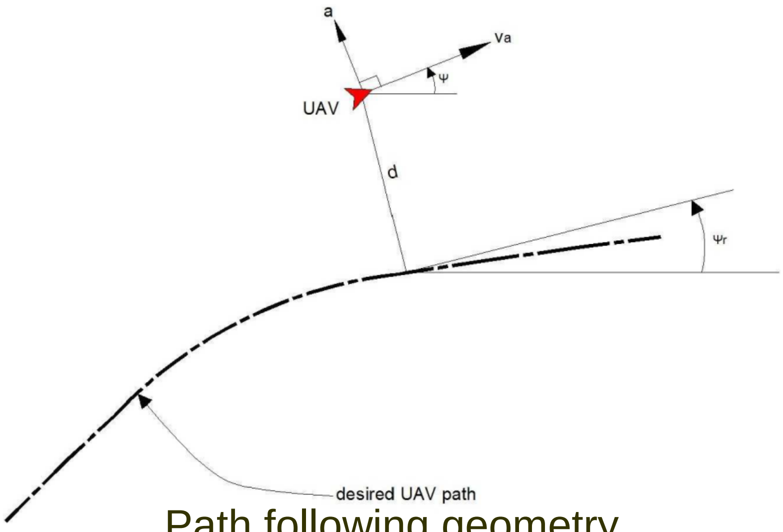


Closed-loop prediction (*a priori* distribution)

