Incremental Sampling-based Algorithm for Risk-aware Planning under Motion Uncertainty

by

Wei Liu and Marcelo H. Ang Jr.

TaeHyoung Kim(김태형)

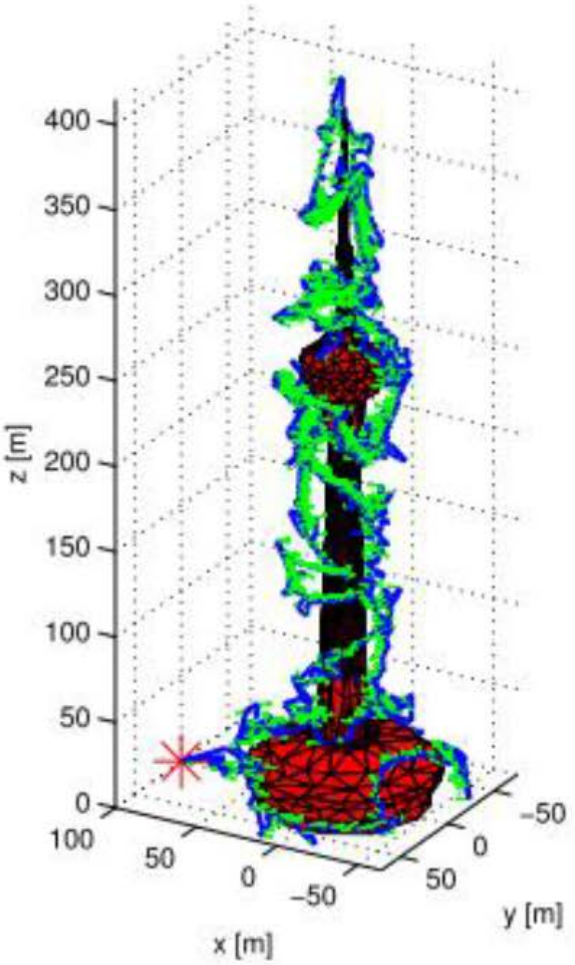
KAIST

Review

- **405m Tower**

N_{facets}	1701	$\angle$incidence	30°
FoV	$[120, 120]^\circ$	Mouting pitch	15°
$d_{\min}$	10m	$d_{\max}$	25m
$v_{\max}$	2m/s	$\dot{\psi}_{\max}$	0.5rad/s

Large scale structure to be inspected:
The 405m high Central Radio & TV Tower in Beijing. The mesh used to compute the path contains 1701 triangular facets. After a computation time of 92s the cost for the inspection is 2997.44s The red point denotes, start– and end–point of the inspection.



Incremental Sampling-based Algorithm for Risk-aware Planning under Motion Uncertainty

by

Wei Liu and Marcelo H. Ang Jr.

TaeHyoung Kim(김태형)

KAIST

Motivation

- **Need for uncertainty robust motion planner**

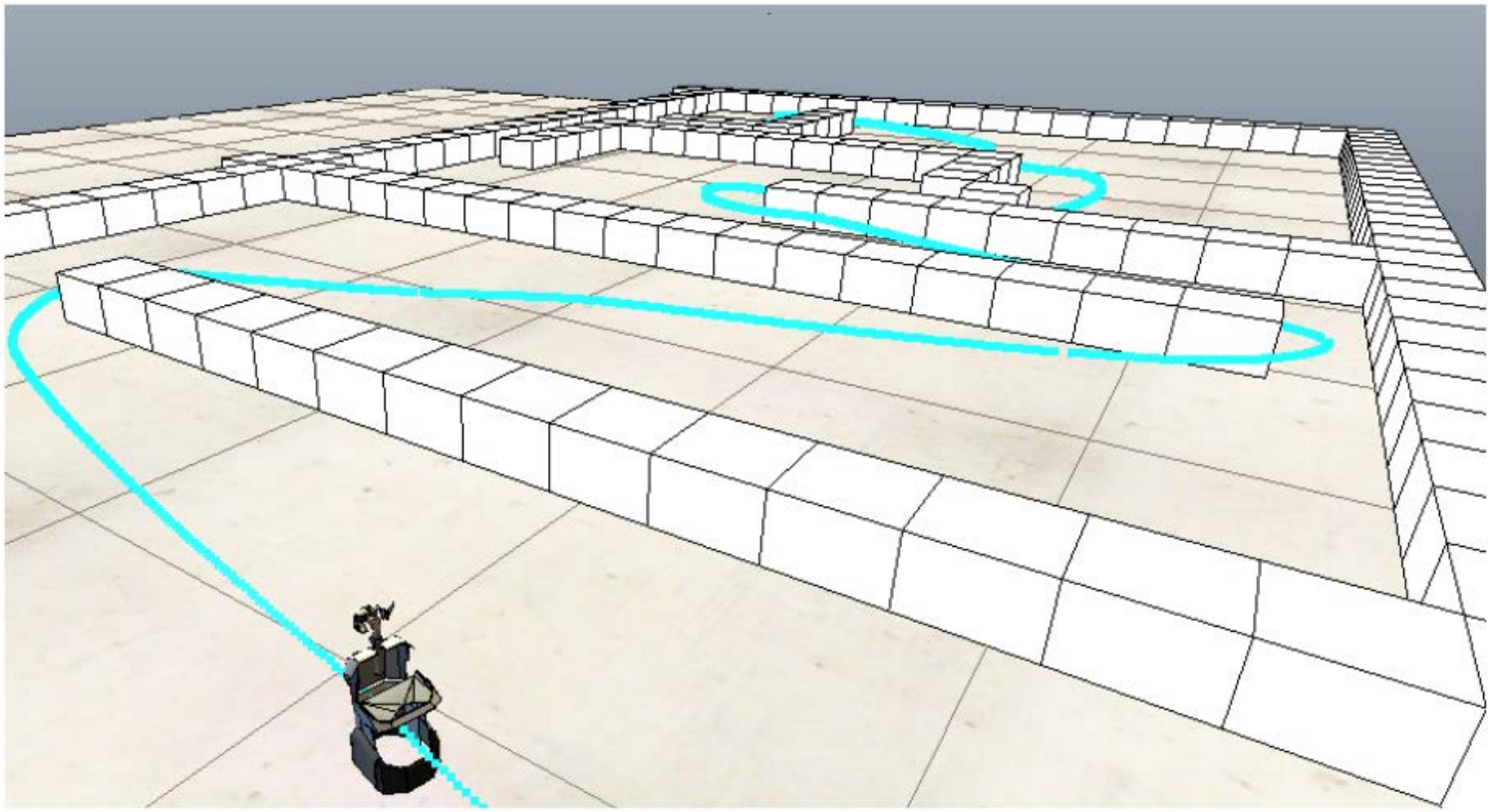


Image from, A Novel RRT Extend Function for Efficient and Smooth Mobile Robot Motion Planning

