Robotic Navigation and Exploration

Week 4: SLAM Back-end (I)

Min-Chun Hu anitahu@cs.nthu.edu.tw
CS, NTHU

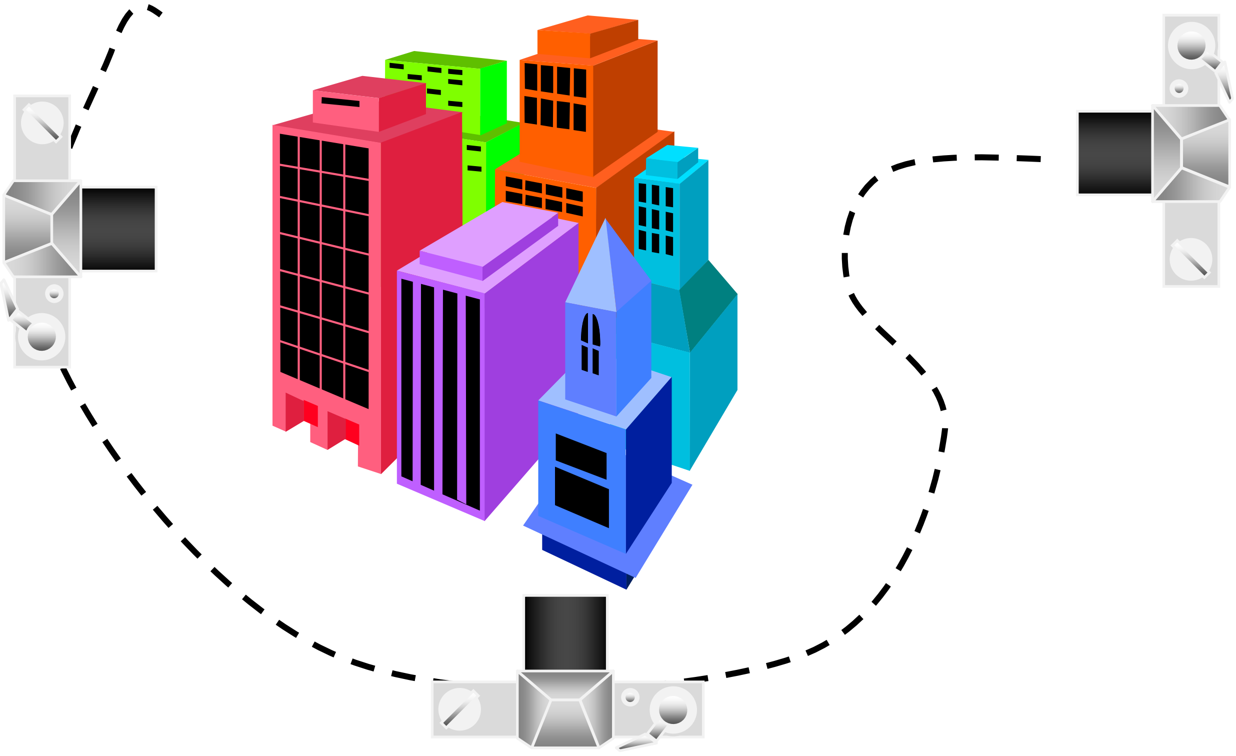
Outline

- State Estimation and SLAM Problem
- SLAM Back-end (Error Compensation)
  - Filter-based Methods
    - Probability Theory and Bayes Filter
    - Kalman Filter (KF) / Extended Kalman Filter (EKF)
      - EKF-SLAM
    - Particle Filter
      - Fast-SLAM
  - Graph-based Methods
    - Pose Graph and Least-square Optimization
    - Gauss-Newton and Levenberg-Marquardt Algorithm
    - Sparse Matrix for Optimization

SLAM Problem

定位与建图

没有先验知识

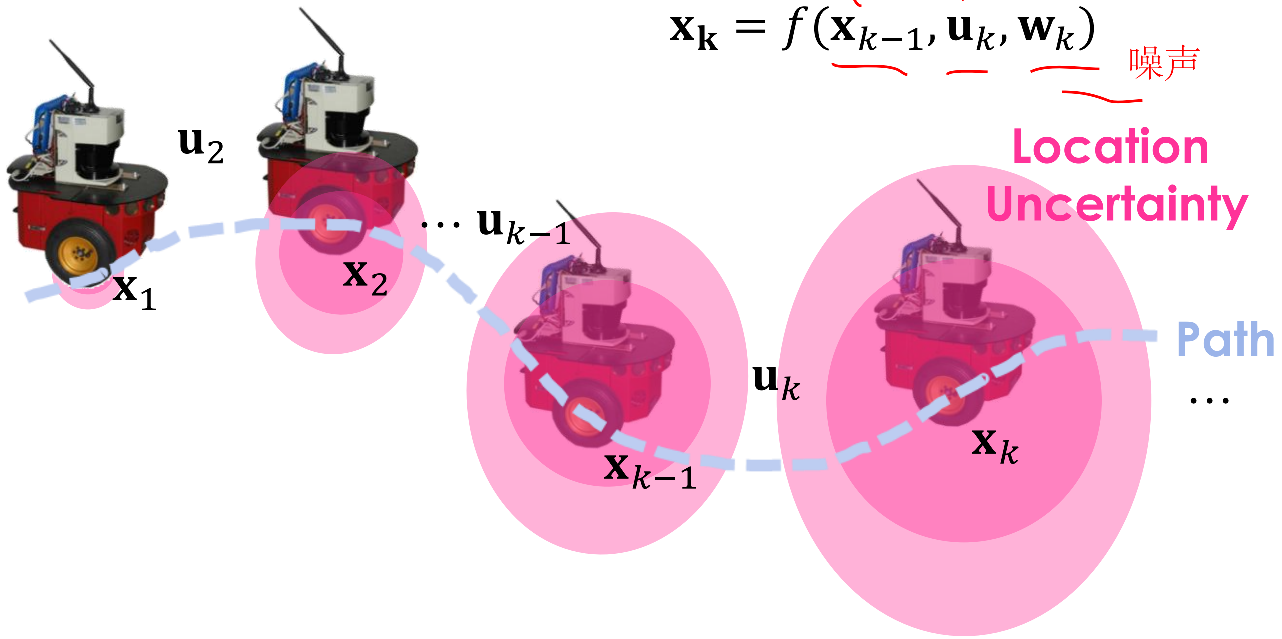


SLAM Problem

前一个位置　　运动控制

$$\mathbf{x_k} = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)$$

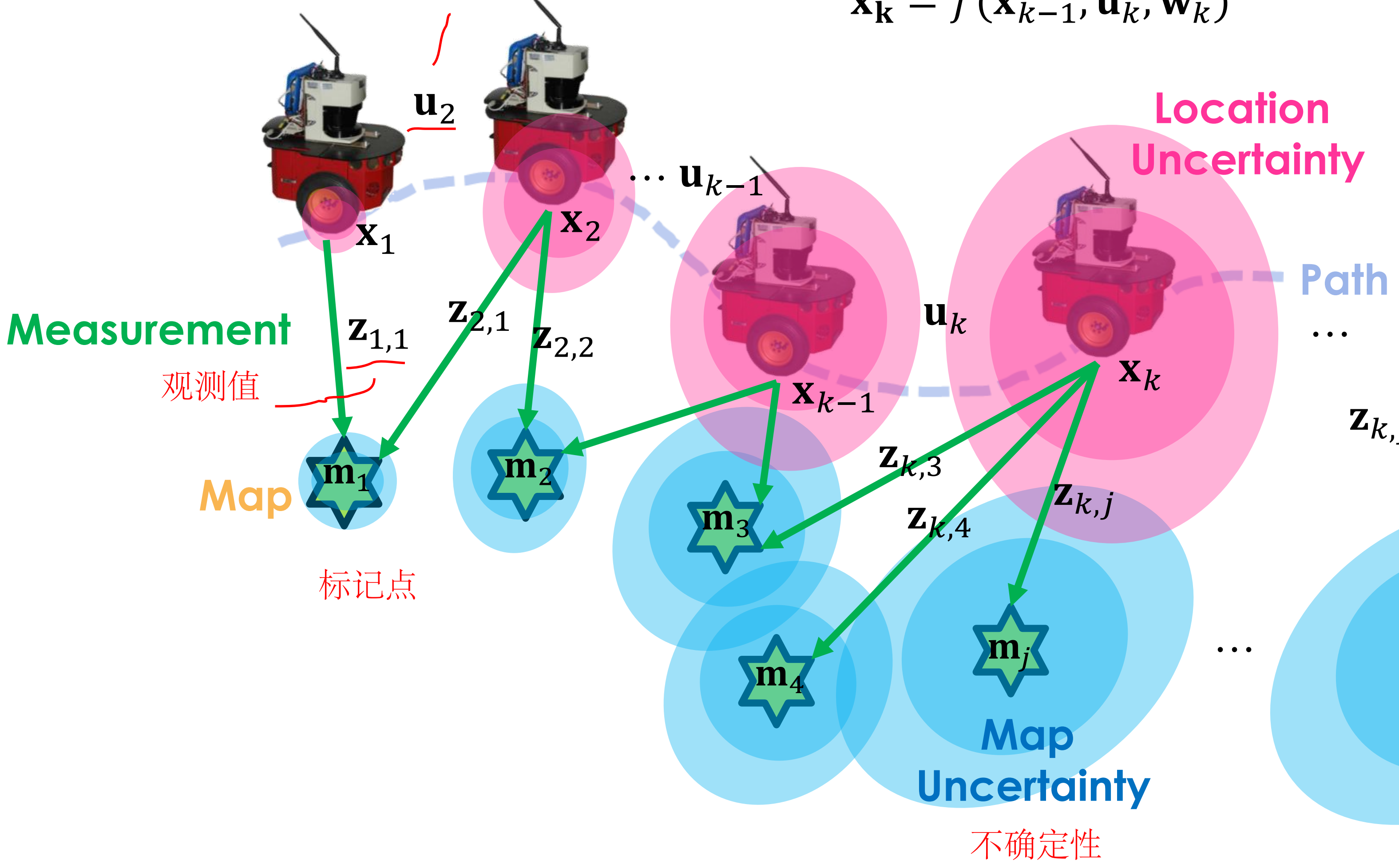
噪声



SLAM Problem

同时达到定位与建图

$$\mathbf{x_k} = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)$$



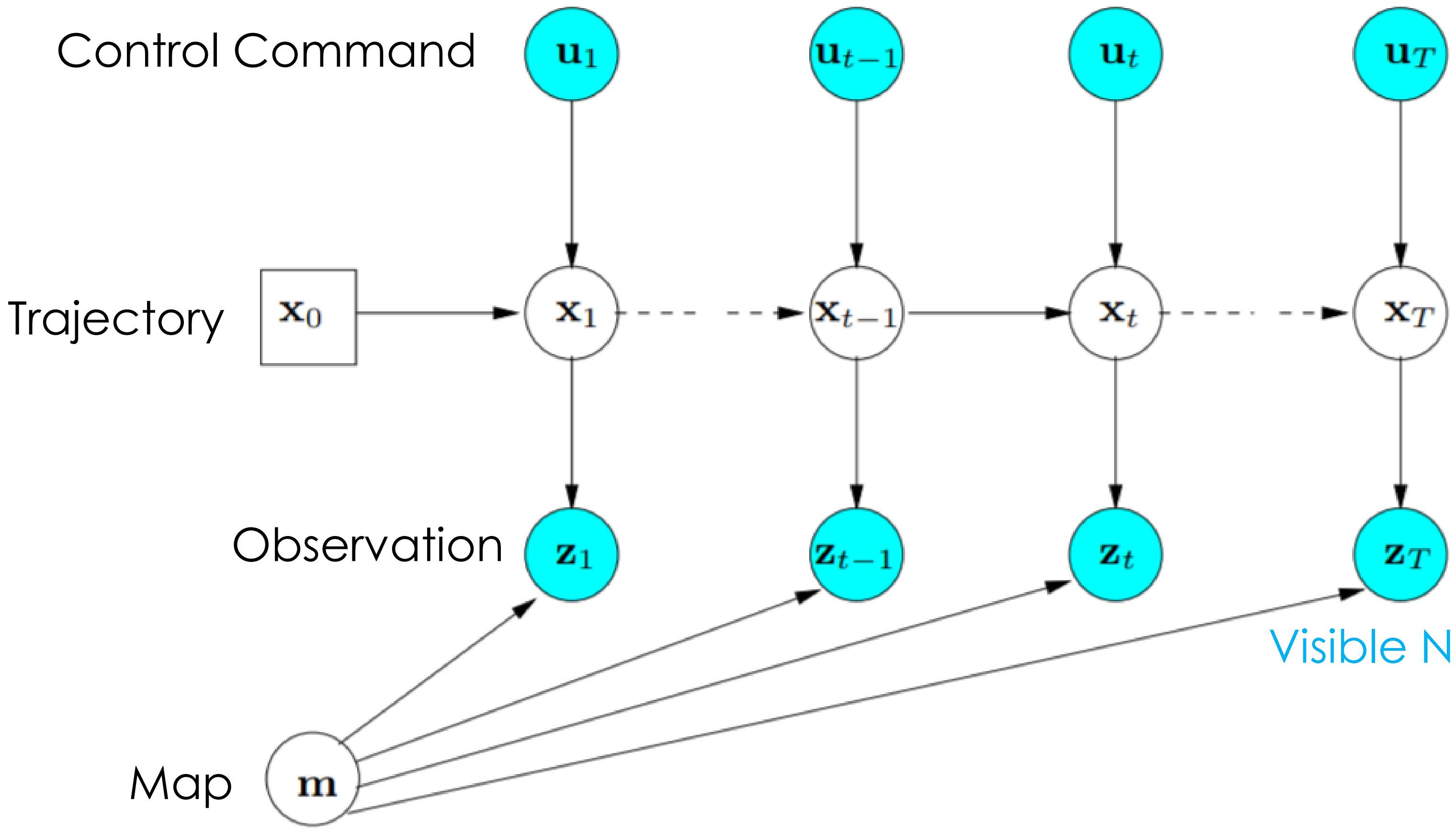
$$\mathbf{z}_{k,j} = h(\mathbf{m}_j, \mathbf{x}_k, \mathbf{v}_{k,j})$$ 观测噪声

机率图模型

Probability Graphical Model for SLAM Problem

每个随机变数是为一个note　有edge相连表示彼此有关联性，方向表示因果关系

Visible Node



Visible Node

运动方程式

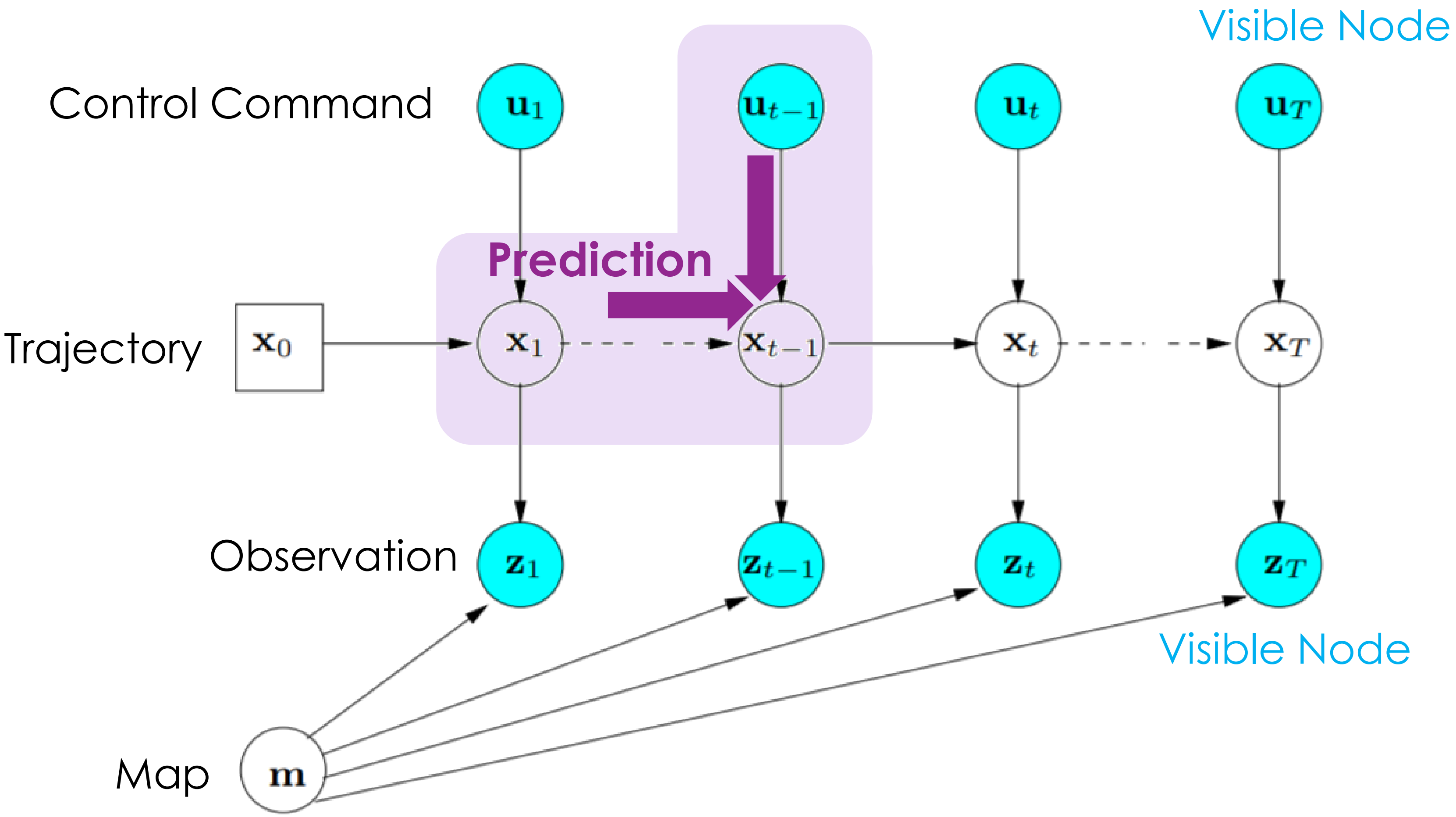
$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{w}_t)$$