Trajectory Generation: Path Following

Fixed-wing UAV kinematic model

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



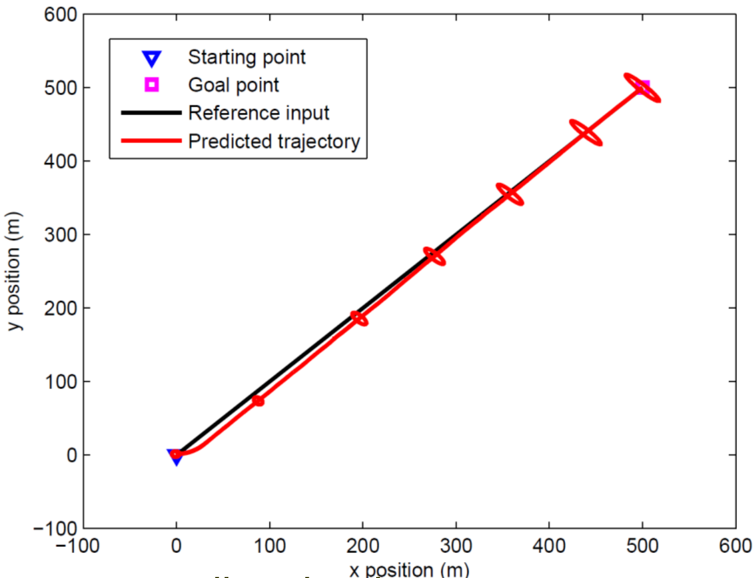
Path following geometry

Path following law: pursuit and LOS components

$$\phi^d = \tan^{-1}\left(\frac{k_1(\psi_r - \psi) + k_2 d}{g}\right)$$

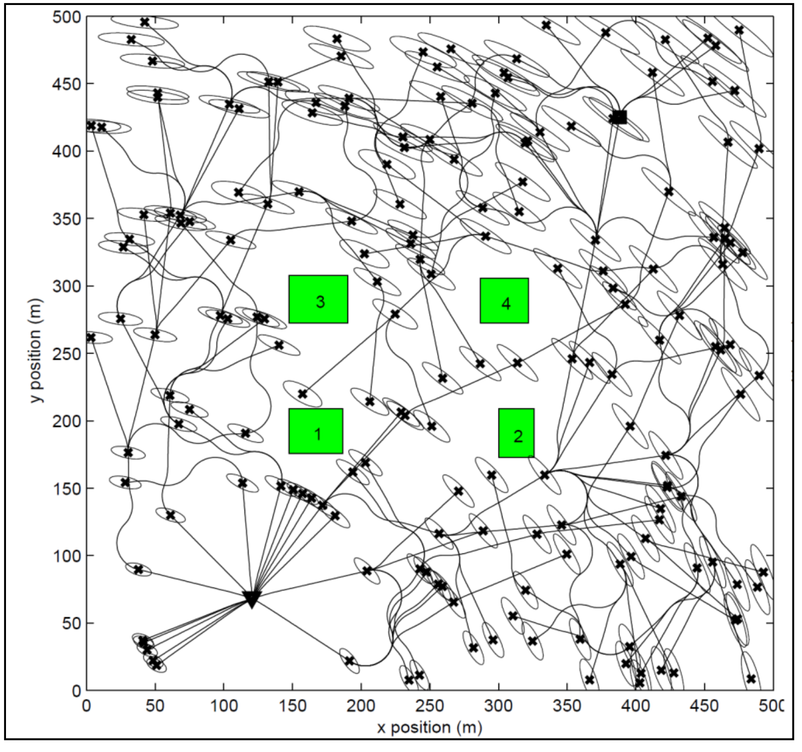
Error dynamics

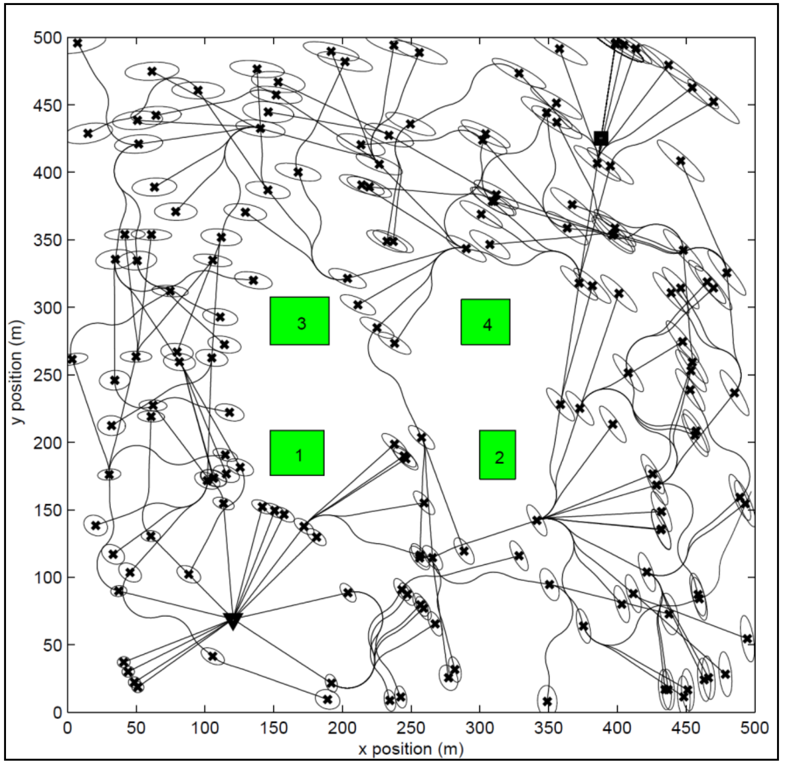
$$\begin{aligned} \dot{d} &= v_a \sin(\psi_r - \psi) \\ \dot{\tilde{\psi}} &= -\frac{k_1}{v_a}(\psi_r - \psi) - \frac{k_2}{v_a}\, d \end{aligned}$$