KAIST

Assumption

- **Robot use feedback controller to track a nominal trajectory.**
- **System can be well-linearized around given nominal path.**
- **Estimate of noise/state is modeled as Gaussian distribution.**
- **Obstacles are convex polyhedral.**

Problem Statement

- **Stochastic system model**

$$x_{t+1} = f(x_t, u_t, m_t)$$

$$m_t \sim N(0, M_t)\ , x_{init} \sim N(\hat{x}_0, \Sigma_{x_0})$$

- **Path**

$$\pi = (x_0, u_0, \dots, x_{T_{goal}}, u_{T_{goal}})$$

- **Object**

$$\min_{u_t} \sum_{t=0}^{T_{goal}} \phi(\hat{x}_t, \chi_{goal}, u_t) \qquad P(x_t \in \chi_F) \geq \delta, t : [0, T_{goal}]$$

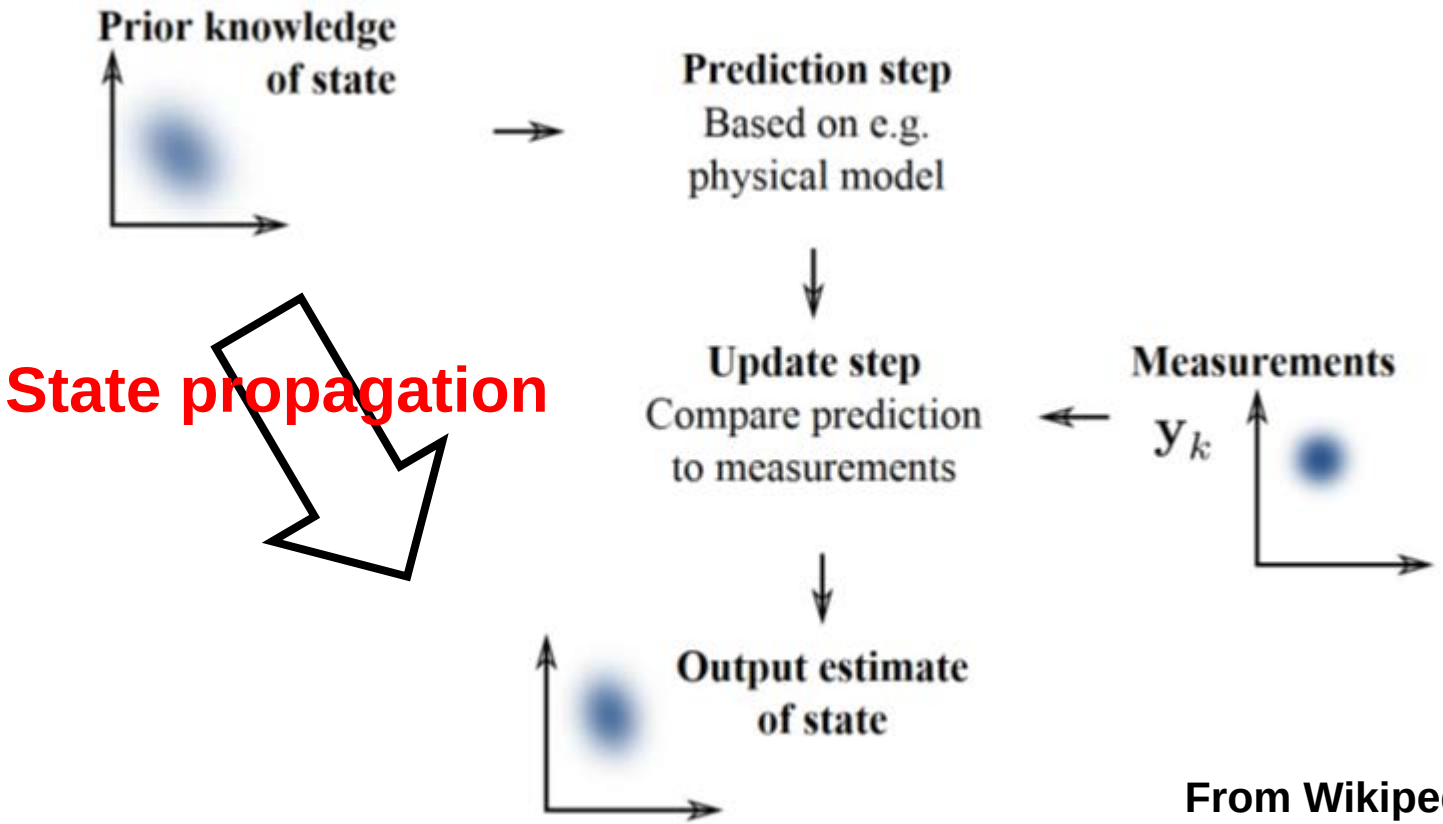
KAIST

Tools & Algorithm

Kalman Filtering

Kalman filtering - 1

- **Linear Quadratic Estimation(LQE)**
- **Keep estimate as Gaussian distribution**

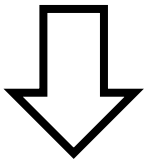


From Wikipedia

Kalman filtering - 2

- given nominal path , x_t^*, u_t^*
- Locally linearize around nominal path

$$x_{t+1} = f(x_t, u_t, m_t)$$



$$\bar{x}_t = A_t \bar{x}_{t-1} + B_t \bar{u}_{t-1} + V_t m_t$$

Where, $\bar{x}_t = \mathrm{x_t} - \mathrm{x_t^*}, \quad \bar{u}_t = u_t - u_t^*$

$$\bar{x}_t \sim N(\hat{\bar{x}}_t, \Sigma_t)$$

Kalman filtering - 3

- **Estimate of $\bar{x}_t$ from $\bar{x}_{t-1}$**

$$\hat{\bar{x}}_t = A_t \hat{\bar{x}}_{t-1} + B_t \bar{u}_{t-1}$$
$$= (A_t + B_t K_t) \hat{\bar{x}}_{t-1} \quad \Leftarrow$$