观测方程式

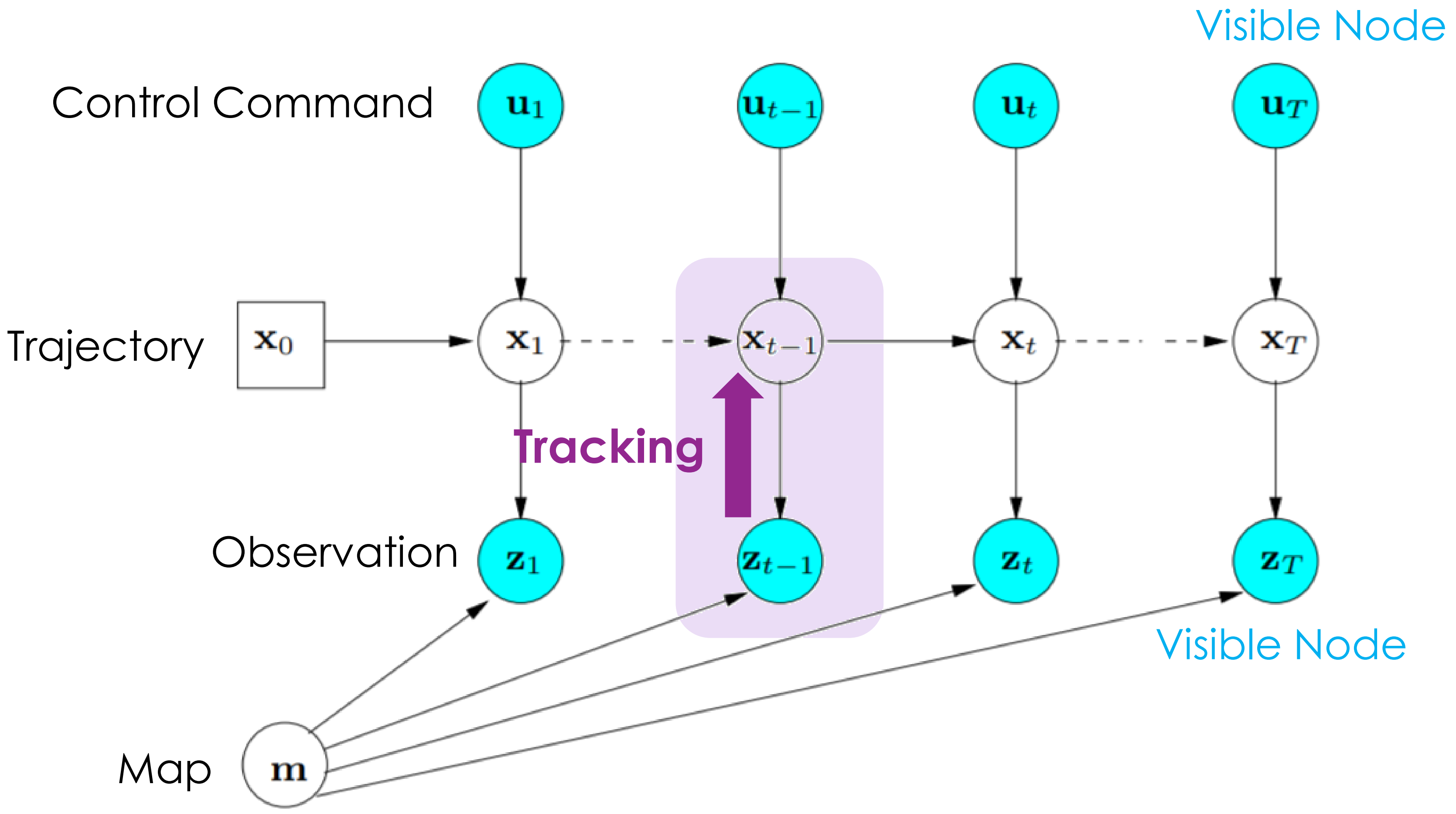
$$\mathbf{z}_t = h(\mathbf{m}, \mathbf{x}_t, \mathbf{v}_t)$$

在图中未呈现

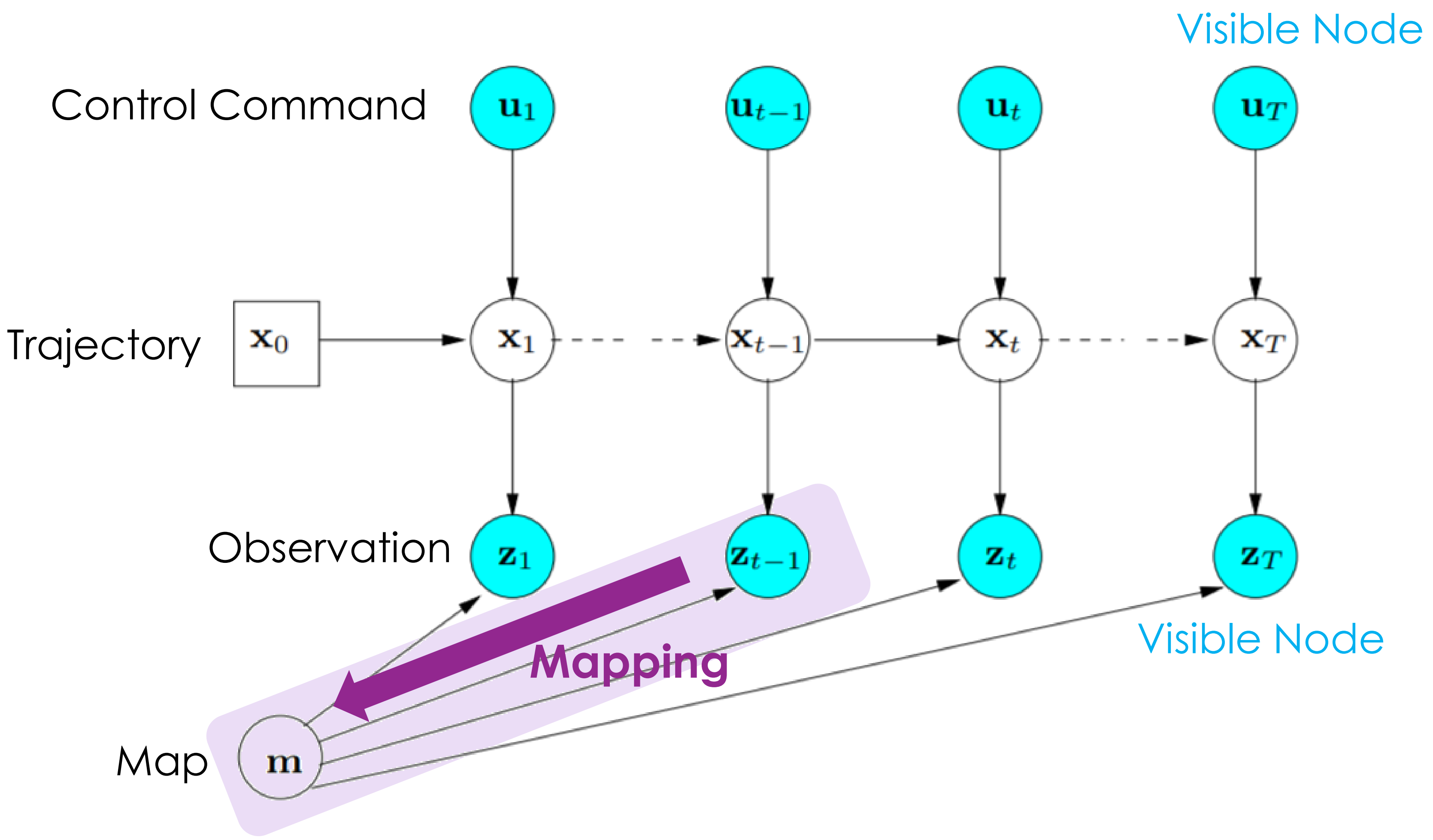
Probability Graphical Model for SLAM Problem



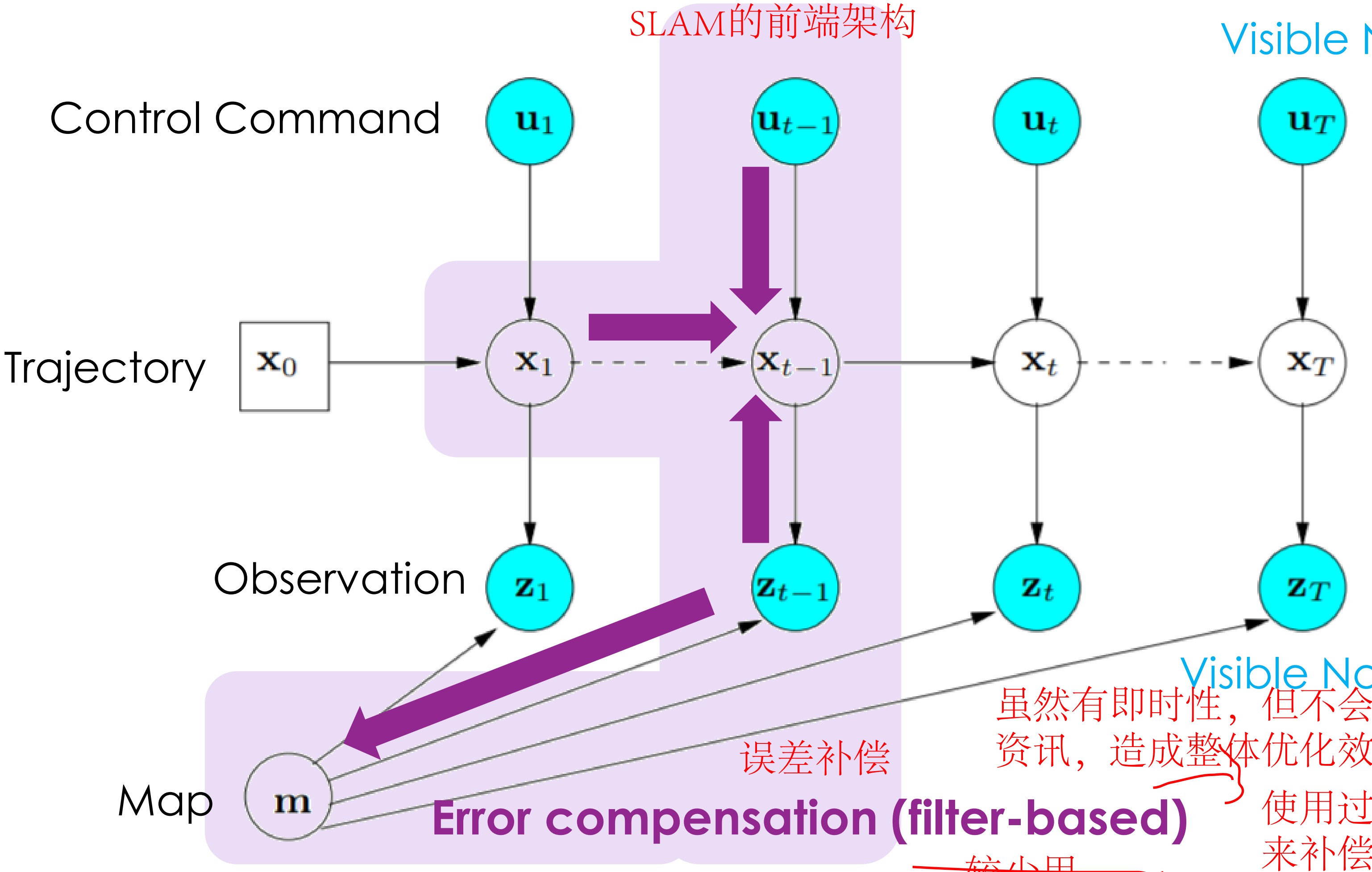
Probability Graphical Model for SLAM Problem

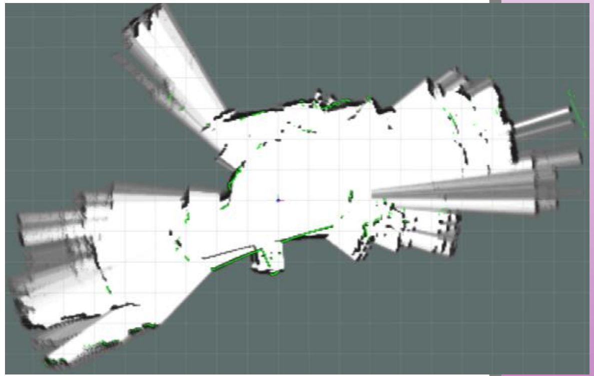


Probability Graphical Model for SLAM Problem



Probability Graphical Model for SLAM Problem





Visible Node

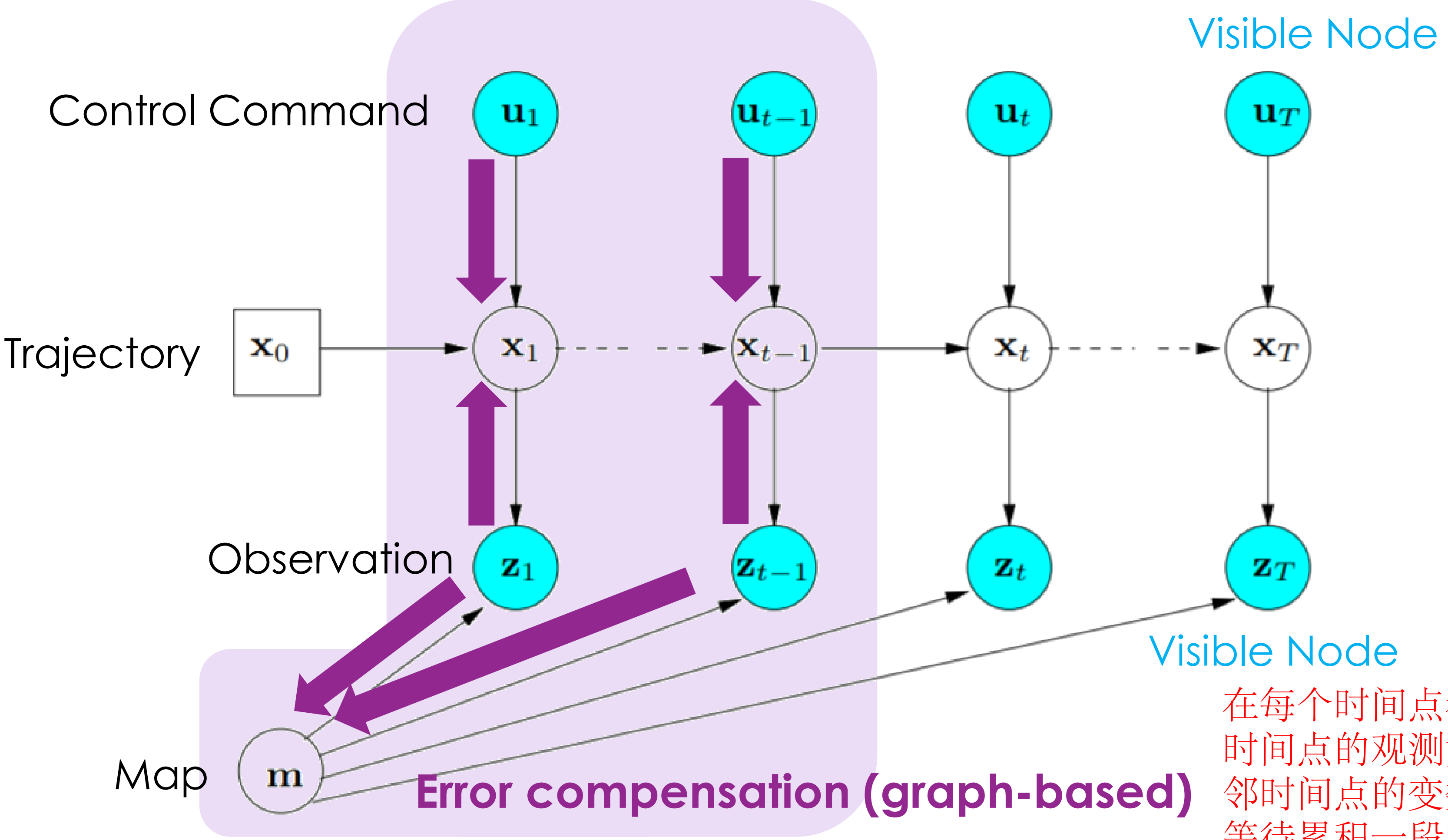
虽然有即时性，但不会在回头去解释过去的量测资讯，造成整体优化效果不佳

使用过去邻近时间点的最佳估测来补偿现在时间点的估测

Error compensation (filter-based)