Predicted trajectory

Tree Expansion: Offline

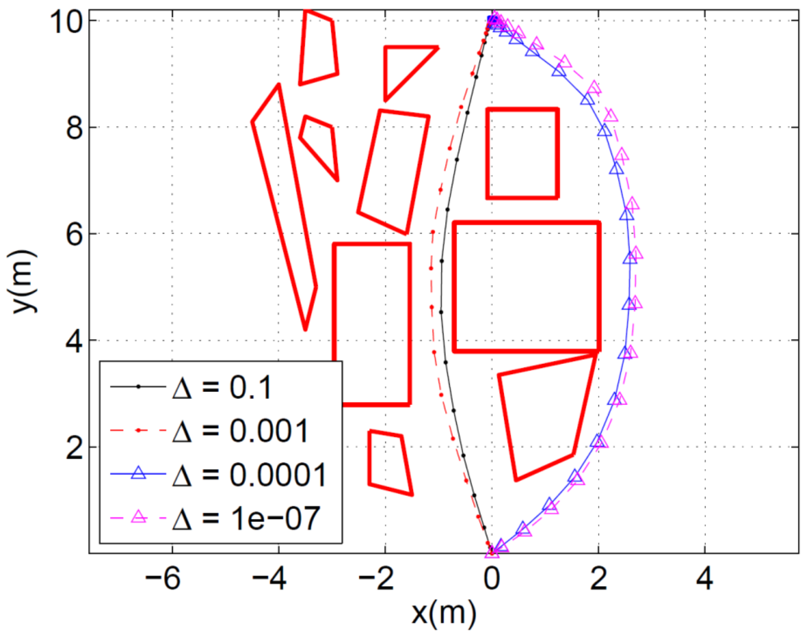


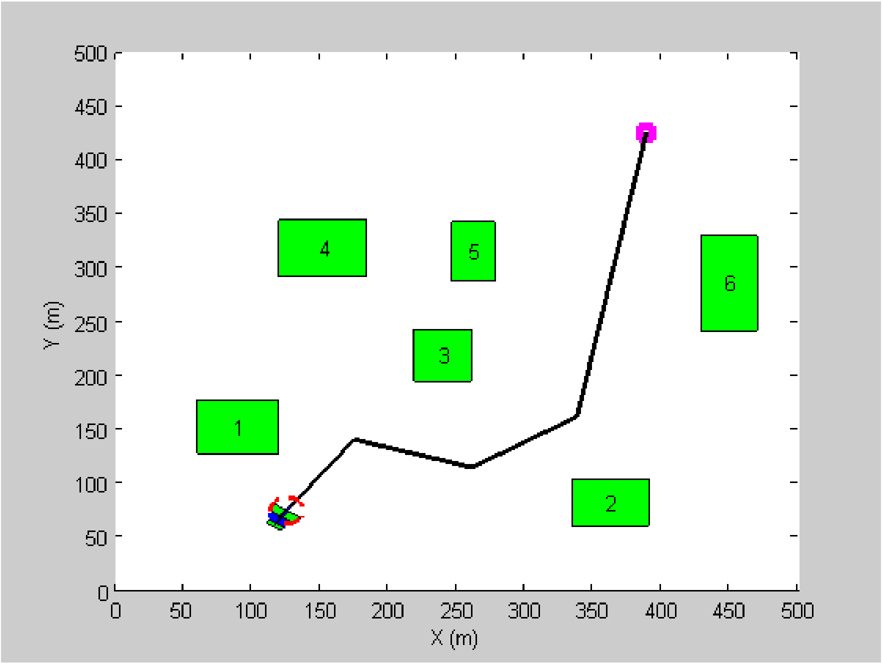


Pr(Δ) = 0.5

Pr(Δ) = 0.1

Numerical Results: Offline



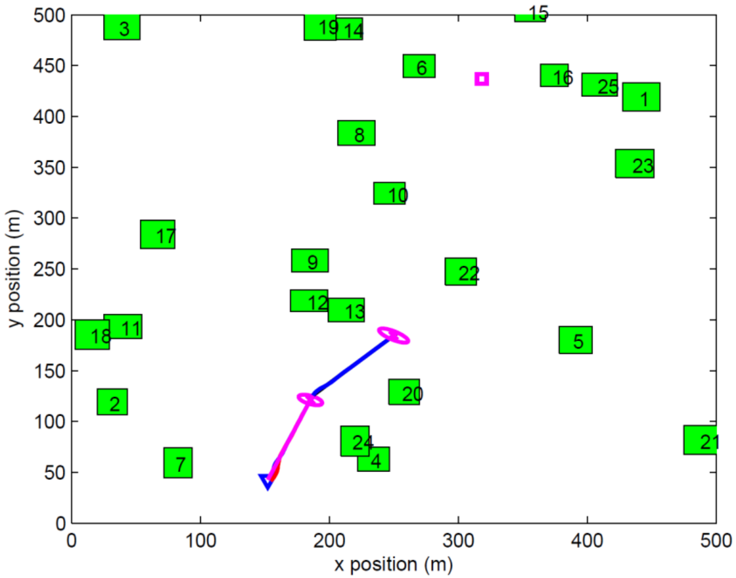


Paths with different Pr(Δ)