Feedback controller

$$\bar{u}_t = K_t \hat{\bar{x}}_t$$

$$\Sigma_t = A_t \Sigma_{t-1} A_t^T + V_t M_t V_t^T$$

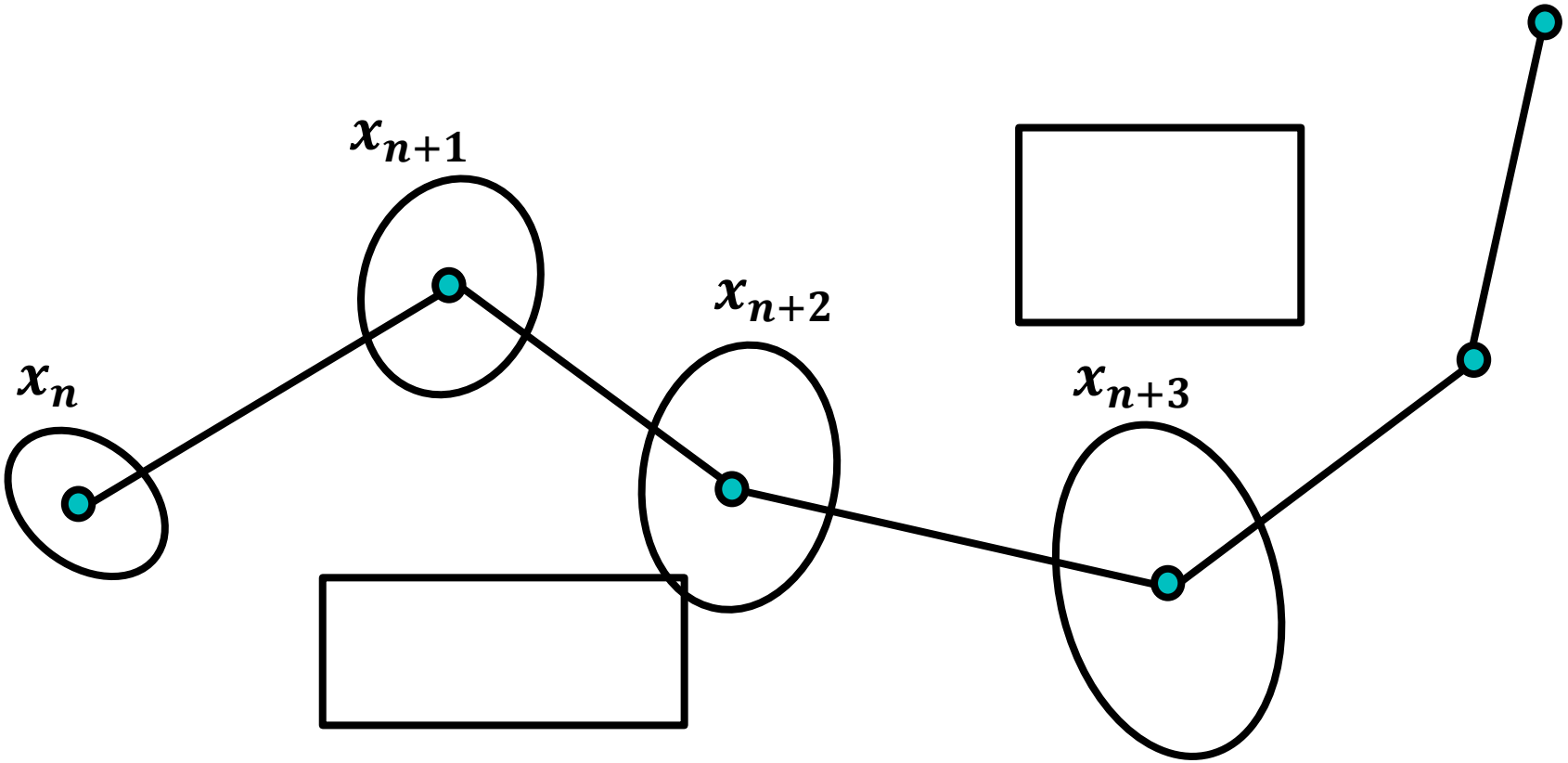
- **Estimate of state, $\tilde{x}_t$**

$$\bar{x}_t \sim N(\hat{\bar{x}}_t, \Sigma_t) \longrightarrow \tilde{x}_t \sim N(\hat{\bar{x}}_t + x_t^*, \Sigma_t)$$