~~较少用~~

Probability Graphical Model for SLAM Problem



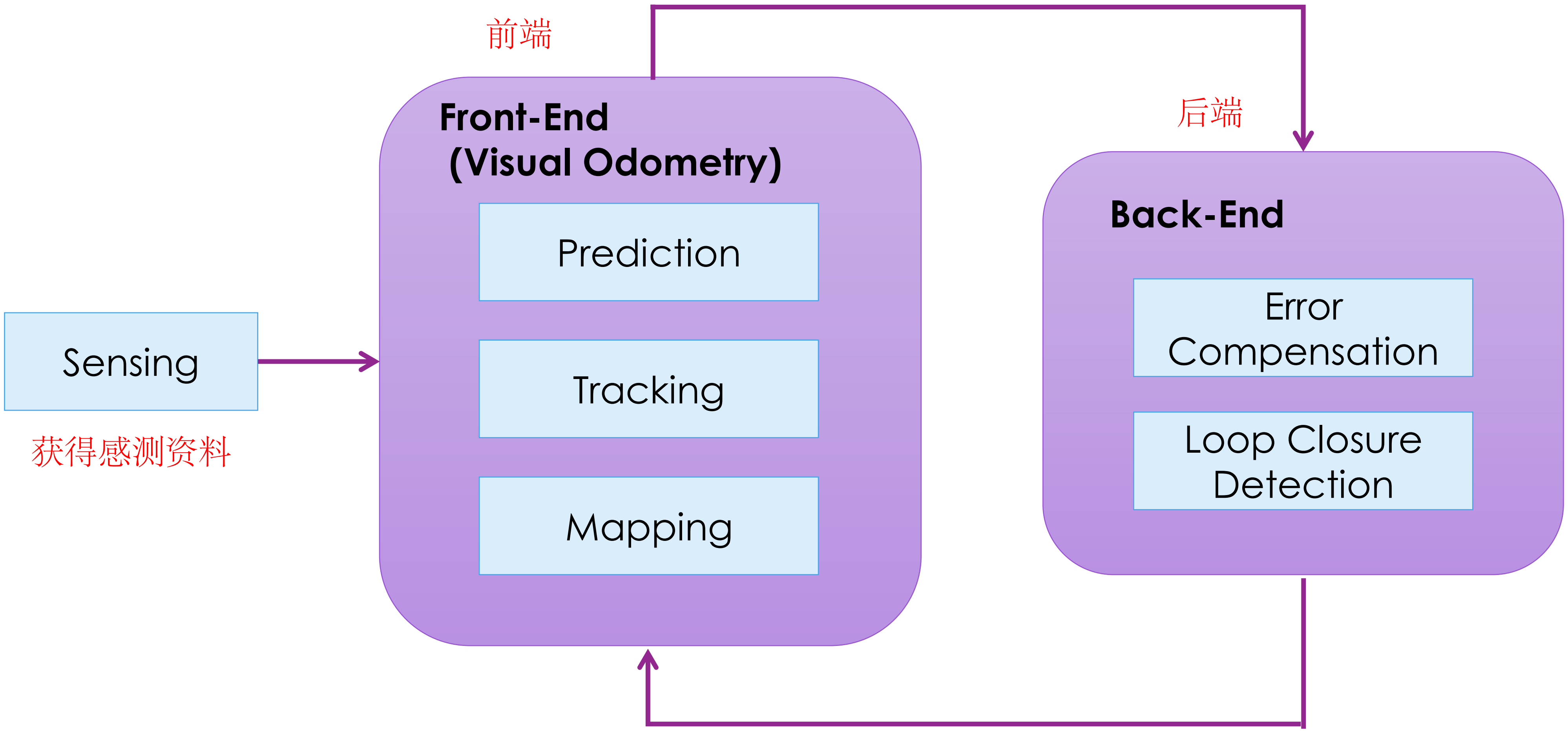
在每个时间点都使用了过去多个时间点的观测资讯，针对多个相邻时间点的变数进行优化。需要等待累积一段时间

Error Compensation Methods

- Filter-based
  - Small Computation
  - On-line Optimization

- Graph-based
  - Large computation
  - High Accuracy
  - Off-line Optimization

SLAM Architecture

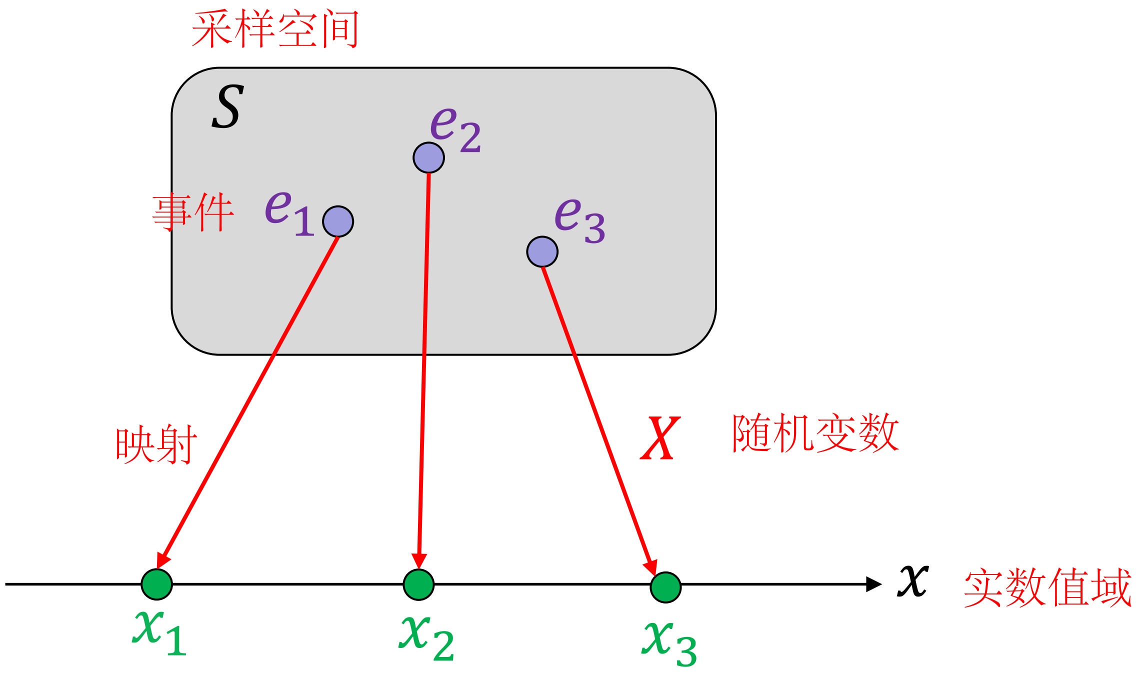


Outline

- State Estimation and SLAM Problem
- SLAM Back-end (Error Compensation)
  - Filter-based Methods
    - Probability Theory and Bayes Filter
    - Kalman Filter (KF) / Extended Kalman Filter (EKF)
      - EKF-SLAM
    - Particle Filter
      - Fast-SLAM
  - Graph-based Methods
    - Pose Graph and Least-square Optimization
    - Gauss-Newton and Levenberg-Marquardt Algorithm
    - Sparse Matrix for Optimization

Random Variable 随机变数

- A random variable is defined as a function that maps the observation results of unpredictable processes to numerical quantities
- Definition:
  - X:Random Variable
  - S:Sample Space
  - e:event ($e \in S$)
  - $X(e)=x$ ($x \in R$)

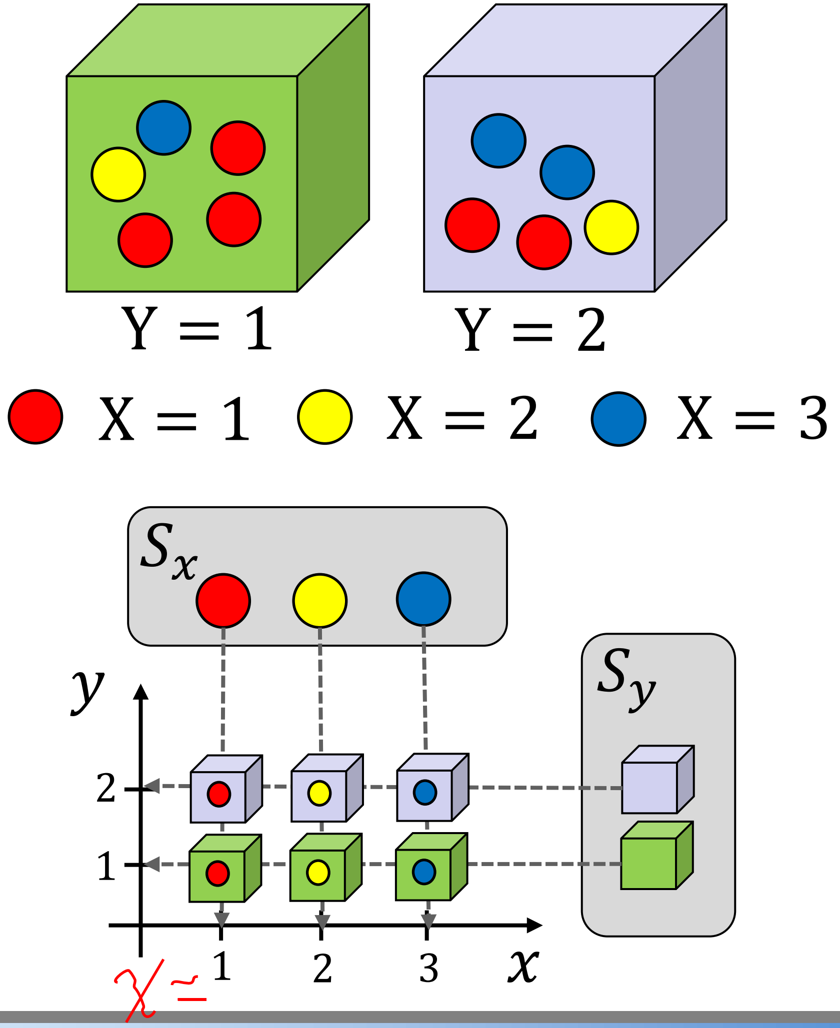


Example of Random Variable

$Two\ Random\ Variable: X, Y$

X: The id of the ball
$X = 1, if\ choose\ the\ \textbf{red}\ ball$
$X = 2, if\ choose\ the\ \textbf{yellow}\ ball$
$X = 3, if\ choose\ the\ \textbf{blue}\ ball$

Y: The id of the box
$Y = 1, if\ choose\ the\ \textbf{green}\ box$
$Y = 2, if\ choose\ the\ \textbf{purple}\ box$



Different Types of Probability

- Joint Probability 两个随机变数的共同机率分布

  联合概率

  $$P(X,Y)$$

- Condition Probability

  条件概率

  $$P(X|Y), P(Y|X)$$

- Marginal Probability 单一随机变数各自的分布

  $$P(X), P(Y)$$

Sum / Product Rule

- Sum Rule

x_i 等于任意y

$$\boxed{P(X = x_i)} = \sum_{Y} P(X, Y)$$

联合概率

Marginal Probability

- Product Rule

各自等于特定值

$$\boxed{P(X = x_i, Y = y_j)} = \underline{P(X|Y)P(Y) = P(Y|X)P(X)}$$

Joint Probability

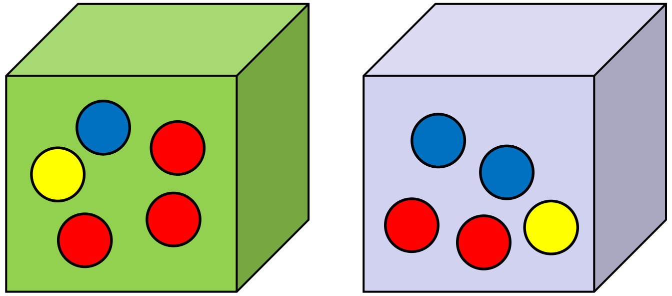
- Bayes Theorem

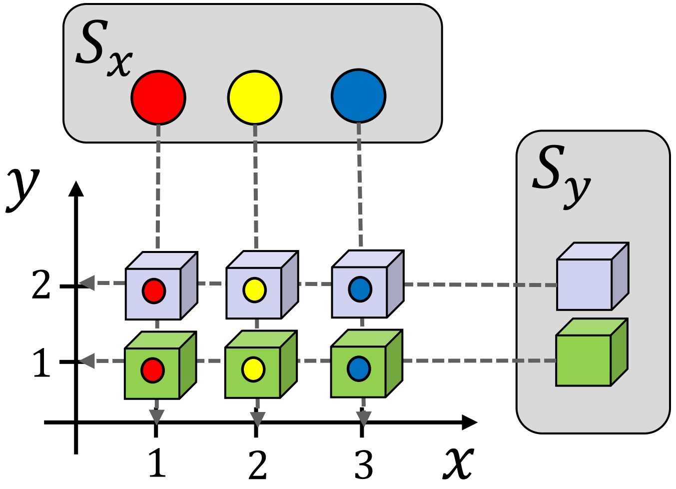
条件概率

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Sum / Product Rule Example

- Joint Probability and Marginal Probability





	X=1	X=2	X=3
P(X)	5/10	2/10	3/10

条件概率

P(X,Y)	X=1	X=2	X=3
Y=1	3/10	1/10	1/10
Y=2	2/10	1/10	2/10

	P(Y)
Y=1	1/2
Y=2	1/2

Independent

两个事件同时发生=两个事件分别发生

- Independent Event 事件独立性

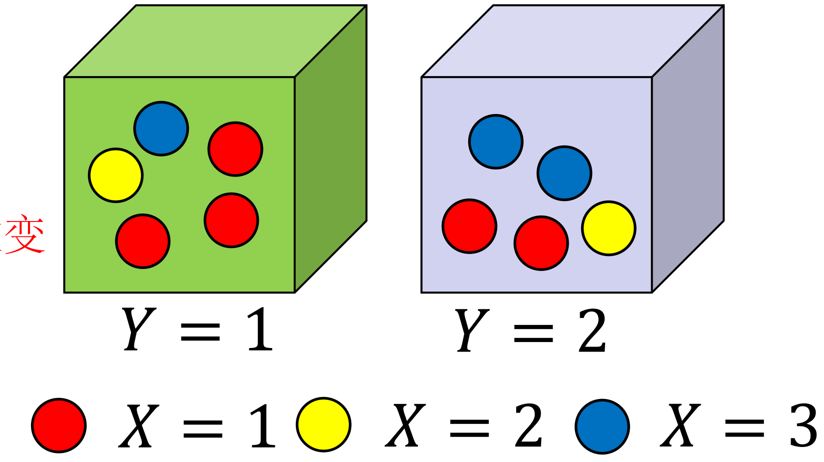
$$P(Y=1,X=2)=P(Y=1)P(X=2)$$

$$\frac{1}{10}=\frac{1}{2}\times\frac{2}{10}$$

y=1不因x=2而改变

$$P(Y=1)=P(Y=1|X=2)$$

$$\frac{1}{2}=\frac{1/10}{1/10+1/10}$$

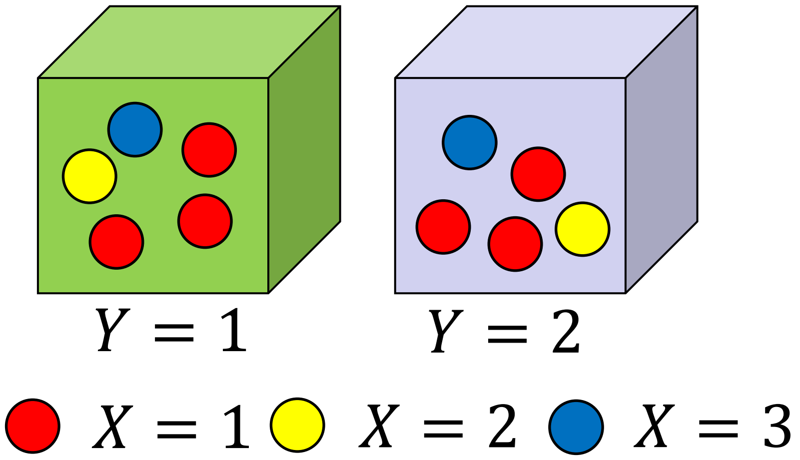


- Independent Random Variable

$$P(Y,X)=P(Y)P(X)$$

$$P(Y|X)=P(Y)$$

XY无论如何变化都不产生影响



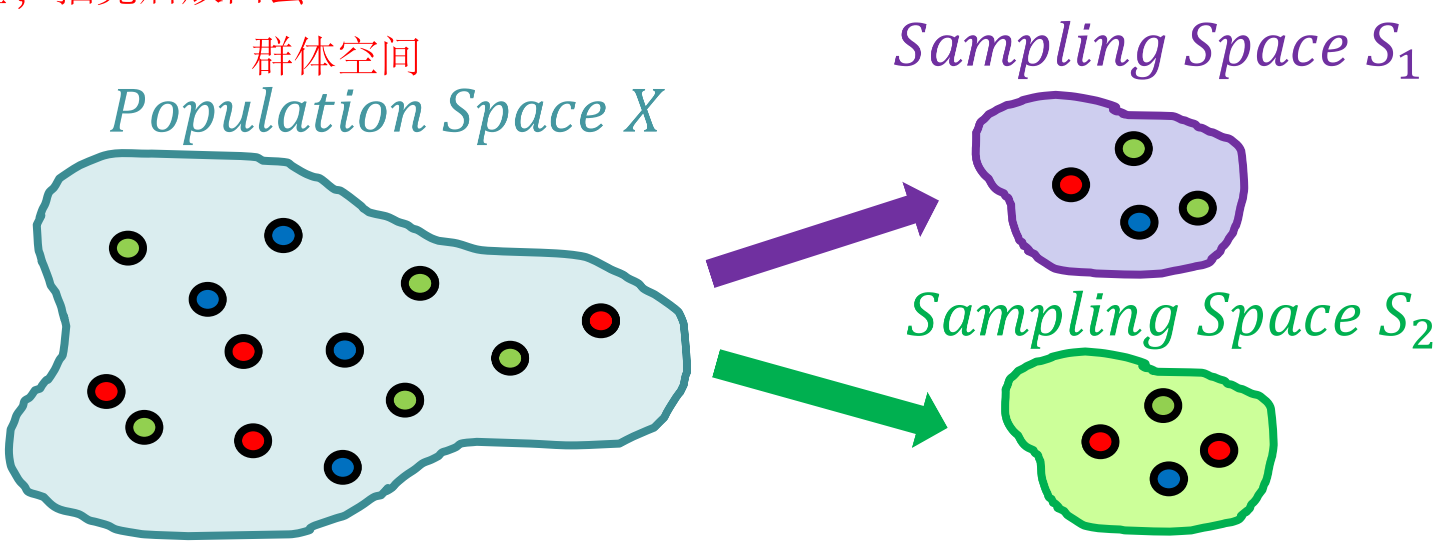
独立同分布

Independent and Identically Distributed (i.i.d.)