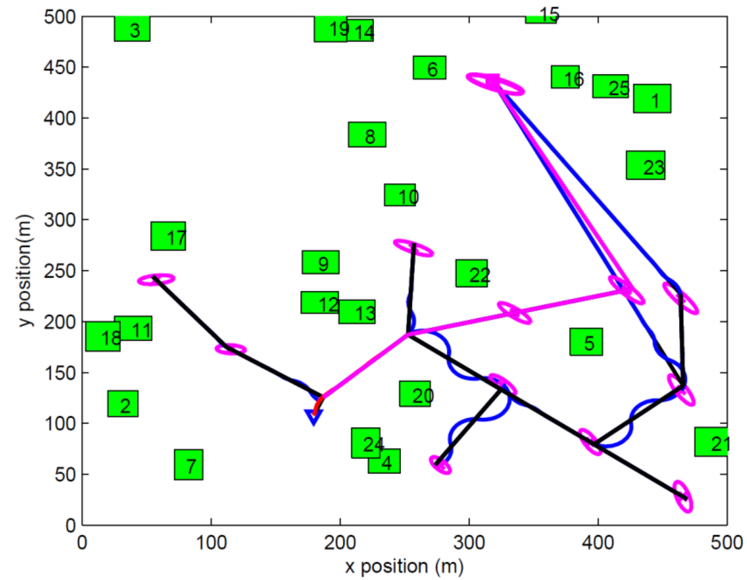
Online Implementation

- 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

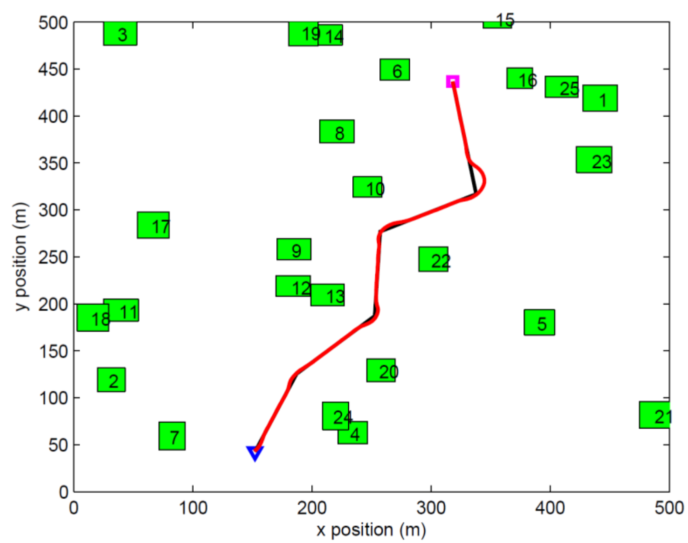
Numerical Results: Online



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

$$\begin{aligned} x_{t+1} &= f(x_t, u_t) + w_t \\ z_t &= h(x_t) + v_t \end{aligned} \xrightarrow{\text{Nominal system}} \begin{aligned} x^*_{t+1} &= f(x^*_t, u^*_t) \\ z^*_t &= h(x^*_t) \end{aligned}$$

Let $x^e_t \triangleq x_t - x^*_t$ be the error between the nominal and actual systems. The linearized error dynamics is given as

$$\begin{aligned} x^e_{t+1} &= A_t x^e_t + B_t u^e_t + w_t \\ z^e_t &= H_t x^e_t + v_t \end{aligned}$$

Compute *a priori* closed-loop distribution

$$\begin{aligned} \hat{x}_t &\triangleq \mathbf{E}[x_t] = \mathbf{E}[x^*_t] + \mathbf{E}[x^e_t] \\ \Sigma_{x_t} &\triangleq \mathbf{E}\left[(x_t - \mathbf{E}[x_t])(x_t - \mathbf{E}[x_t])^T\right] \end{aligned}$$