Kalman filtering - 4

- **State propagation**

$$x_i \sim N\left(\hat{x}_i, \Sigma_{x_i}\right)$$

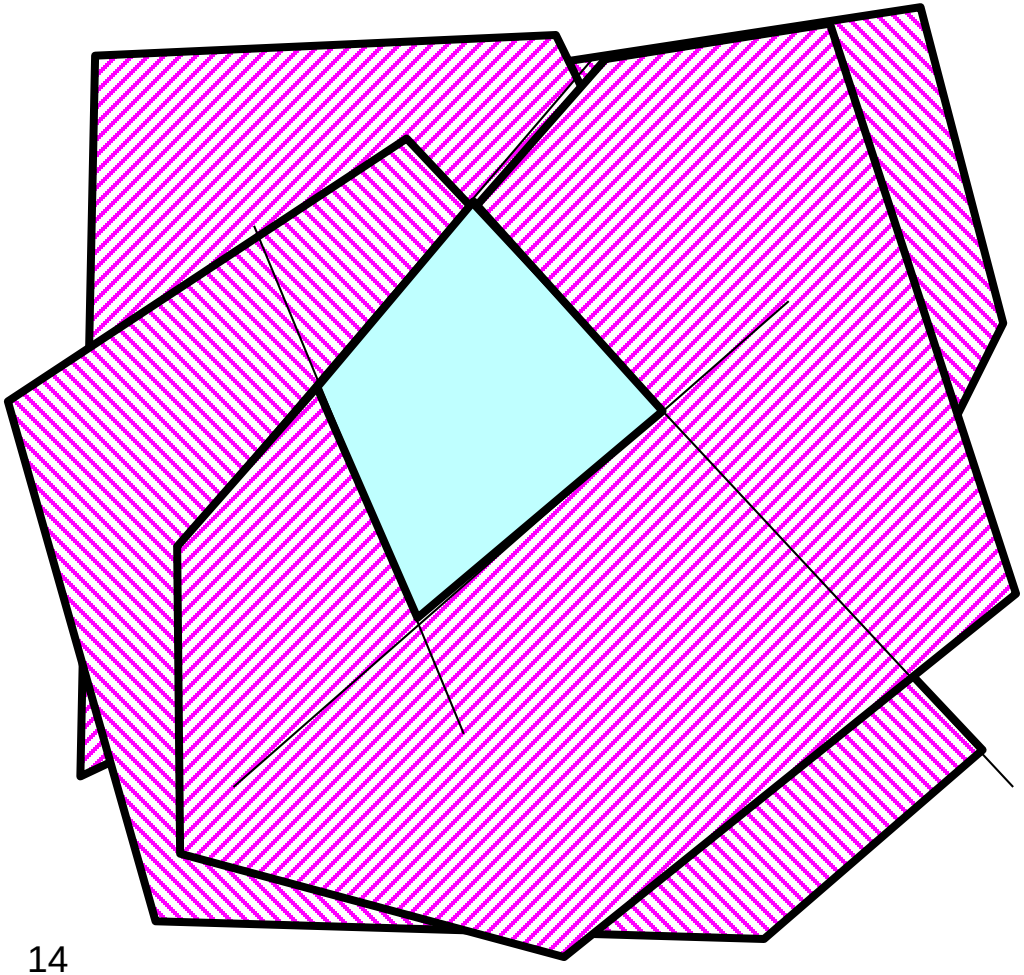


KAIST

Collision Probability

Collision probability

- **Convex polyhedral obstacle**

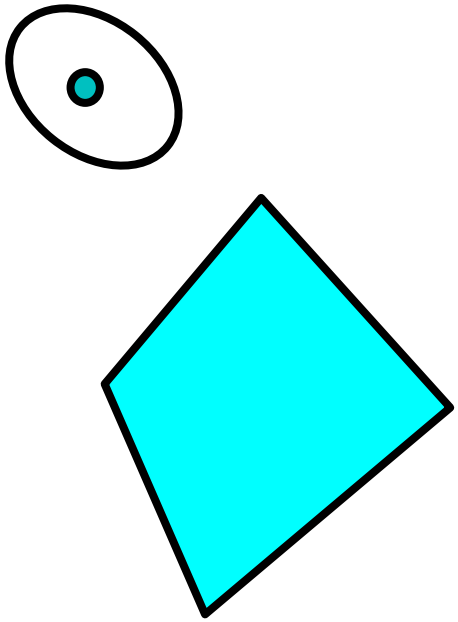


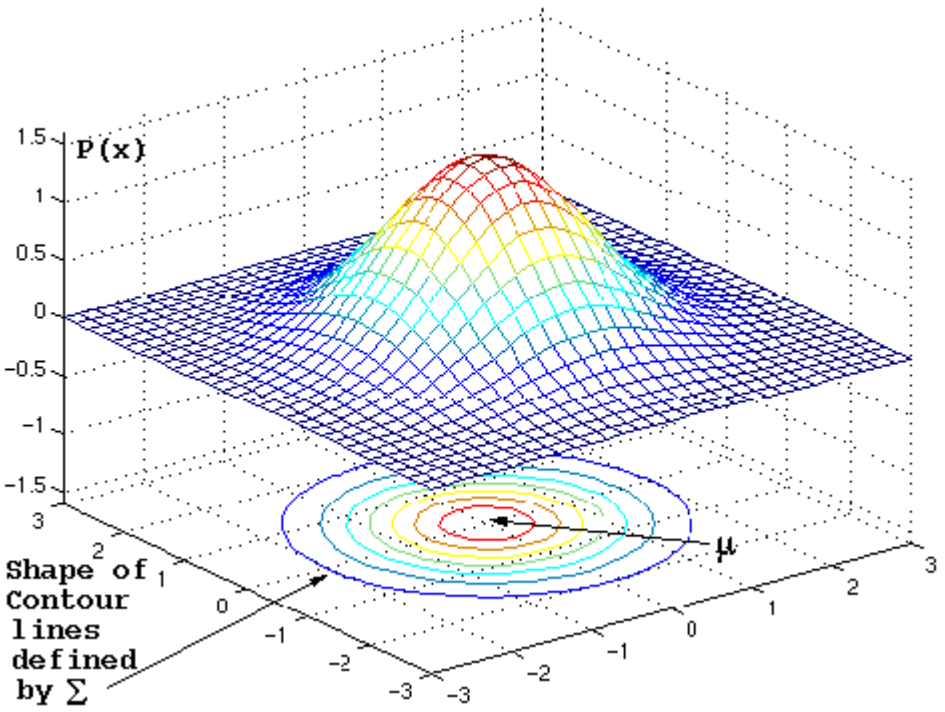
$$\bigwedge_{i=1,\ldots,L} a_i^T x_t \leq b_i$$