在不同次采样时都是从同样的population space采样，不会因为不同次采样，采样机率不一样

- We hope that the sampling process is Independent and Identically Distributed (i.i.d)
  - → The probability of each sampling data is independent and came from same probability distribution

抽签，抽完后放回去。



不同次采样机率是独立的

$$p(x_i \in S_1 \cap x_i \in S_2) = p(x_i \in S_1)p(x_i \in S_2)$$

Inference

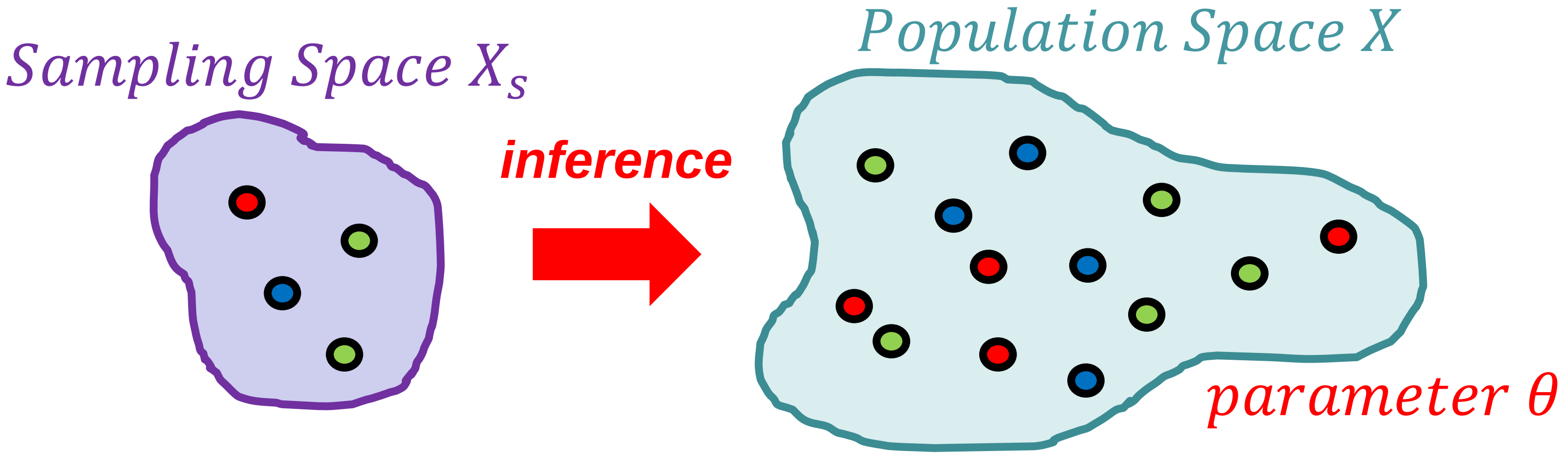
- Inference: 推论因果关系
  - A process to find the logical consequences from premises
  - In machine learning, we want to inference the probability of an event for a given condition $\boldsymbol{p}(\boldsymbol{Event} \mid \boldsymbol{Condition})$

- Example: Supervised Model train和test都是inference
  - $\boldsymbol{x}$ is input, $\boldsymbol{y}$ is output, $\boldsymbol{\theta}$ is the parameter of the model
  - Learning and Predicting are both inference tasks
  - Learning Tasks: $p(\theta \mid x, y)$
  - Predicting Tasks: $p_\theta(y \mid x)$ or $p(y \mid \theta, x)$

Statistical Inference 统计推理

不可能采样全部的种群空间进行学习

- A process to inference the parameters of population based on the information of sampling data
- $\boldsymbol{x_s}$ is sampled data, $\boldsymbol{\theta}$ is the parameter of distribution over population, statistical inference is to inference $\boldsymbol{p}(\boldsymbol{\theta} \mid \boldsymbol{x\_s})$



Statistical Inference

- Two approaches of statistical inference
  - Hypothesis Testing (Top-Down) 假设检验
    - Given a hypothesis of parameters, evaluate the correctness from sampling data

    假设一组参数，将train的数据代入参数的模型中，推测结果是否和标签相似
  - Estimation (Bottom-Up)
    - Find the most likely parameters from sampling data

    直接估测最适合的参数

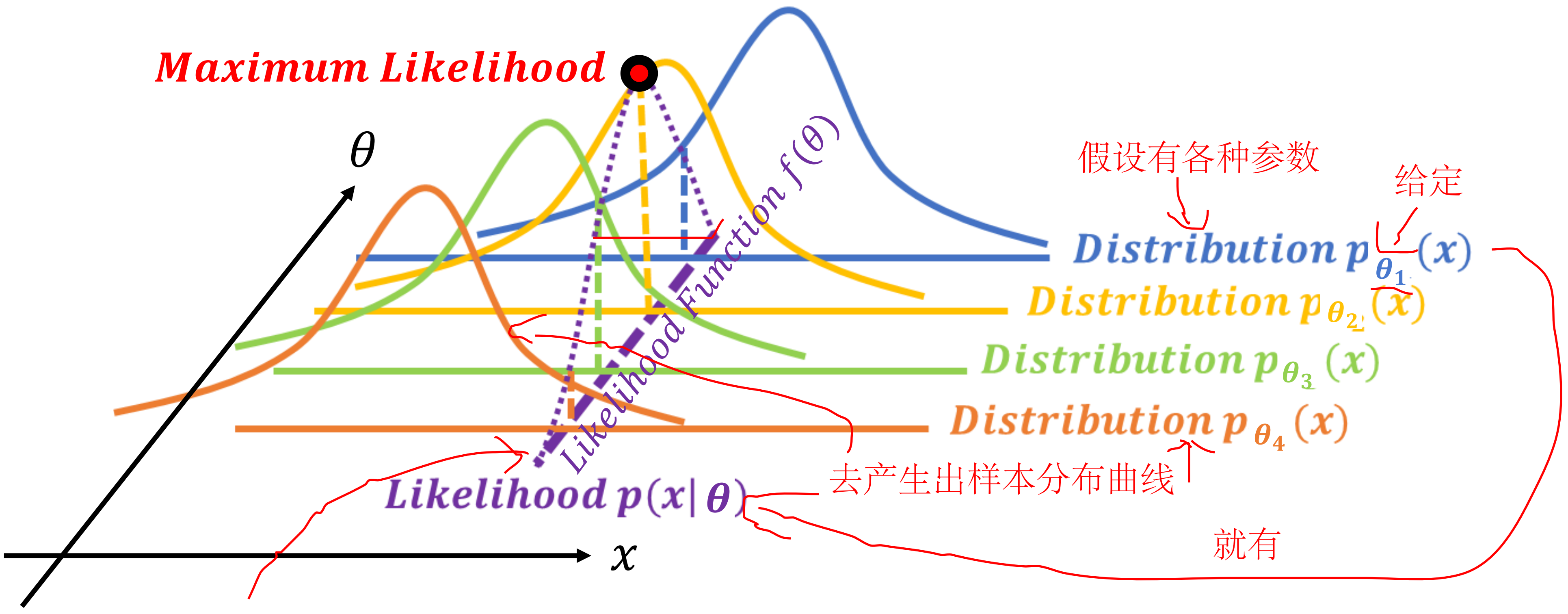
最大似然估计

Maximum Likelihood Estimation (MLE)

参数估测方法

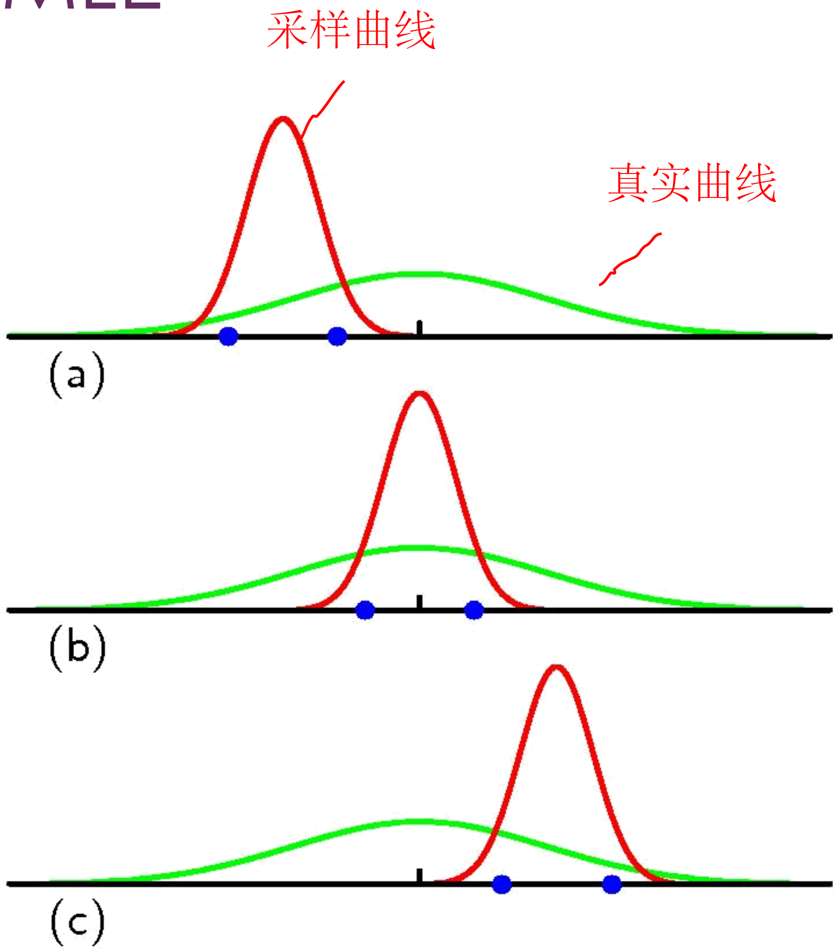
- Visualization of likelihood function

给一组x，找到对应参数。可以看哪一组参数有最高的机率产生x



指定一个x，就可以看出不同参数下的机率值

Problem of MLE



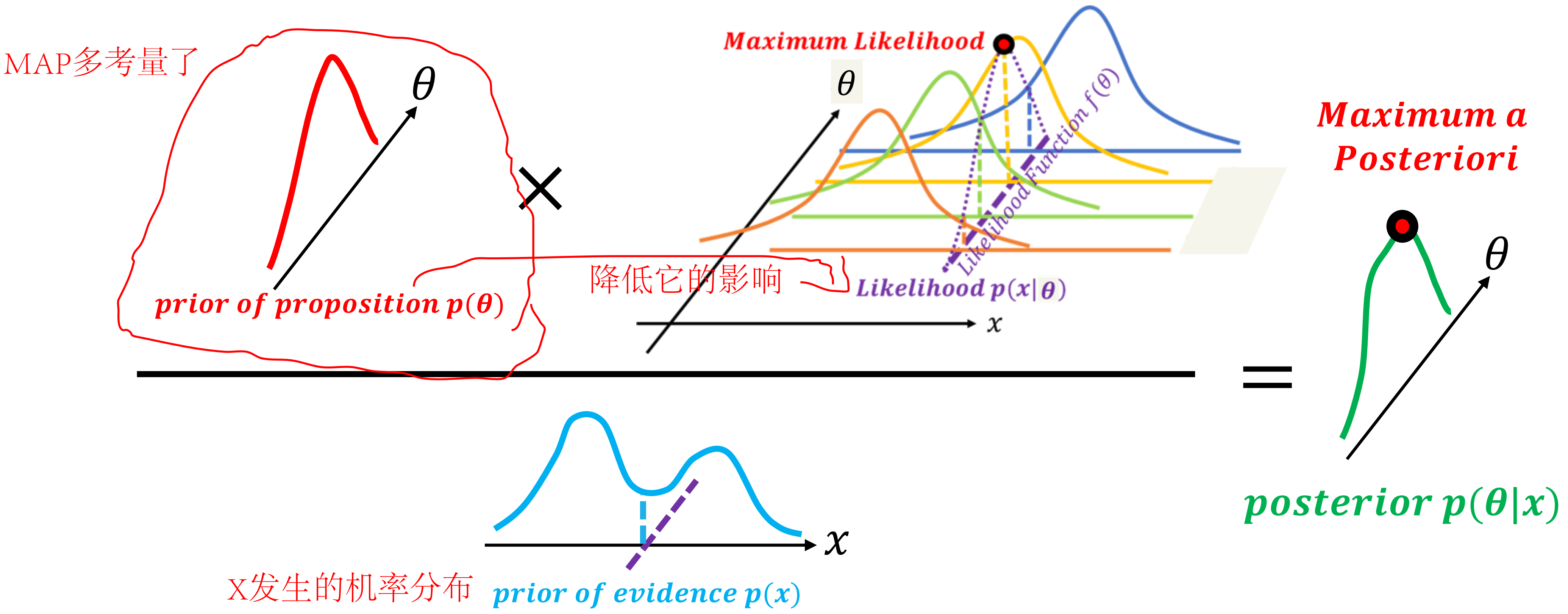
预测和真实差异大

最大化后验估计

Maximum a Posteriori Estimation (MAP)

- Visualization of Posterior Probability

$$\frac{p(\theta)p(x|\theta)}{p(x)} = p(\theta|x)$$



Example: Coin Estimation

- Toss a coin
  - [tail, tail, tail, head, tail]

- Likelihood $\boldsymbol{P(x \mid \theta)}$:

正面次数

  - ***Bernoulli distribution***: $\theta^{n}(1-\theta)^{m-n}$ 反面次数

正面机率 反面机率

  - ***MLE Estimation***:

最大化

$$\rightarrow \max_{p} \; \theta(1-\theta)^4$$

求极值，微分等于0

$$\frac{d\theta(1-\theta)^4}{d\theta} = (1-\theta)^4 + 4\theta(1-\theta)^3(-1) = (1-\theta)^3(5\theta-1) = 0$$

$\boldsymbol{\theta}$ =0.2

Example: Coin Estimation

- MAP Estimation (Assume Discrete Uniform Prior)

Prior
(Discrete Uniform)

Likelihood
(Bernoulli)
$\theta^n(1-\theta)^{m-n}$

$$\frac{p(\theta)p(x|\theta)}{p(x)} = p(\theta|x)$$

likelihood

Posterior

$$\frac{\begin{array}{l}\theta = 0.0\\ \theta = 0.1\\ \theta = 0.2\\ \vdots\end{array}\begin{bmatrix}1/11\\1/11\\1/11\\\vdots\end{bmatrix} \times \begin{bmatrix}(0)^1(1)^4\\(0.1)^1(0.9)^4\\(0.2)^1(0.8)^4\\\vdots\end{bmatrix}}{p(x) = \sum_\theta p(x,\theta) = \sum_\theta p(\theta)p(x|\theta)} = \begin{bmatrix}0.000\\0.213\\0.333\\\vdots\end{bmatrix}$$

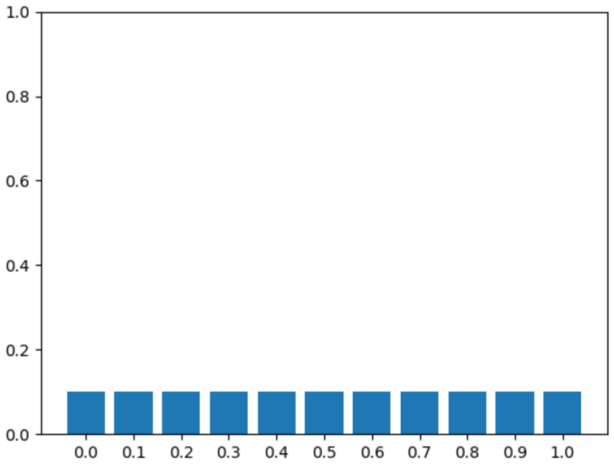
$\theta = 1$

Marginal Probability

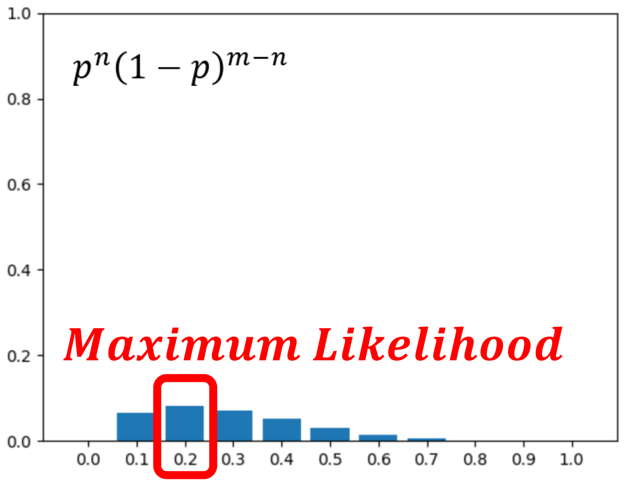
Example: Coin Estimation

- MAP Estimation
  - Prior: Discrete Uniform Distribution
  - Likelihood: Bernoulli distribution

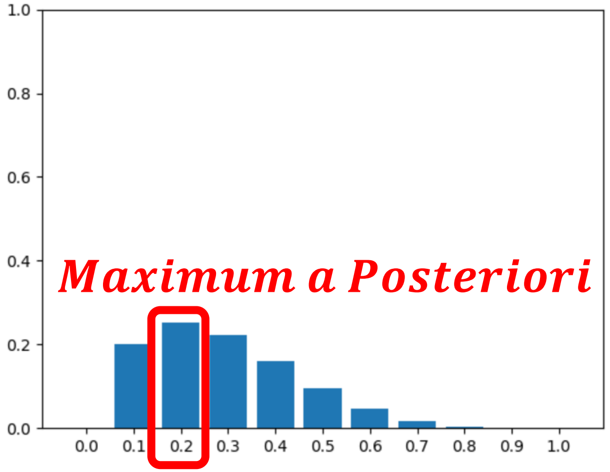
两种解法结果很类似



Prior: $\boldsymbol{P(\theta)}$



Likelihood: $\boldsymbol{P(x|\theta)}$



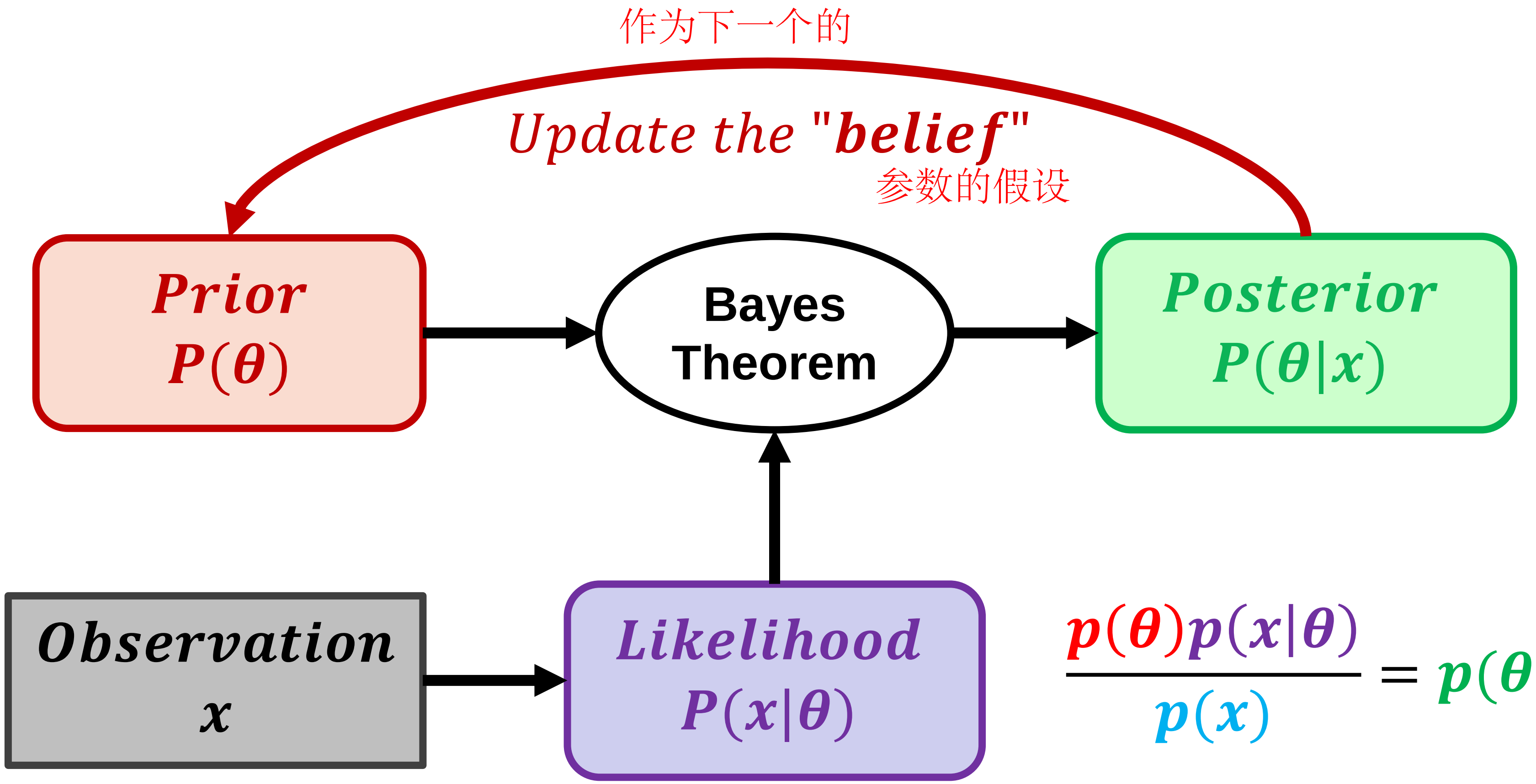
Posterior: $\boldsymbol{P(\theta|x)}$

Bayesian Probability

- Classical Probability View
  - Model parameters have a certain value. 认为预测参数有特定值
  - The goal of learning is to inference the parameters from sampling data which we call "Estimation". 推论
    估计
- Bayesian Probability View
  - Model parameters have uncertainty. 不确定性
  - The goal of learning is to inference the probability over every possible parameters, or inference the hyper-parameters. 推论机率分布
    超参数