Closed-loop Distributions

The measurement step of Kalman Filter is given by

$$\boxed{\begin{aligned} x_{t+1|t+1} &= x_{t|t} + L_{t+1}\ (z_{t+1} - H_t\ x_{t+1|t}) \\ \Sigma_{t+1|t+1} &= (I - L_{t+1}\ H_t)\ A_t\ \Sigma_{t+1|t} \end{aligned}}$$

where

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

The augmented system

$$\boxed{\xi_{t+1} \triangleq \begin{bmatrix} x^e_{t+1} \\ x^e_{t+1|t+1} \end{bmatrix} = \begin{bmatrix} A_t & 0 \\ L_{t+1}H_tA_t & A_t - L_{t+1}H_tA_t \end{bmatrix} \xi_t + \begin{bmatrix} B_t \\ B_t \end{bmatrix} u_t + \begin{bmatrix} I & 0 \\ L_{t+1}H_t & L_{t+1} \end{bmatrix} \begin{bmatrix} w_t \\ v_{t+1} \end{bmatrix}}$$

Closed-loop Update: Most likely Measurements

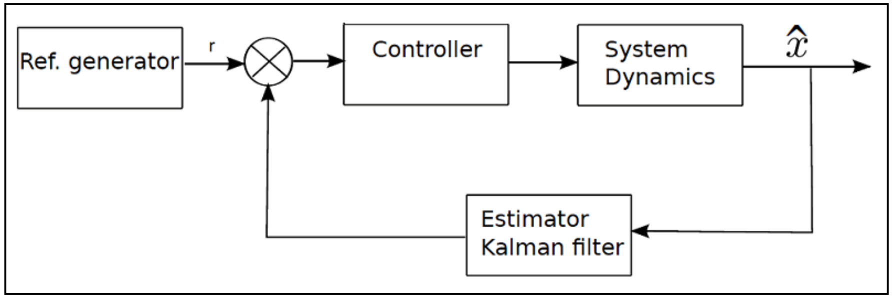
A priori distribution of the augmented system

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

Closed-loop belief update

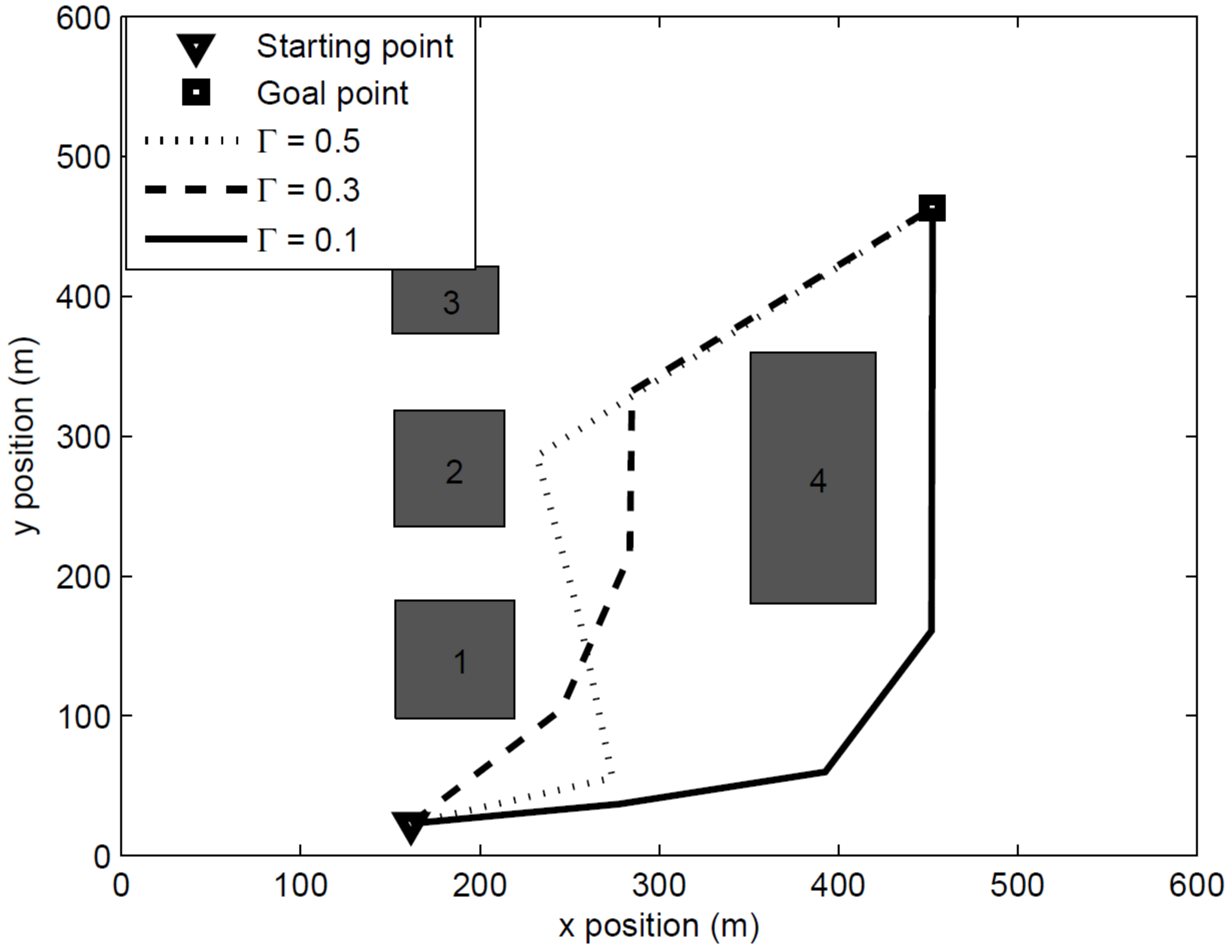
$$
\hat{x}_t^e = \mathbf{E}[\Lambda \xi_t] = \Lambda \hat{\xi}_t
$$