Collision probability

- **True collision probability**

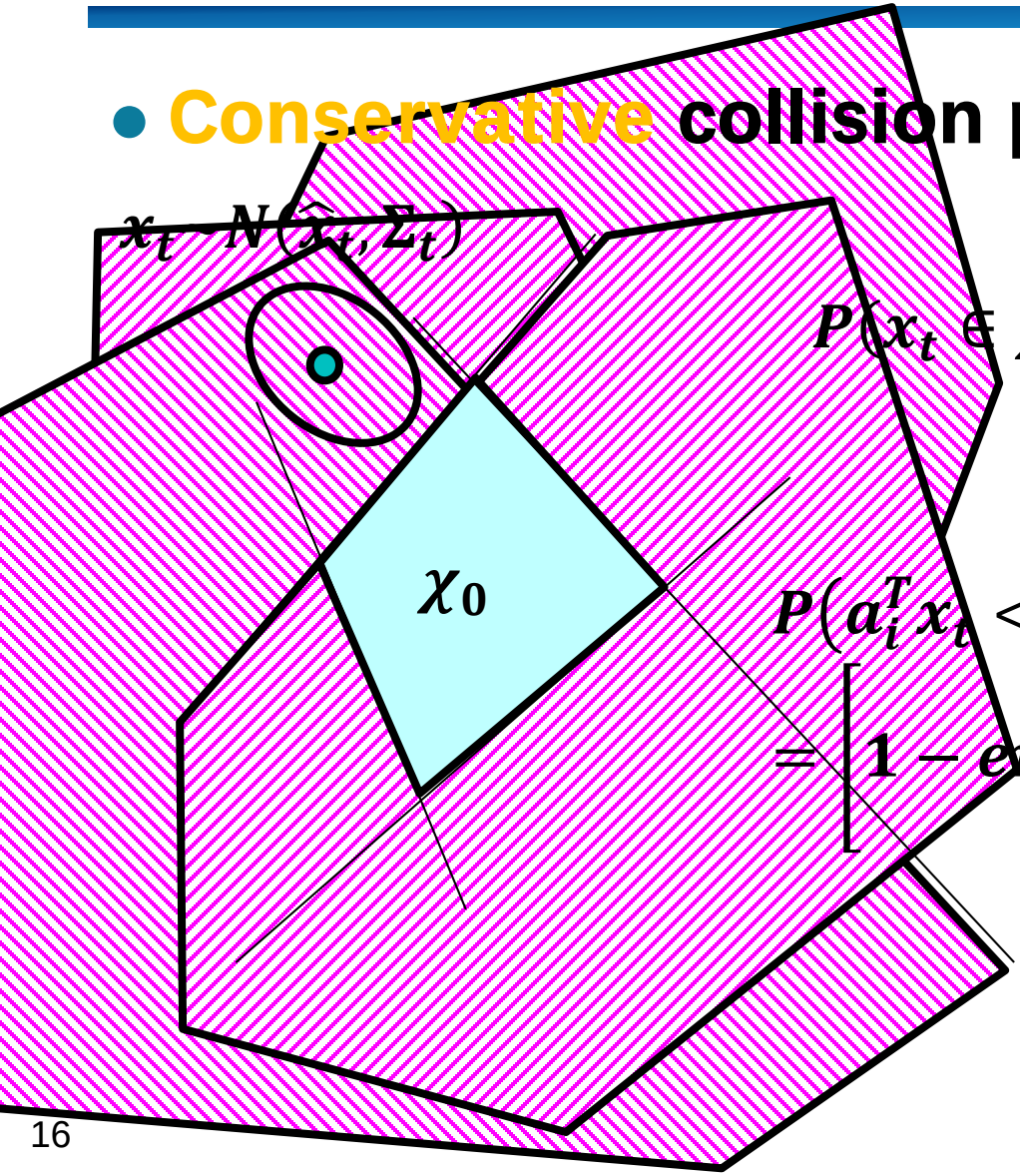
$$x_n \sim N(\hat{x}_n, \Sigma_n)$$





Collision probability

- **Conservative collision probability**



$$P(x_t \in \chi_0) \leq \min_{i \in \{1,\ldots,L\}} P(a_i^T x_t < b_i)$$

$$P(a_i^T x_t < b_i)$$

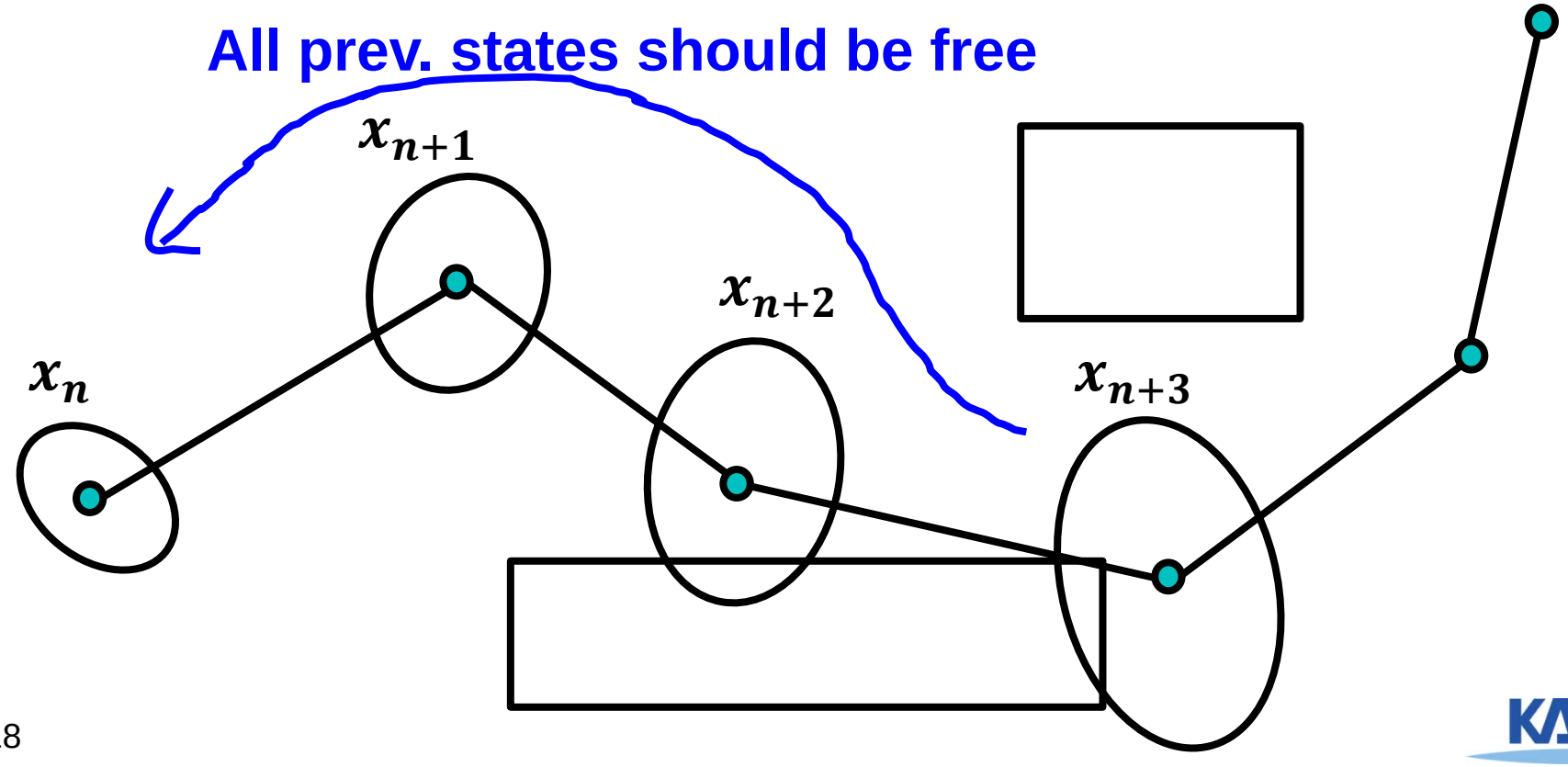
$$= \left[1 - erf\left((a_i^T \mu_t - b_i) / \sqrt{2 a_i^T \Sigma_t a_i}\right)\right] / 2$$

KAIST

Conditional State Propagation

State Propagation - unconditioned

- **Propagate states without considering obstacles**
- **It is not reasonable and over conservative.**



KAIST

Conditional State Propagation - 1

- **Only collision free part should propagate to next state**
- **Conditional state propagation**