Bayesian Approach

- The current hypothesis of the parameters is the “belief”



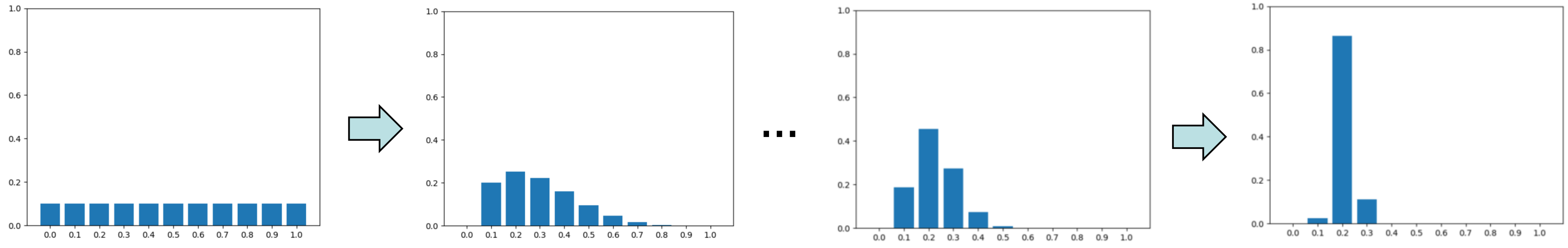
$$\frac{p(\theta)p(x|\theta)}{p(x)} = p(\theta|x)$$

Bayesian Approach (Tossing Coins Example)

$$\frac{p(\theta)p(x|\theta)}{p(x)} = p(\theta|x)$$

不在看让这两个值最大的结果

而是看thta的分布



Belief: $p^{(0)}(\theta)$ $p^{(1)}(\theta)$ $p^{(N-1)}(\theta)$ $p^{(N)}(\theta)$

Posterior: $p^{(0)}(\theta|x_1)$ $p^{(1)}(\theta|x_2)$ … $p^{(N-1)}(\theta|x_N)$

Bayes Filter

$\overline{bel}(\boldsymbol{\theta})$ 要估测thta就要得到它的分布

初始化 $\frac{p(\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x})} = p(\boldsymbol{\theta}|\boldsymbol{x})$

$bel(\boldsymbol{\theta})$

State Prediction:

$$P(\mathbf{x_t}|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t}) = \int P(\mathbf{x_t}|\mathbf{x}_{t-1},\mathbf{z}_{1:t-1},\mathbf{u}_{1:t})\mathrm{P}(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t})\,dx_{t-1}$$

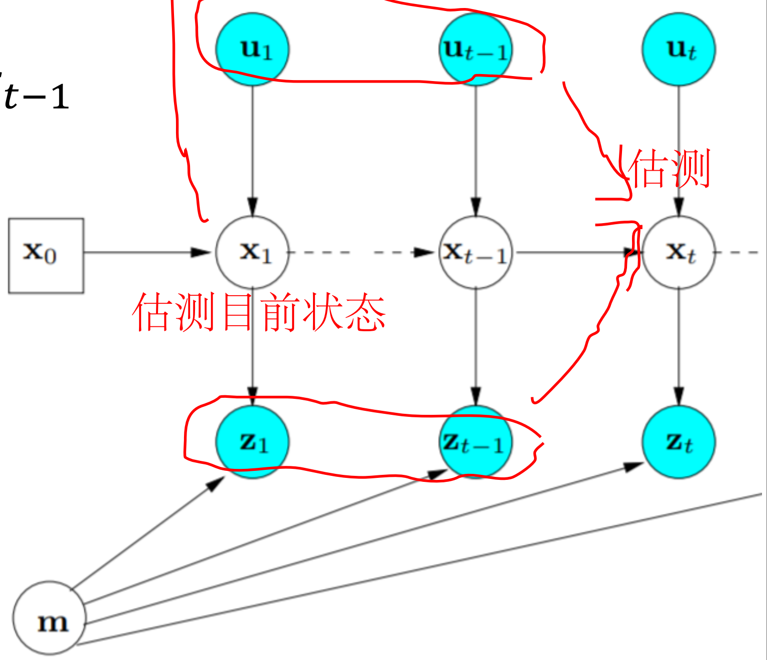
这一刻的估测

已知，之前的观测就不再重要了

$$= \int P(\mathbf{x_t}|\mathbf{x}_{t-1},\mathbf{u}_t)\mathrm{P}(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t})\,dx_{t-1}$$

前一刻的估测

$$\overline{bel}(\mathbf{x_t}) = \int P(\mathbf{x_t}|\mathbf{x}_{t-1},\mathbf{u}_t)bel(\mathbf{x_t})dx_{t-1}$$



Bayes Filter

$$\overline{bel}(\boldsymbol{\theta})$$

$$\frac{p(\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x})} = p(\boldsymbol{\theta}|\boldsymbol{x})$$

$$bel(\boldsymbol{\theta})$$

Measurement Update:

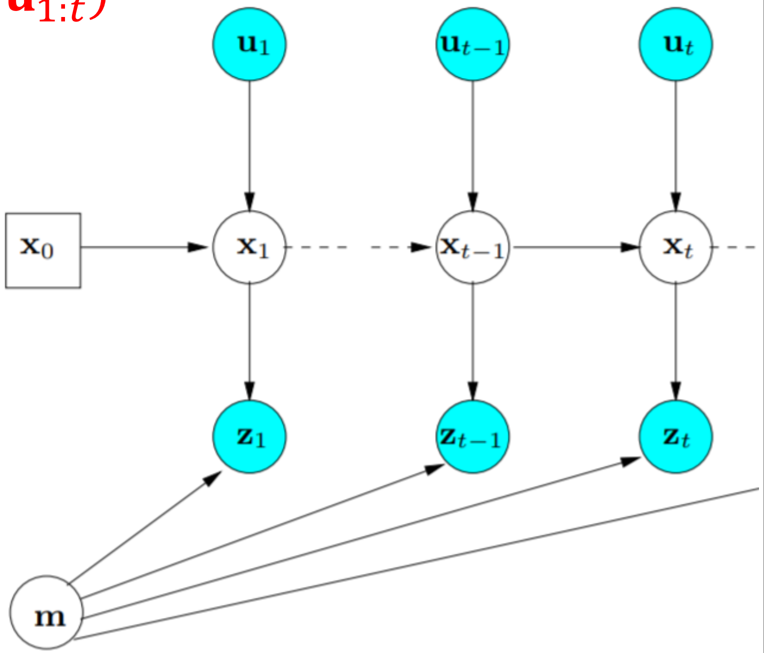
$$P(\mathbf{x_t}|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = \frac{P(\mathbf{z_t}|\mathbf{x_t}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})P(\mathbf{x_t}|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})}{P(\mathbf{z_t}|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})}$$

已经观测到Zt

$$= \eta\, P(\mathbf{z_t}|\mathbf{x_t}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})P(\mathbf{x_t}|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})$$

$$= \eta\, P(\mathbf{z_t}|\mathbf{x_t})P(\mathbf{x_t}|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})$$

$$bel(\mathbf{x_t}) = \eta P(\mathbf{z_t}|\mathbf{x}_t)\overline{bel}(\mathbf{x_t})$$



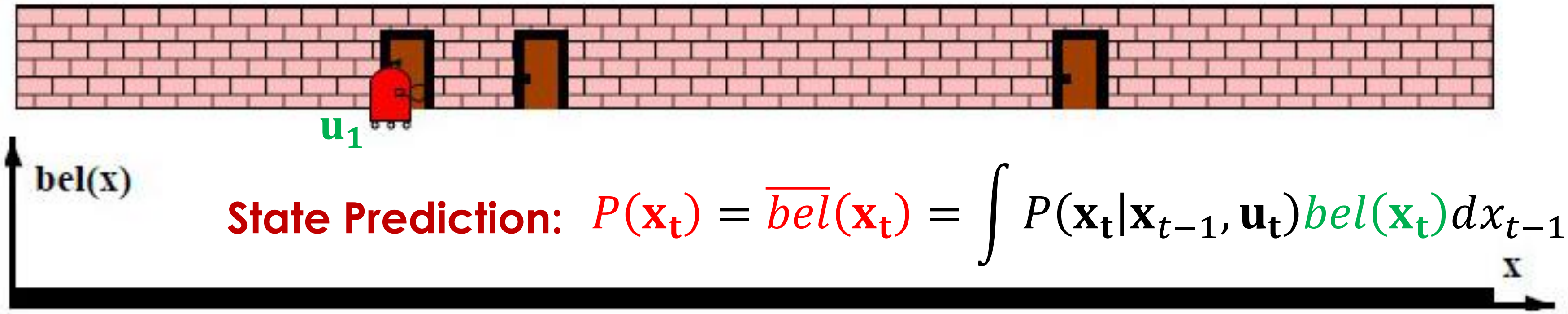
Bayes Filter

State Prediction: $P(\mathbf{x_t}) = \overline{bel}(\mathbf{x_t}) = \int P(\mathbf{x_t}|\mathbf{x}_{t-1}, \mathbf{u_t})bel(\mathbf{x_t})dx_{t-1}$

Measurement Update: $P(\mathbf{x_t}|\mathbf{z}_t) = bel(\mathbf{x_t}) = \eta P(\mathbf{z_t}|\mathbf{x}_t)\overline{bel}(\mathbf{x_t})$

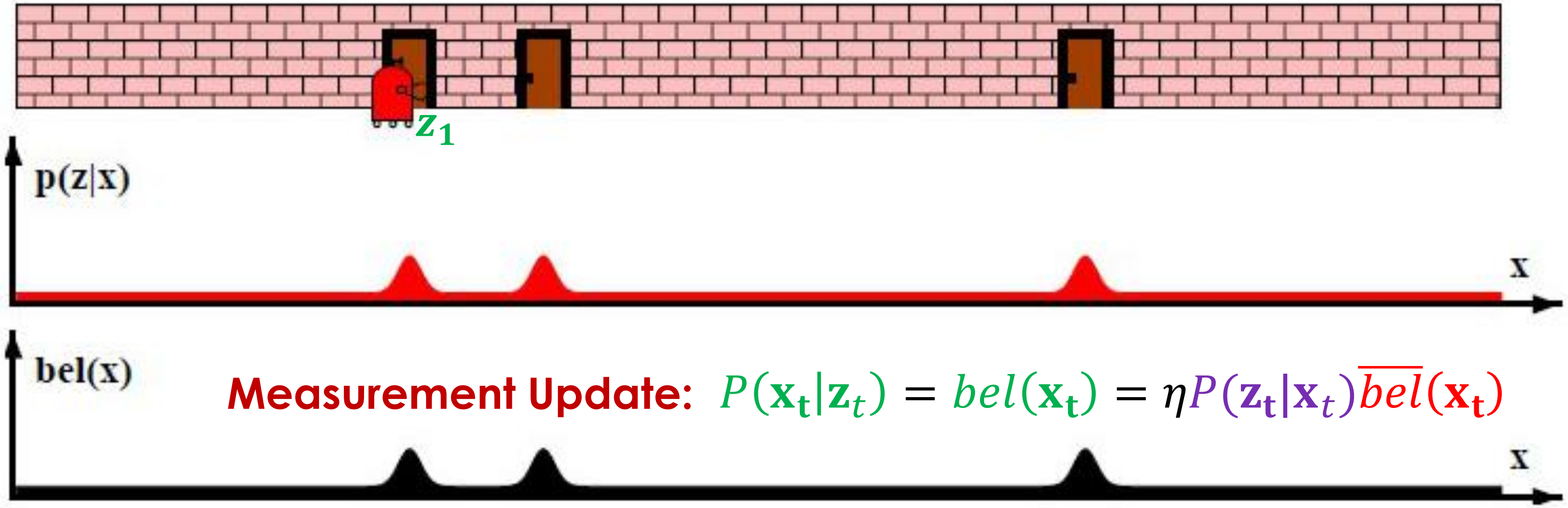
Algorithm Bayes_filter($bel(x_{t-1}), u_t, z_t$**):**
    for all x_t do
        $\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\ bel(x_{t-1})\ dx$
        $bel(x_t) = \eta\ p(z_t \mid x_t)\ \overline{bel}(x_t)$
    endfor
    return $bel(x_t)$

Localization



$\overline{bel}(\mathbf{x}_1)$

State Prediction: $P(\mathbf{x_t}) = \overline{bel}(\mathbf{x_t}) = \int P(\mathbf{x_t}|\mathbf{x}_{t-1}, \mathbf{u_t})bel(\mathbf{x_t})dx_{t-1}$

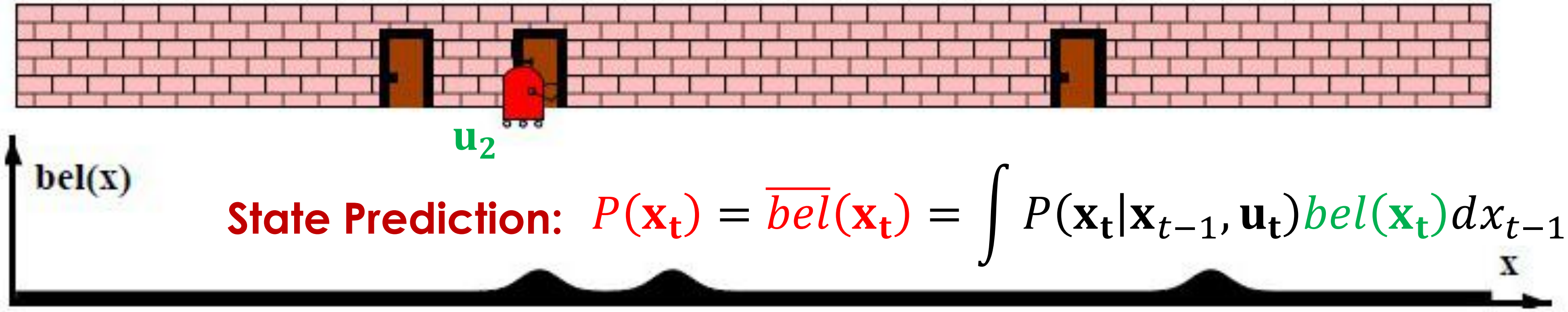


$bel(\mathbf{x}_1)$

Measurement Update: $P(\mathbf{x_t}|\mathbf{z}_t) = bel(\mathbf{x_t}) = \eta P(\mathbf{z_t}|\mathbf{x}_t)\overline{bel}(\mathbf{x_t})$

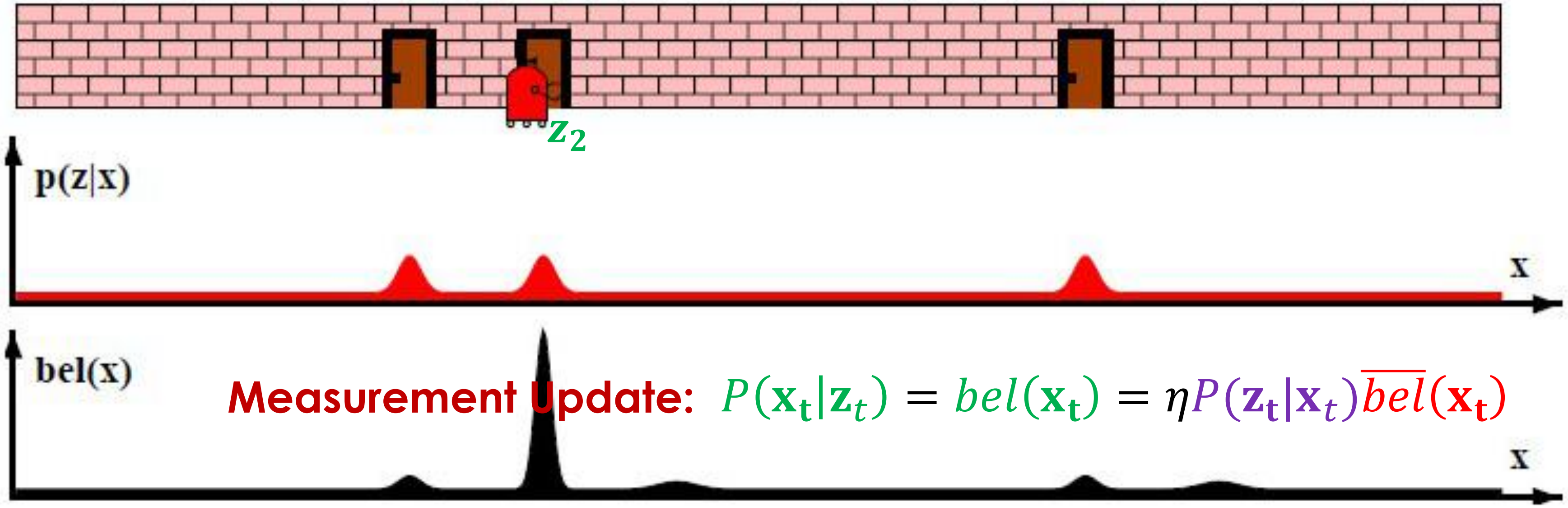
Localization

$\overline{bel}(\mathbf{x}_2)$



State Prediction: $P(\mathbf{x_t}) = \overline{bel}(\mathbf{x_t}) = \int P(\mathbf{x_t}|\mathbf{x}_{t-1}, \mathbf{u_t})bel(\mathbf{x_t})dx_{t-1}$