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

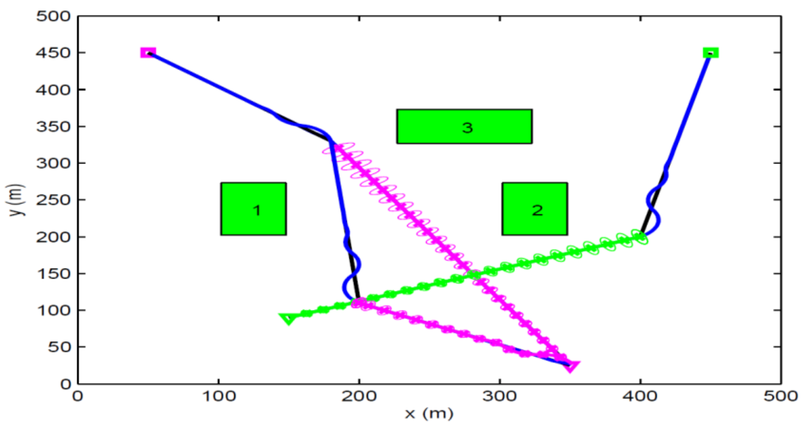


Closed-loop prediction (with most likely future measurements)

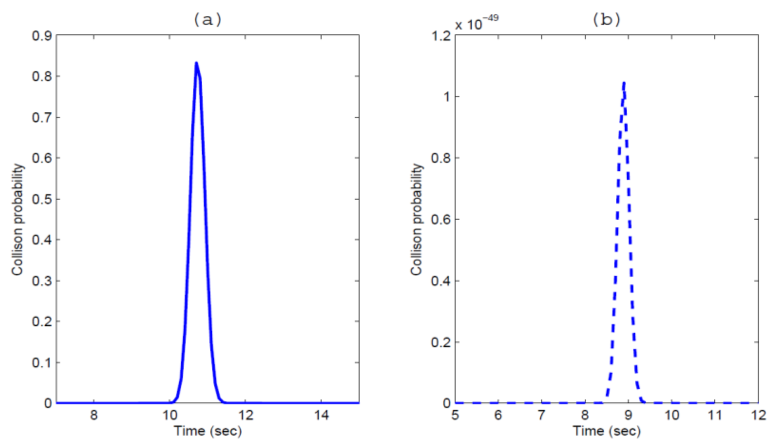
Numerical Results: A Single Agent System



A Two Agent System



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



(b) Probability of collision

Thank You