$$P\left(x_t \in \chi_F \middle| \bigwedge_{i=0:t-1} x_i \in \chi_F\right)$$

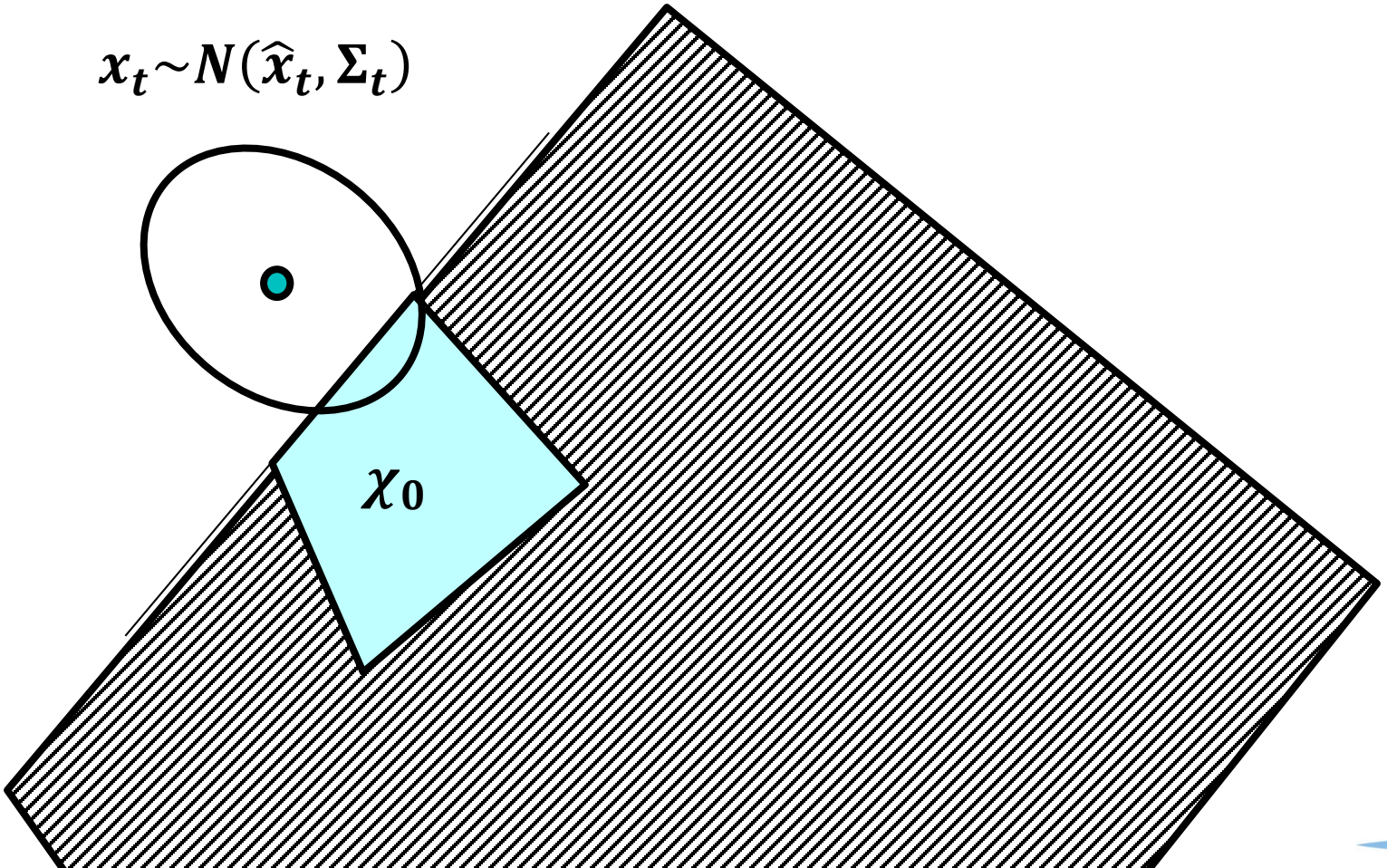
propagate

$$x_{t+1} \sim N(\hat{x}_{t+1}, \Sigma_{t+1})$$

KAIST

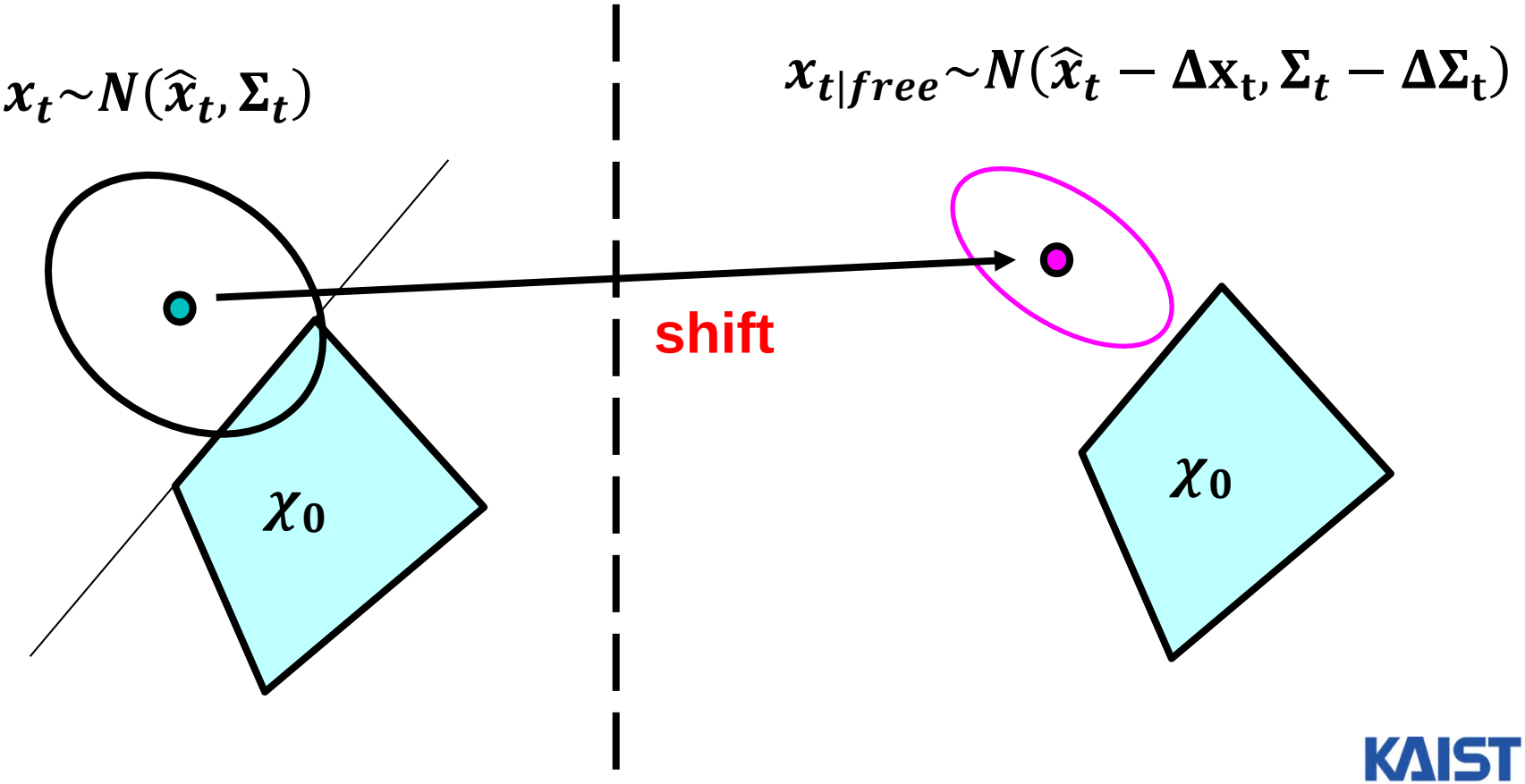
Conditional State Propagation - 2

- **Truncate using linear constraint**



Conditional State Propagation - 3

- **Result in shift of mean and covariance matrix**



KAIST

Conditional State Propagation

- **Truncation with linear constraint**

$$\Delta \mathbf{x_t} = -\frac{\lambda}{\sqrt{a^T \Sigma_t a}} (\Sigma_t \mathbf{a})$$

$$\Delta \Sigma_t = -\frac{\lambda^2 - \beta\lambda}{a^T \Sigma_t a} (\Sigma_t \mathbf{a})(\Sigma_t \mathbf{a})^T$$