$bel(\mathbf{x}_2)$

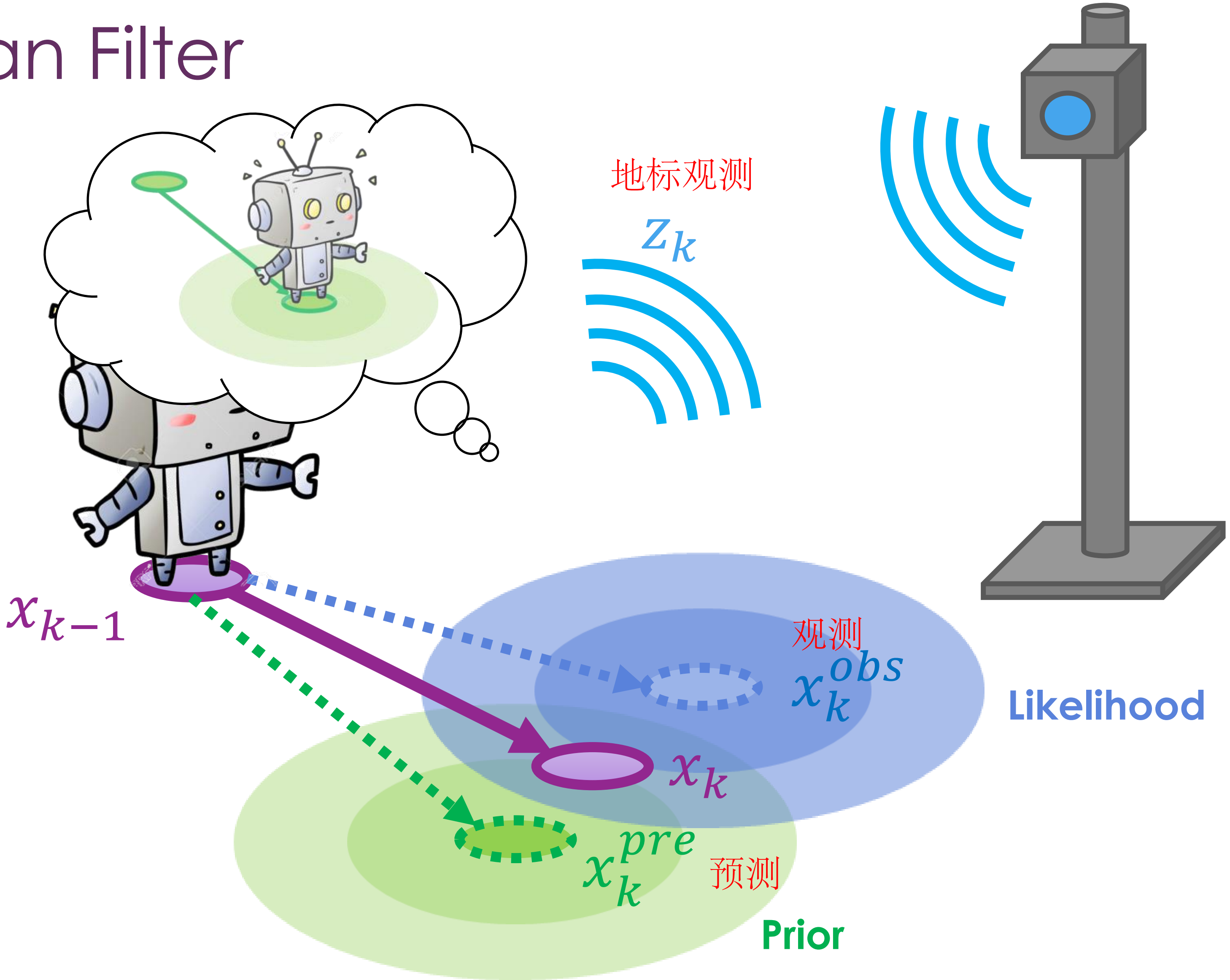


Measurement Update: $P(\mathbf{x_t}|\mathbf{z}_t) = bel(\mathbf{x_t}) = \eta P(\mathbf{z_t}|\mathbf{x}_t)\overline{bel}(\mathbf{x_t})$

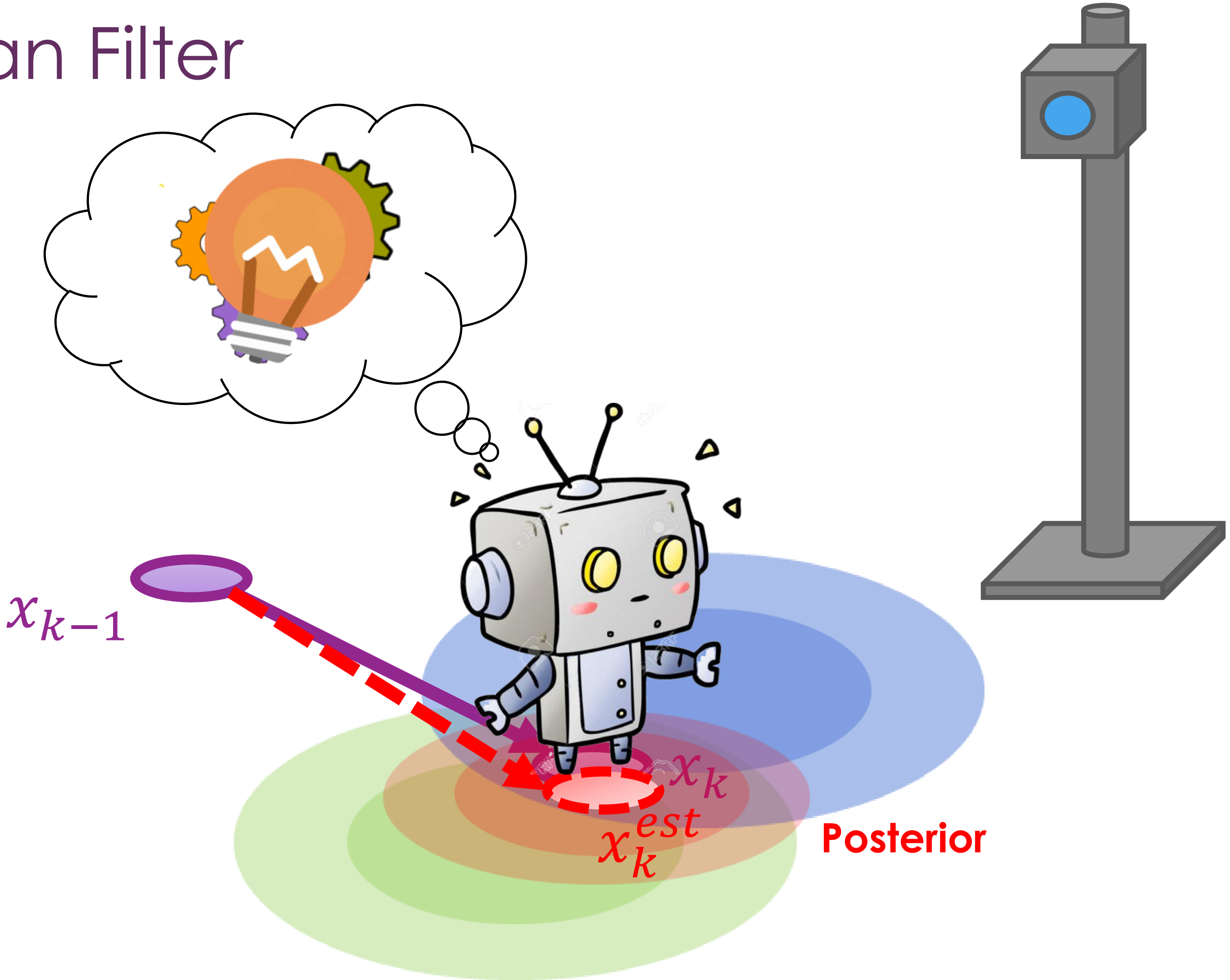
Outline

- State Estimation and SLAM Problem
- SLAM Back-end (Error Compensation)
  - Filter-based Methods
    - Probability Theory and Bayes Filter
    - Kalman Filter (KF) / Extended Kalman Filter (EKF) 经典基础
      - EKF-SLAM
    - Particle Filter
      - Fast-SLAM
  - Graph-based Methods
    - Pose Graph and Least-square Optimization
    - Gauss-Newton and Levenberg-Marquardt Algorithm
    - Sparse Matrix for Optimization

Kalman Filter

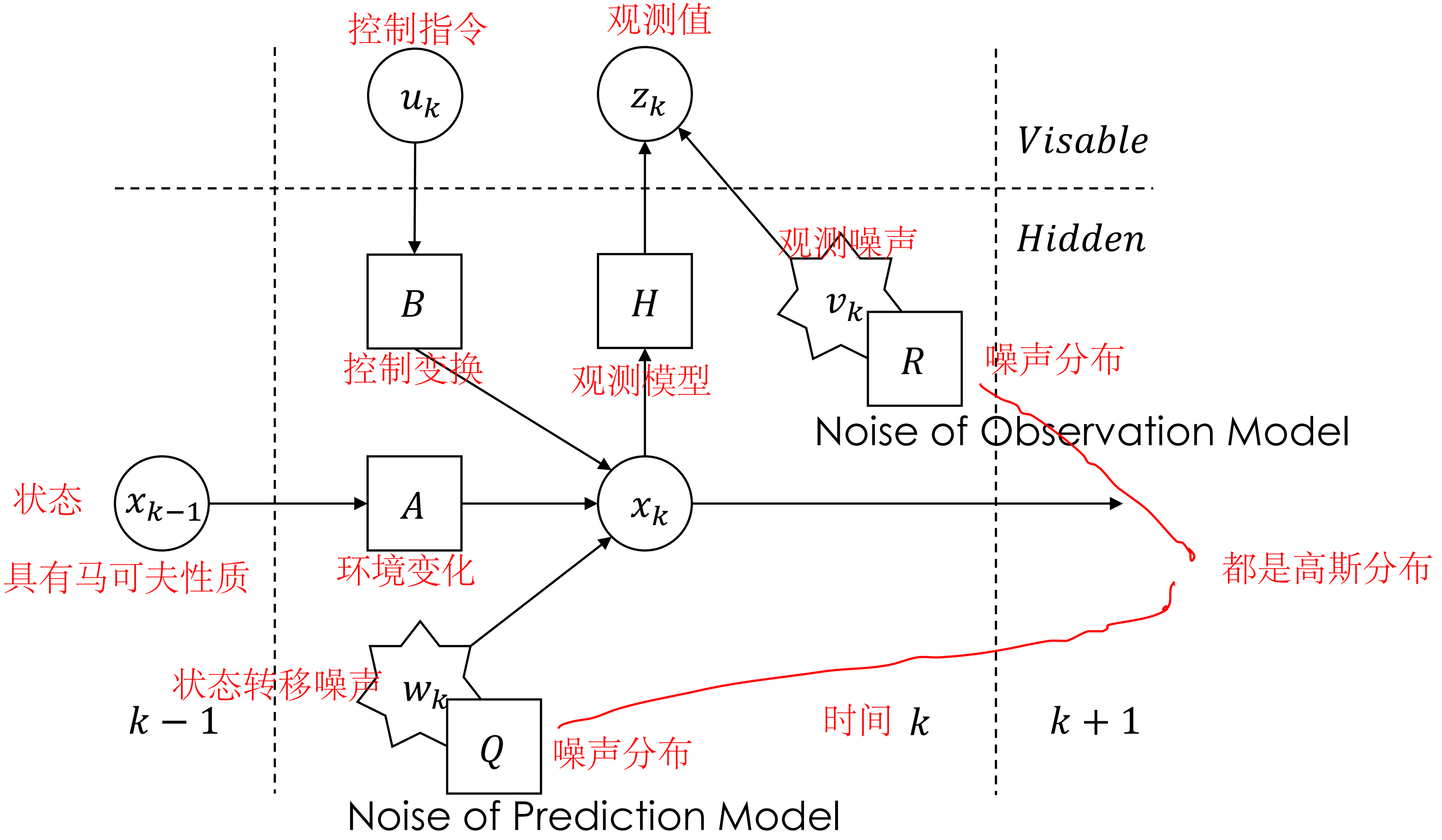


Kalman Filter



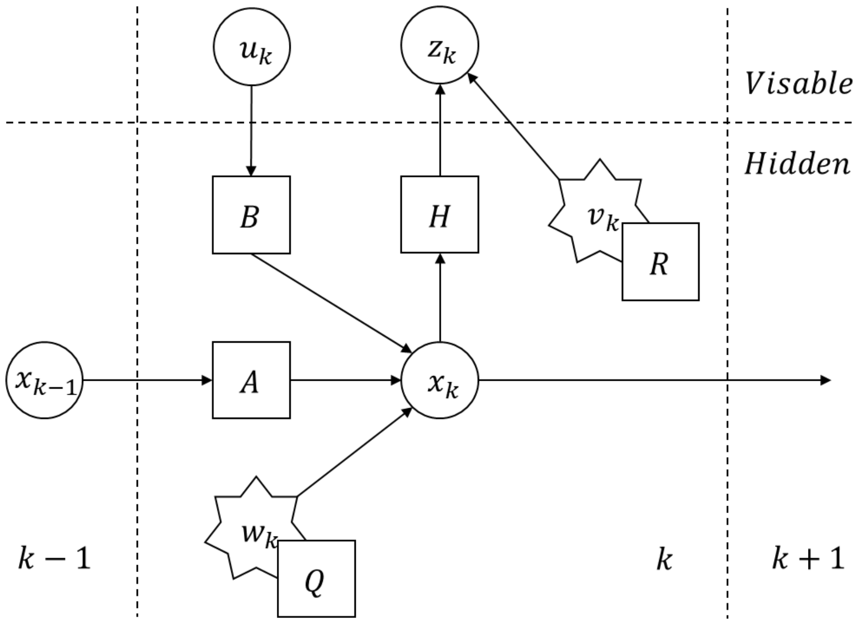
Kalman Filter

$$x_k = Ax_{k-1} + Bu_k + w_k$$
$$z_k = Hx_k + v_k$$

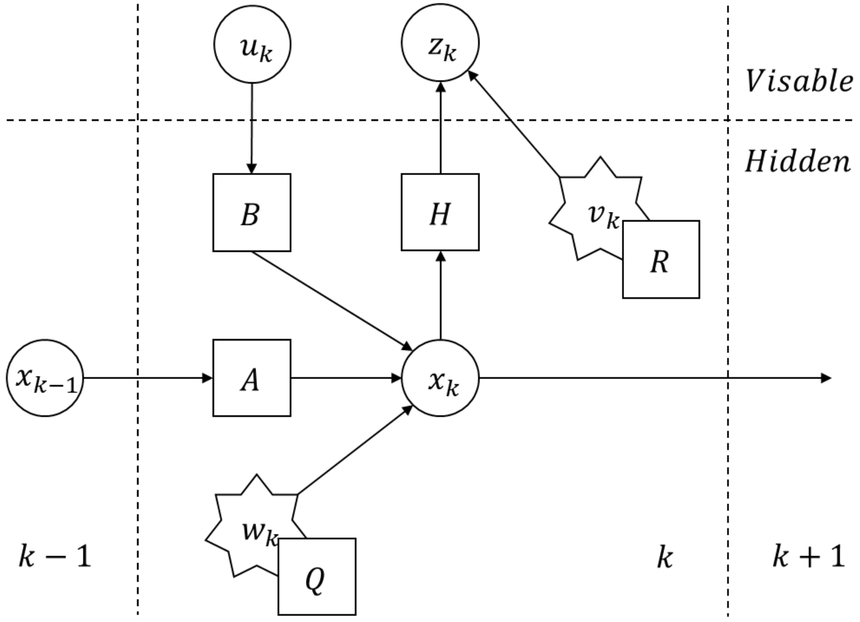


Kalman Filter

- Proof of Kalman Filter
- Notation
  - Ground Truth State: x_k 真实状态不可视
  - Prediction
    - ✓State: x_k^{pre} 预测状态
    - ✓Error: $e_k^{pre} = x_k - x_k^{pre}$,
    - ✓Covariance: $P_k^{pre} = E[e_k^{pre} {e_k^{pre}}^T]$
      期望值
  - Estimation
    - ✓State: x_k^{est} 估计状态
    - ✓Error: $e_k^{est} = x_k - x_k^{est}$
    - ✓Covariance: $P_k^{est} = E[e_k^{est} {e_k^{est}}^T]$



Kalman Filter



- The prediction of the state:

线性模型 ➢ $x_k^{pre} = Ax_{k-1}^{est} + Bu_k$

- Define the feedback equation:

预测状态误差补偿 ➢ $x_k^{est} = x_k^{pre} + \mathbf{K}(z_k - z_k^{pre})$

Kalman Gain

Observation Feedback

- Substitute the observation term of the feedback equation:

$$z_k = Hx_k + v_k\ , z_k^{pre} = Hx_k^{pre}$$

状态估计式

$$x_k^{est} = x_k^{pre} + K(Hx_k + v_k - Hx_k^{pre})$$
$$= x_k^{pre} + KH(x_k - x_k^{pre}) + Kv_k$$

- The object is to find the optimal Kalman Gain **K** to minimize the covariance of the estimation :

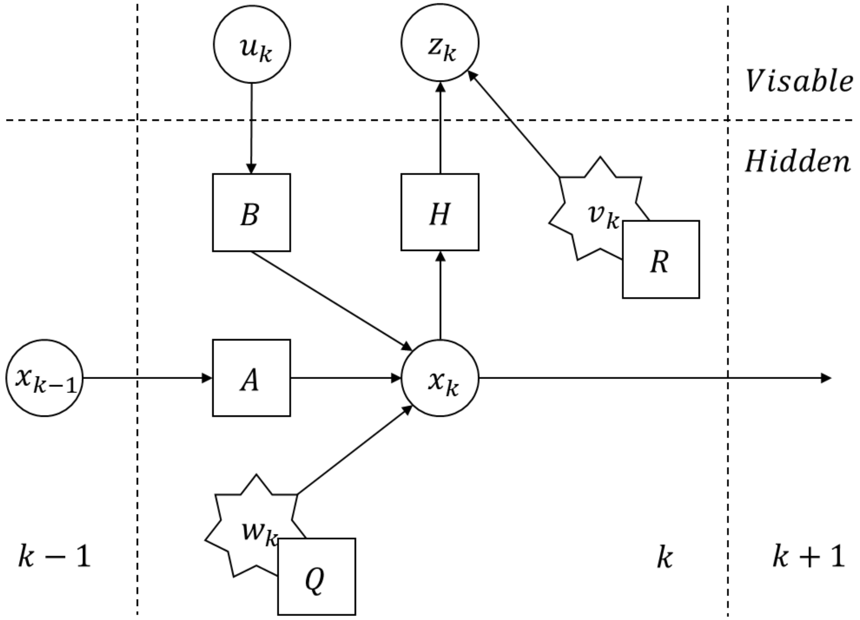
➢ $J = \sum_{min} P_k^{est}$ 最小化，找到卡尔曼增益。

Kalman Filter

- Propagate the error along the system.
- Compute the covariance of prediction
  - Prediction error:

预测误差

$$
\begin{aligned}
e_k^{pre} &= x_k - x_k^{pre} \\
&= (Ax_{k-1} + Bu_{k-1} + w_k) - (Ax_{k-1}^{est} + Bu_k) \\
&= A(x_{k-1} - x_{k-1}^{est}) + w_k = Ae_{k-1}^{est} + w_k
\end{aligned}
$$



协方差

  - Covariance:

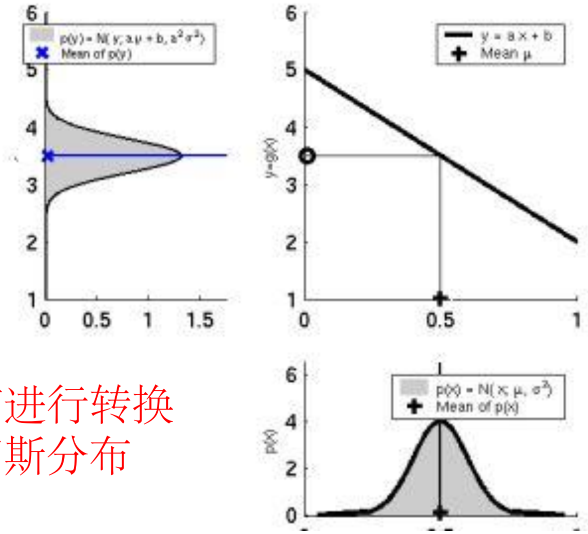
$$
\begin{aligned}
P_k^{pre} &= E\left[e_k^{pre} e_k^{pre\,T}\right] \\
&= E[(Ae_{k-1}^{est} + w_k)(Ae_{k-1}^{est} + w_k)^T] \\
&= E\left[Ae_{k-1}^{est} e_{k-1}^{est\,T} A^T\right] + E[w_k w_k^T] \\
&= AP_{k-1}^{est} A^T + Q
\end{aligned}
$$

这一刻产生新误差的协方差

前一次的协方差估计

转换矩阵二次缩放

误差根据斜率缩放



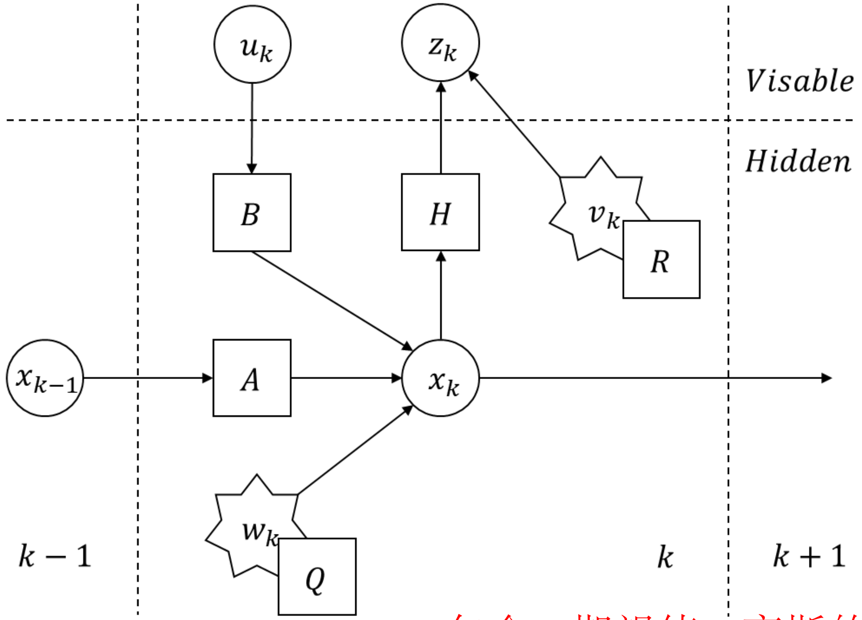
高斯分布进行转换
后还是高斯分布

Kalman Filter

- Estimate the covariance of posterior
  - Estimation error:

$$x_k^{est} = x_k^{pre} + KH(x_k - x_k^{pre}) + Kv_k$$

$$
\begin{aligned}
e_k^{est} &= x_k - x_k^{est} \\
&= \left(x_k - x_k^{pre}\right) - KH\left(x_k - x_k^{pre}\right) - Kv_k \\
&= (I - KH)e_k^{pre} - Kv_k
\end{aligned}
$$



  - Covariance:

预测的协方差

观测误差的协方差

包含vk期望值，高斯的期望值为0

$$
\begin{aligned}
P_k^{est} &= E\left[x_k^{est} {x_k^{est}}^T\right] \\
&= (I - KH)E\left[e_k^{pre} {e_k^{pre}}^T\right](I - KH)^T + KE\left[v_k v_k^T\right]K^T \cancel{- (I - KH)e_k^{pre} K \underbrace{E[v_k]}_{=0} - K^T E\left[v_k^T\right] {e_k^{pre}}^T (I - KH)^T} \\
&= (I - KH)P_k^{pre}(I - KH)^T + KRK^T = P_k^{pre} - KHP_k^{pre} - P_k^{pre}H^T K^T + K\left(HP_k^{pre}H_T + R\right)K^T
\end{aligned}
$$

- Optimize the objective function

求极值

$$\frac{\partial P_k^{est}}{\partial K} = -2\left(P_k^{pre}H^T\right) + 2K\left(HP_k^{pre}H^T + R\right) = 0$$

$$K = P_k^{pre}H^T\left(HP_k^{pre}H^T + R\right)^{-1}$$

Kalman Filter

- Kalman Filter Computation Steps:

Set the parameters of Kalman filter A, B, Q, R

1. Predict the next state

$$x_k^{pre} = Ax_{k-1}^{est} + Bu_k$$

2. Compute the prediction covariance

$$P_k^{pre} = AP_{k-1}^{est}A^T + Q$$

3. Compute Kalman-gain

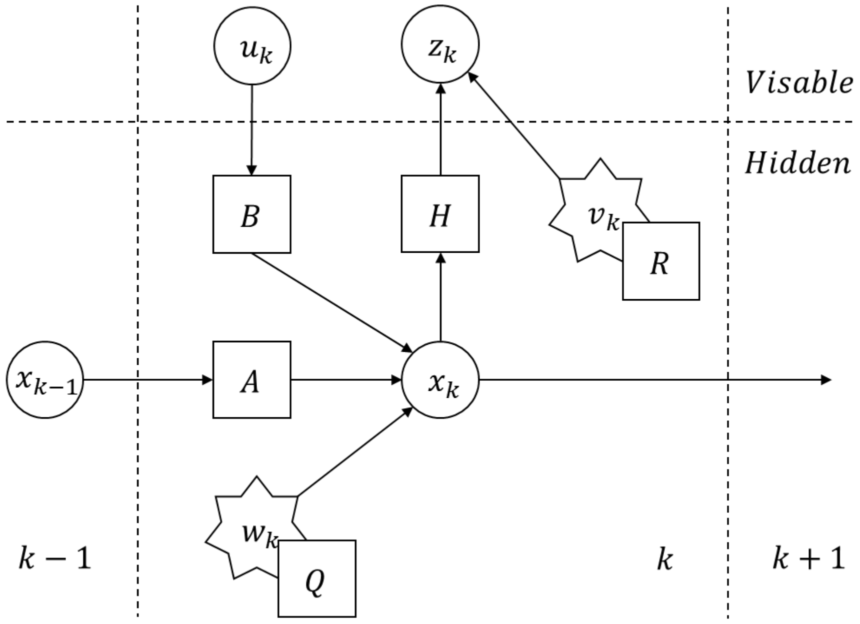
$$K_k = P_k^{pre}H^T\left(HP_k^{pre}H^T + R\right)^{-1}$$

4. Estimate the mean of the state

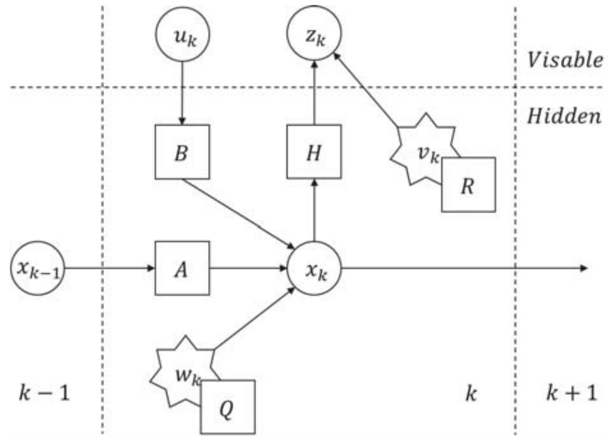
$$x_k^{est} = x_k^{pre} + K_k\left(z_k - Hx_k^{pre}\right)$$

5. Estimate the covariance of the state

$$P_k^{est} = (I - K_kH)P_k^{pre}$$



$$\begin{aligned} x_k^{pre} &= Ax_{k-1}^{est} + Bu_k \\ P_k^{pre} &= AP_{k-1}^{est}A^T + Q \\ K_k &= P_k^{pre}H^T\left(HP_k^{pre}H^T + R\right)^{-1} \\ x_k^{est} &= x_k^{pre} + K_k\left(z_k - Hx_k^{pre}\right) \\ P_k^{est} &= (I - K_kH)P_k^{pre} \end{aligned}$$

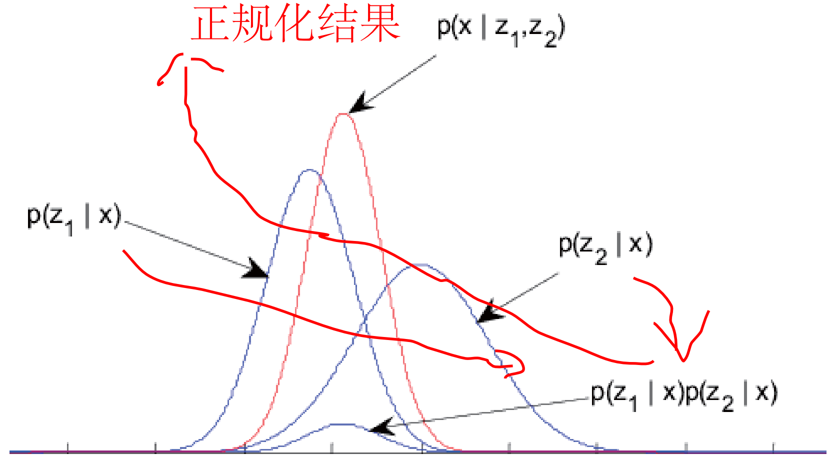


Kalman Filter

基于机率

- Probabilistic view of Kalman filter
- Prediction Model & Observation Model
  - $x_k = Ax_{k-1} + Bu_k + w_k$
  - $z_k = Hx_k + v_k$ $\quad\quad \boxed{x_k = H^{-1}(z_k - v_k)}$
- Probability distribution of the prediction and observation
  - $p(x_k^{pre}) = \mathcal{N}(Ax_{k-1}^{est} + Bu_k, AP_{k-1}^{est}A^T + Q)$
  - $p(x_k^{obs}) = \mathcal{N}(H^{-1}z_k, H^{-1}RH^{-T})$
- Fusion of Gaussian Distribution
  - $S = \Sigma_0(\Sigma_0 + \Sigma_1)^{-1}$
  - $\mu = \mu_0 + S(\mu_1 - \mu_0)$
  - $\Sigma = \Sigma_0 - S\Sigma_0$

正规化结果



Kalman Filter

$$S = \Sigma_0(\Sigma_0 + \Sigma_1)^{-1}$$
$$\mu = \mu_0 + S(\mu_1 - \mu_0)$$
$$\Sigma = \Sigma_0 - K\Sigma_0$$

$$p(x_k^{pre}) = \mathcal{N}(Ax_{k-1}^{est} + Bu_k, AP_{k-1}^{est}A + Q)$$
$$p(x_k^{obs}) = \mathcal{N}(H^{-1}z_k, H^{-1}RH^{-T})$$

- Fusion the distribution of prediction and observation
  - Mean: x_k^{est}

$$= x_{k-1}^{pre} + P_k^{pre}\left(P_k^{pre} + H^{-1}RH^{-T}\right)^{-1}\left(x_k^{pre} - H^{-1}z_k\right)$$
$$= x_{k-1}^{pre} + P_k^{pre}H^TH^{-T}\left(P_k^{pre} + H^{-1}RH^{-T}\right)^{-1}H^{-1}H\left(x_k^{pre} - H^{-1}z_k\right)$$
$$= x_{k-1}^{est} + Bu_k + P_k^{pre}H^T\left(HP_k^{pre}H^T + R\right)^{-1}\left(Hx_k^{pre} - z_k\right)$$
$$= x_{k-1}^{est} + Bu_k + K_k\left(Hx_k^{pre} - z_k\right)$$

$$A^{-1}B^{-1} = (BA)^{-1}$$

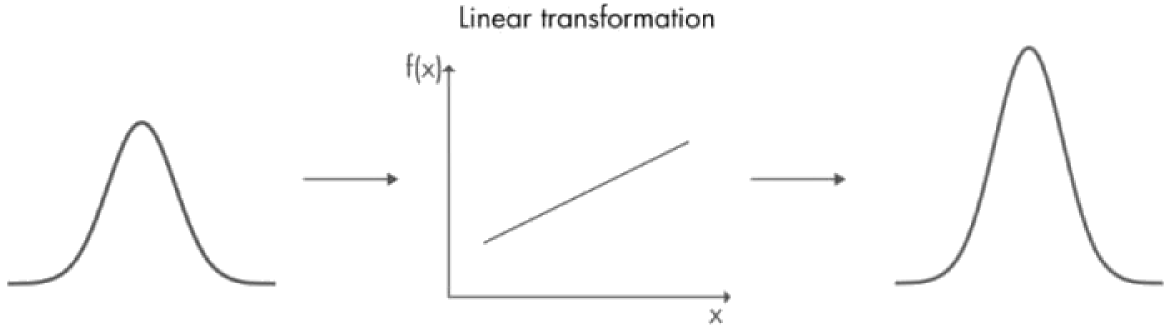
  - Covariance: P_k^{est}

$$= P_k^{pre} - P_k^{pre}\left(P_k^{pre} + H^{-1}RH^{-T}\right)^{-1}P_k^{pre}$$
$$= P_k^{pre} - P_k^{pre}H^TH^{-T}\left(P_k^{pre} + H^{-1}RH^{-T}\right)^{-1}H^{-1}HP_k^{pre}$$
$$= P_k^{pre} - P_k^{pre}H^T\left(HP_k^{pre}H^T + R\right)^{-1}HP_k^{pre}$$
$$= P_k^{pre} - K_kHP_k^{pre} = (I - K_kH)P_k^{pre}$$

$$x_k^{pre} = Ax_{k-1}^{est} + Bu_k$$
$$P_k^{pre} = AP_{k-1}^{est}A^T + Q$$
$$K_k = P_k^{pre}H^T\left(HP_k^{pre}H^T + R\right)^{-1}$$
$$x_k^{est} = x_k^{pre} + K_k\left(z_k - Hx_k^{pre}\right)$$
$$P_k^{est} = (I - K_kH)P_k^{pre}$$

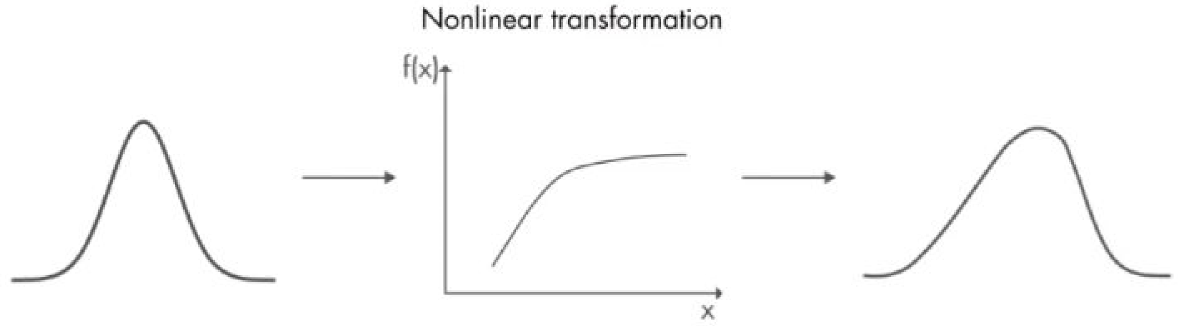
Extended Kalman Filter (EKF)

- Kalman filter assumes the prediction model to be linear, the Gaussian distribution of the state will transform to another Gaussian :



- However, the prediction model is usually nonlinear, the state distribution after transformation will not be a Gaussian.