Where,

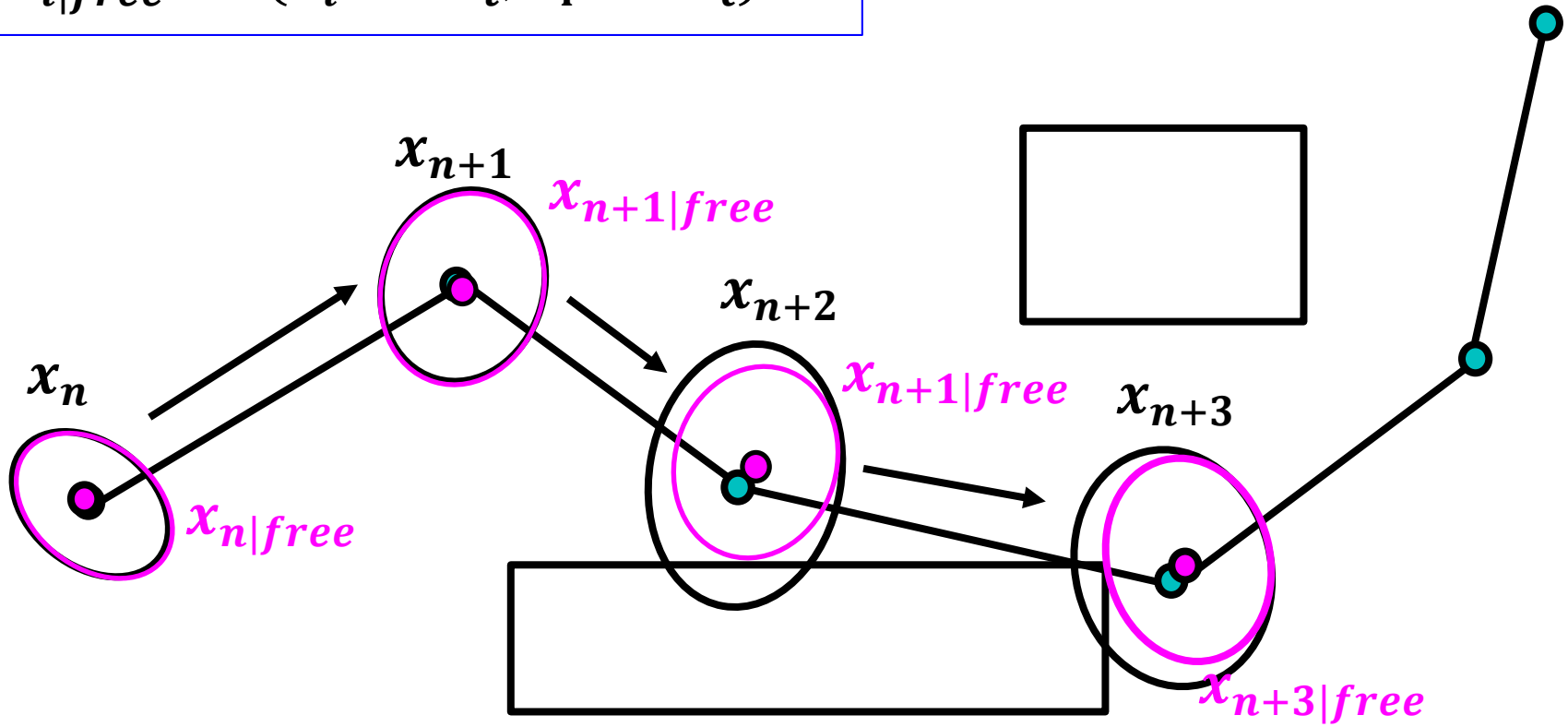
$$\beta = \frac{b - a^T}{\sqrt{a^T \Sigma_t \mathbf{a}}}, \qquad \lambda = \frac{pdf(\beta)}{1 - cdf(\beta)}$$

Degree of truncation

Conditional State Propagation

$$x_i \sim N(\hat{x}_i, \Sigma_i) \longrightarrow P(x_i \in \chi_F)$$

$$x_{i|free} \sim N(\hat{x}_i - \Delta x_t, \Sigma_i - \Delta\Sigma_t)$$



KAIST

Integrate with RRT*

Integrate in RRT* framework

- **Kalman filtering**
  : estimate / propagate state

- **Approximate collision probability**
  : conservatively & easily

- **Conditional state propagation**
  : state dependency

+ RRT* Framework

KAIST

Integrate in RRT* framework

- **RRT***
  - **Refine the connection with re-wiring operation**

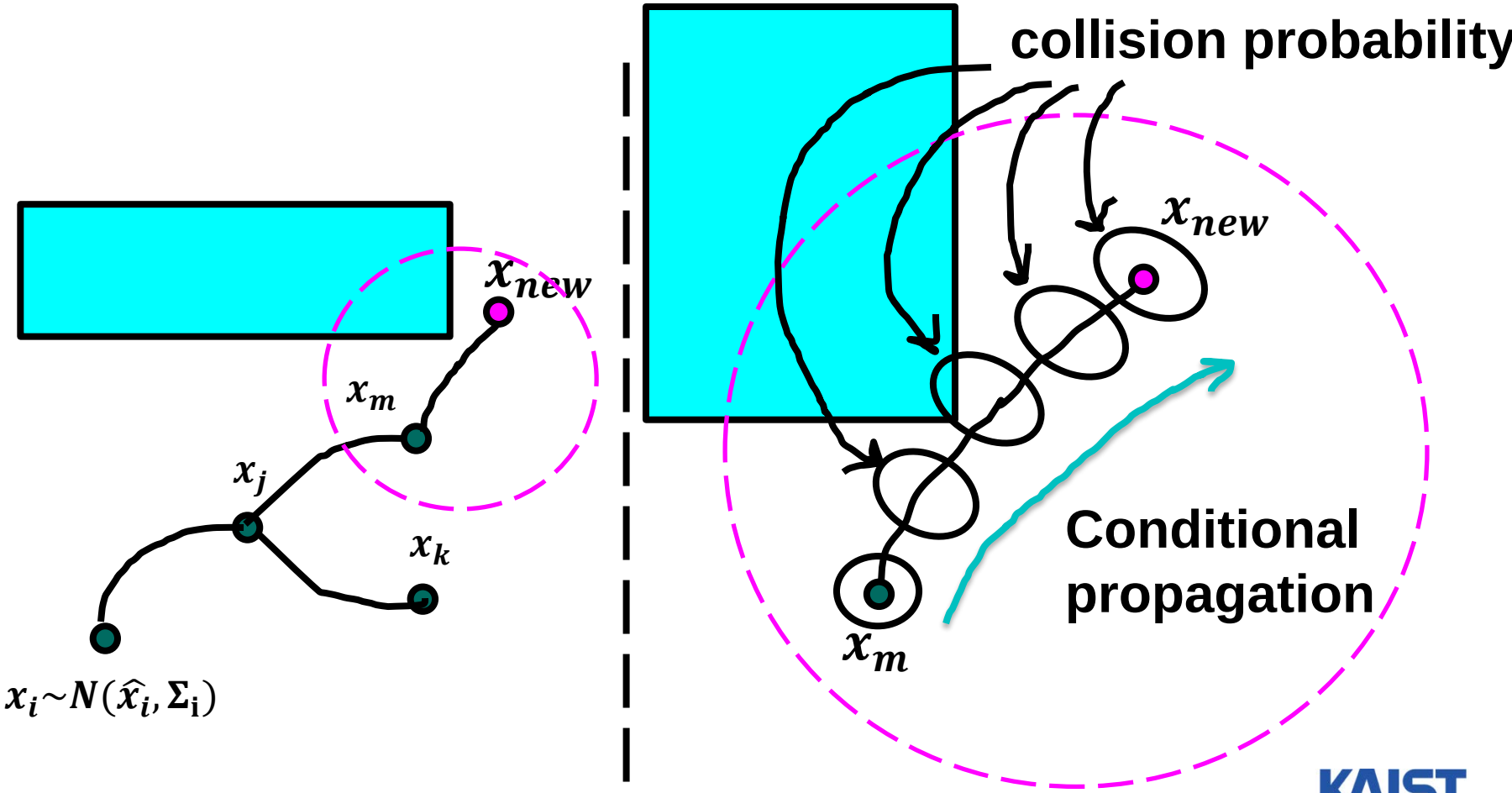
  +

  - **Evaluate collision probability of each states**
  - **Only allows nodes that satisfy collision constraints**

KAIST

Integrate in RRT* framework

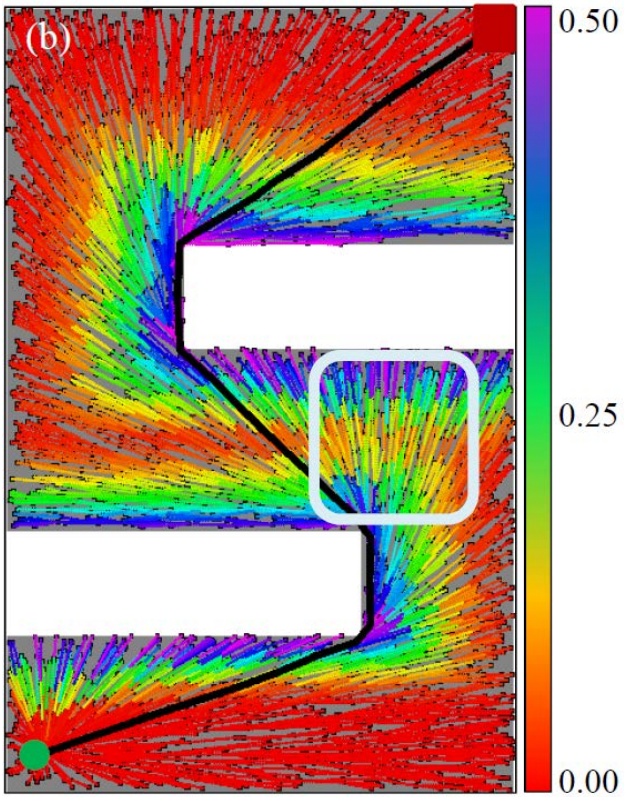
- **Example – Tree Expansion**



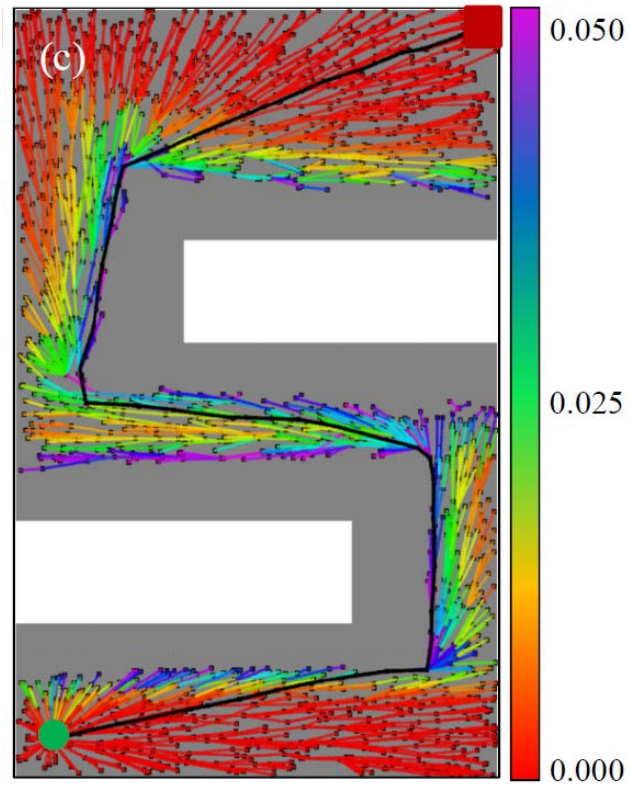
Simulation

Result

- **RRT* vs CC-RRT*-D**



RRT*



CC-RRT*-D

KAIST

Summary & Limitation

Summary

- **CC-RRT*-D**

A) Conditional state propagation & collision probability

→ **probabilistically feasible path**

B) RRT* framework

→ **asymptomatically optimized path**

KAIST

Summary & Limitation

- **Limitation**
  - **Kalman filtering without measurement update**
  - **Only applies to well linearizable system**
  - **Convex polyhedral obstacle**

Q&A