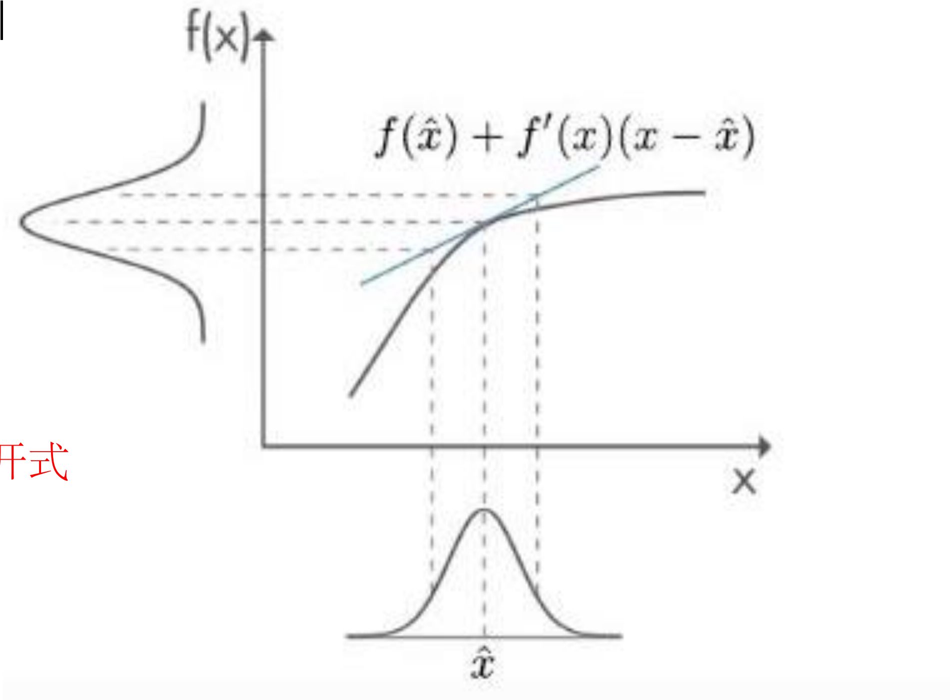
非线性的，卡曼滤波就不适用了



Extended Kalman Filter (EKF)

线性近似

- In this case, we can approximate the nonlinear transform by utilizing the 1st order Taylor expansion at the mean of the state:
- Prediction Model & Observation Model
  - $x_k = f(x_{k-1}, u_k) + w_k$
  - $z_k = h(x_k) + v_k$



- Jacobian Matrix:

ABH →
  - $F_k = \frac{\partial f(\hat{x}_{k-1}, u_k)}{\partial x}, H_k = \frac{\partial h(\hat{x}_k)}{\partial x}$
- Linearized System

常数项，泰勒展开式
  - $x_k = f(\hat{x}_{k-1}, u_k) + F_k(x_{k-1} - \hat{x}_{k-1}) + w_k$
  - $z_k = h(\hat{x}_k) + H_k(x_k - \hat{x}_k) + v_k$

Extended Kalman Filter (EKF)

- Linearized System
  - $x_k = f(\hat{x}_{k-1}, u_k) + F_k(x_{k-1} - \hat{x}_{k-1}) + w_k$
  - $z_k = h(\hat{x}_k) + H_k(x_k - \hat{x}_k) + v_k$
- Computation of EKF

$$x_k^{pre} = f(x_{k-1}^{est}, u_k)$$

$$P_k^{pre} = F_k P_{k-1}^{pre} F_k^T + Q$$

$$K_k = P_k^{pre} H^T \left(H P_k^{pre} H^T + R\right)^{-1}$$

$$x_k^{est} = x_k^{pre} + K_k\left(z_k - H x_k^{pre}\right)$$

$$P_k^{est} = (I - K_k H) P_k^{pre}$$

Kalman-Filter

$$x_k^{pre} = A x_{k-1}^{est} + B u_k$$

$$P_k^{pre} = A P_{k-1}^{est} A^T + Q$$

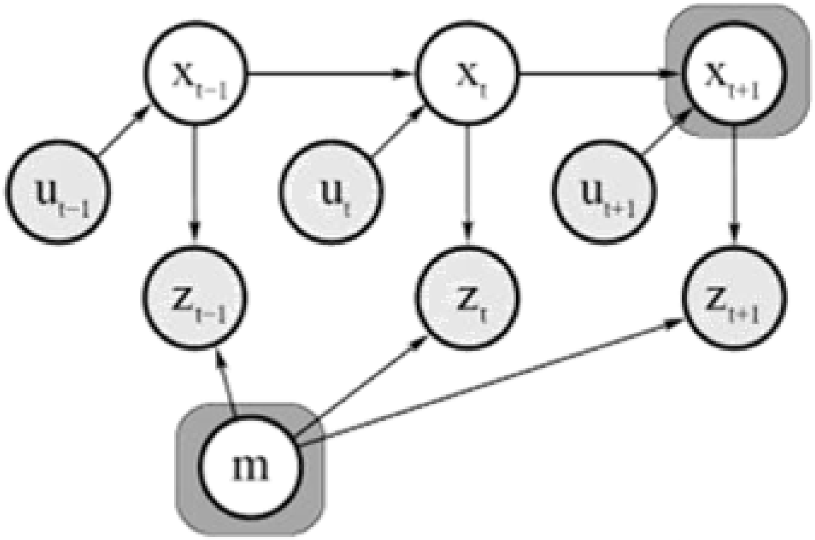
$$K_k = P_k^{pre} H^T \left(H P_k^{pre} H^T + R\right)^{-1}$$

$$x_k^{est} = x_k^{pre} + K_k\left(z_k - H x_k^{pre}\right)$$

$$P_k^{est} = (I - K_k H) P_k^{pre}$$

EKF-SLAM

- Consider the SLAM problem
- Define the state as the concatenation of robot's pose and landmarks position:



map

$$s_k = (\underbrace{x, y, \theta}_{\text{robot's pose}}, \underbrace{m_{1,x}, m_{1,y}}_{\text{Landmark 1}}, \underbrace{m_{2,x}, m_{2,y}}_{\text{Landmark 2}}, \dots, \underbrace{m_{n,x}, m_{n,y}}_{\text{Landmark n}})^T$$

- Probability distribution of the state:

状态的分布

pose自己的关联性

pose与地图的关联性

$$\begin{bmatrix} x \\ y \\ \theta \\ m_{1,x} \\ m_{1,y} \\ \vdots \\ m_{n,x} \\ m_{n,y} \end{bmatrix} \rightarrow \mu = \begin{bmatrix} \mathbf{x} \\ \mathbf{m} \end{bmatrix}, \Sigma = \begin{bmatrix} \Sigma_{\mathbf{xx}} & \Sigma_{\mathbf{xm}} \\ \Sigma_{\mathbf{mx}} & \Sigma_{\mathbf{mm}} \end{bmatrix}$$

地图自己的关联性

Extended Kalman-Filter

$$x_k^{pre} = f(x_k^{est}, u_k)$$
$$P_k^{pre} = F_k P_{k-1}^{est} F_k^T + Q$$
$$K_k = P_k^{pre} H^T (H P_k^{pre} H^T + R)^{-1}$$
$$x_k^{est} = x_k^{pre} + K_k(z_k - H x_k^{pre})$$
$$P_k^{est} = (I - K_k H) P_k^{pre}$$

EKF-SLAM

- In the past section, we have learnt the equation of motion model

运动微分方程 $$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\ \cos\theta \\ v\ sin\theta \\ \omega \end{bmatrix}$$

- In simulation process, we utilize the numerical integral to compute the future state with a small interval dt.

- However, in SLAM task we need an accurate state prediction for a given interval Δt, which can be obtained by integrating over the motion equation:

$$\begin{cases} x(t) = \int v\ cos\theta\ dt \\ y(t) = \int v\ sin\theta\ dt \\ \theta(t) = \int \omega\ dt \end{cases}$$

避免数值积分带来的误差

直接对微分方程进行积分

EKF-SLAM (Prediction Model)

- First, we integrate the angle:
  - $\theta(t) = \int \omega \, dt, \quad \theta(t) = \omega t + C$ 常数 时间的线性函数
- Consider the initial condition of angle
  - $\theta(0) = \hat{\theta}$, we can get the scalar term $C = \hat{\theta}$ 初始角度
- Then we can substitute the angle term for integral of x and y
  - $x(t) = \int v \cos(\hat{\theta} + \omega t) \, dt = \frac{v}{\omega} \sin\left(\hat{\theta} + \omega t\right) + C$

    $y(t) = \int v \sin(\hat{\theta} + \omega t) \, dt = -\frac{v}{\omega} \cos\left(\hat{\theta} + \omega t\right) + C$
- Consider the initial condition of position
  - $x(0) = \hat{x}, y(0) = \hat{y}$, we can get

运动方程

$$x(t) = \int v \cos\left(\hat{\theta} + \omega t\right) dt = \frac{v}{\omega} \sin\left(\hat{\theta} + \omega t\right) - \frac{v}{\omega} \sin\left(\hat{\theta}\right) + \hat{x}$$

$$y(t) = \int v \sin\left(\hat{\theta} + \omega t\right) dt = -\frac{v}{\omega} \cos\left(\hat{\theta} + \omega t\right) + \frac{v}{\omega} \cos\left(\hat{\theta}\right) + \hat{y}$$

EKF-SLAM (Prediction Model)

- Prediction Model

$$\begin{cases} x' = \hat{x} - \frac{v}{\omega}\sin(\hat{\theta}) + \frac{v}{\omega}\sin(\hat{\theta} + \omega\Delta t) \\ y' = \hat{y} + \frac{v}{\omega}\cos(\hat{\theta}) - \frac{v}{\omega}\cos(\hat{\theta} + \omega\Delta t), \\ \theta' = \omega\Delta t + \hat{\theta} \end{cases} \quad \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} -\frac{v}{\omega}\sin(\theta) + \frac{v}{\omega}\sin(\theta + \omega_t\Delta t) \\ \frac{v}{\omega}\cos(\theta) - \frac{v}{\omega}\cos(\theta + \omega_t\Delta t) \\ \omega\Delta t \end{bmatrix}$$

- Linearized the velocity motion model:

矩阵偏微分

➢ $F_t^x = \frac{\partial f}{\partial(x,y,\theta)^T}\left[\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} -\frac{v_t}{\omega_t}\sin(\theta) + \frac{v_t}{\omega_t}\sin(\theta + \omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos(\theta) - \frac{v_t}{\omega_t}\cos(\theta + \omega_t\Delta t) \\ \omega\Delta t + \theta \end{bmatrix}\right] = I + \frac{\partial f}{\partial(x,y,\theta)^T}\begin{bmatrix} -\frac{v_t}{\omega_t}\sin(\theta) + \frac{v_t}{\omega_t}\sin(\theta + \omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos(\theta) - \frac{v_t}{\omega_t}\cos(\theta + \omega_t\Delta t) \\ \omega_t\Delta t \end{bmatrix}$

$$= I + \begin{bmatrix} 0 & 0 & -\frac{v_t}{\omega_t}\cos(\theta) + \frac{v_t}{\omega_t}\cos(\theta + \omega_t\Delta t) \\ 0 & 0 & -\frac{v_t}{\omega_t}\sin(\theta) + \frac{v_t}{\omega_t}\sin(\theta + \omega_t\Delta t) \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\frac{v_t}{\omega_t}\cos(\theta) + \frac{v_t}{\omega_t}\cos(\theta + \omega_t\Delta t) \\ 0 & 1 & -\frac{v_t}{\omega_t}\sin(\theta) + \frac{v_t}{\omega_t}\sin(\theta + \omega_t\Delta t) \\ 0 & 0 & 1 \end{bmatrix}$$

EKF-SLAM (Observation Model)

- Obtain the relative measurement of landmarks: $z_i = (r_i, \phi_i)^T$

距离　车子坐标系的角度

  - $\begin{bmatrix} m_{i,x} \\ m_{i,y} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} r_i \cos(\phi_i + \theta) \\ r_i \sin(\phi_i + \theta) \end{bmatrix}$

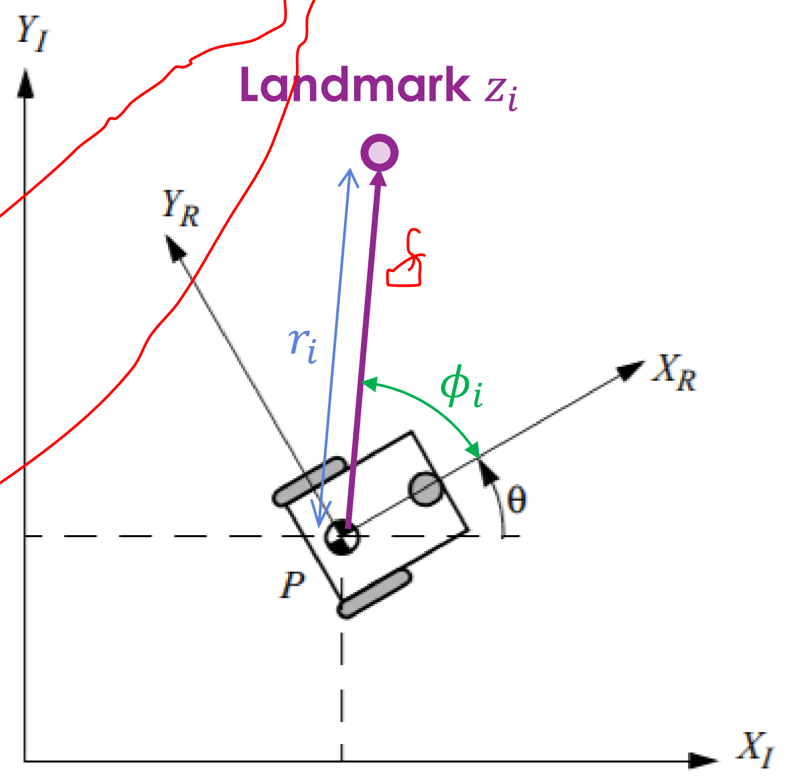
反观测模型　观测转换到状态

- Define the following term:

向量

  - $\delta = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = \begin{bmatrix} m_{i,x} - x \\ m_{i,y} - y \end{bmatrix}$, $q = \delta^T \delta = \delta^2$

- The observation can be represented as:
  - $z_i = \begin{bmatrix} \sqrt{q} \\ atan2(\delta_x, \delta_y) - \theta \end{bmatrix}$



EKF-SLAM (Observation Model)

- Given observation model

$$z_i = \begin{bmatrix} \sqrt{q} \\ atan2(\delta_x, \delta_y) - \theta \end{bmatrix}, \delta = \begin{bmatrix} m_{i,x} - x \\ m_{i,y} - y \end{bmatrix}, q = \delta^T \delta$$

- Linearized the observation model :

$$H^i = \frac{\partial z_i}{\partial(x, y, \theta, m_{i,x}, m_{i,y})} = \begin{bmatrix} \frac{\partial\sqrt{q}}{\partial x} & \frac{\partial\sqrt{q}}{\partial y} & \cdots \\ \frac{\partial atan2(\delta_x, \delta_y)}{\partial x} & \frac{\partial atan2(\delta_x, \delta_y)}{\partial y} & \cdots \end{bmatrix}$$

$$= \frac{1}{q}\begin{bmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & \sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{bmatrix}$$

某一个mark的坐标

车子状态变数

mark状态变数

$$\boxed{\frac{\partial\sqrt{q}}{\partial x} = \frac{1}{2}\frac{1}{\sqrt{q}}2\delta_x(-1) = \frac{1}{q}(-\sqrt{q}\delta_x)}$$

$$\boxed{\begin{aligned} \frac{\partial}{\partial x}\operatorname{atan2}(y, x) &= \frac{\partial}{\partial x}\arctan\left(\frac{y}{x}\right) = -\frac{y}{x^2+y^2}, \\ \frac{\partial}{\partial y}\operatorname{atan2}(y, x) &= \frac{\partial}{\partial y}\arctan\left(\frac{y}{x}\right) = \frac{x}{x^2+y^2}. \end{aligned}}$$

EKF-SLAM

- Prediction Model
  - $F_t = \begin{bmatrix} 1 & 0 & 0 & 0 \cdots 0 \\ 0 & 1 & 0 & 0 \cdots 0 \\ 0 & 0 & 1 & 0 \cdots 0 \end{bmatrix}^T * F_t^x$, in which $F_t^x = \begin{bmatrix} 1 & 0 & -\frac{v_t}{\omega_t}\cos(\theta) + \frac{v_t}{\omega_t}\cos(\theta + \omega_t \Delta t) \\ 0 & 1 & -\frac{v_t}{\omega_t}\sin(\theta) + \frac{v_t}{\omega_t}\sin(\theta + \omega_t \Delta t) \\ 0 & 0 & 1 \end{bmatrix}$
- Observation Model
  - $$H_t = \begin{array}{c} \begin{bmatrix} 1 & 0 & 0 & 0 \cdots 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 1 & 0 & 0 \cdots 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 1 & 0 \cdots 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & 0 \cdots 0 & 1 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & 0 \cdots 0 & 0 & 1 & 0 \cdots 0 \end{bmatrix}^T \\ \begin{array}{cccccccc} x & y & \theta & & m_{i,x} & m_{i,y} & \end{array} \end{array} * H_t^i$$

Extended Kalman-Filter

$x_k^{pre} = f(x_k^{est}, u_k)$

$P_k^{pre} = F_k P_{k-1}^{est} F_k^T + Q$

$K_k = P_k^{pre} H^T \left(H P_k^{pre} H^T + R\right)^{-1}$

$x_k^{est} = x_k^{pre} + K_k\left(z_k - H x_k^{pre}\right)$

$P_k^{est} = (I - K_k H) P_k^{pre}$

, in which $H_t^i = \frac{1}{q}\begin{bmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & \sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{bmatrix}, \delta = \begin{bmatrix} m_{i,x} - x \\ m_{i,y} - y \end{bmatrix}, q = \delta^T \delta$