

Robotic Navigation and Exploration

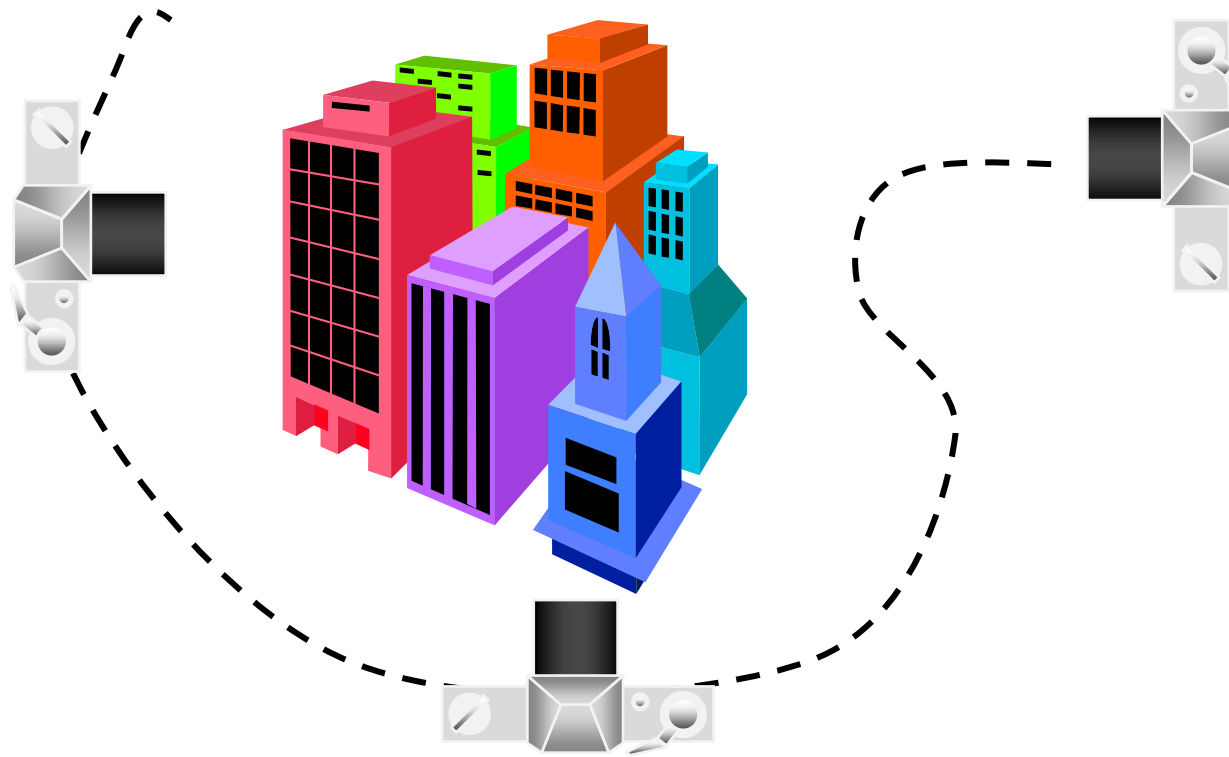
Week 4: SLAM Back-end (I)

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

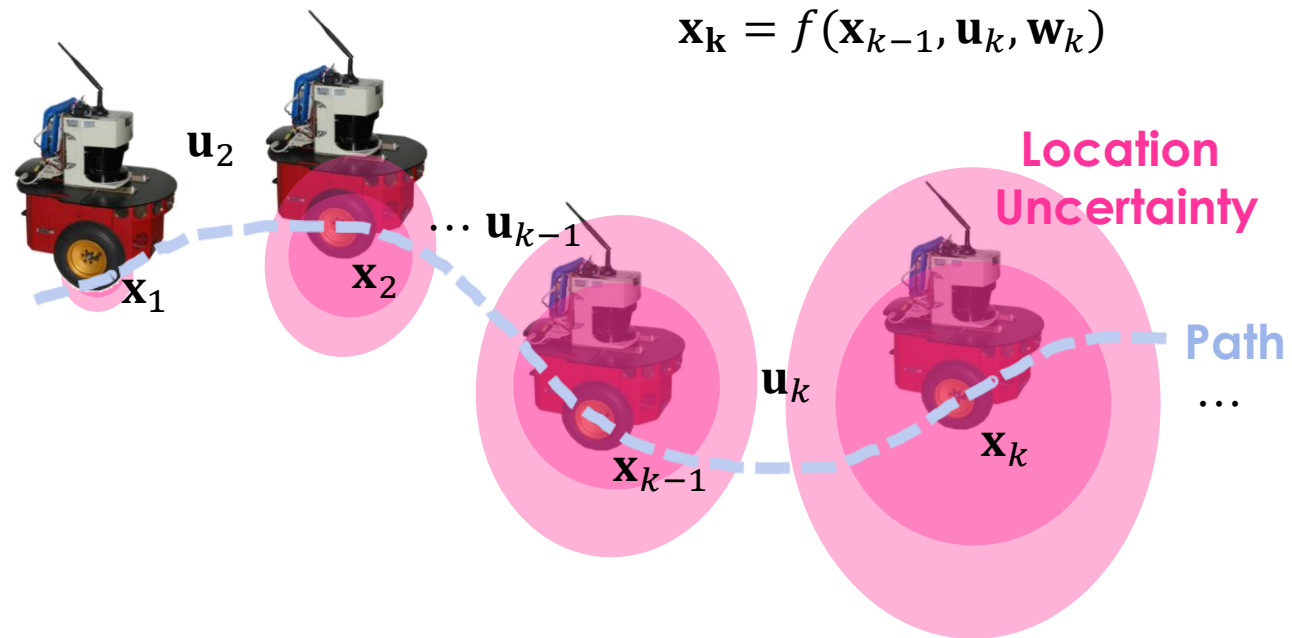
Outline

- 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

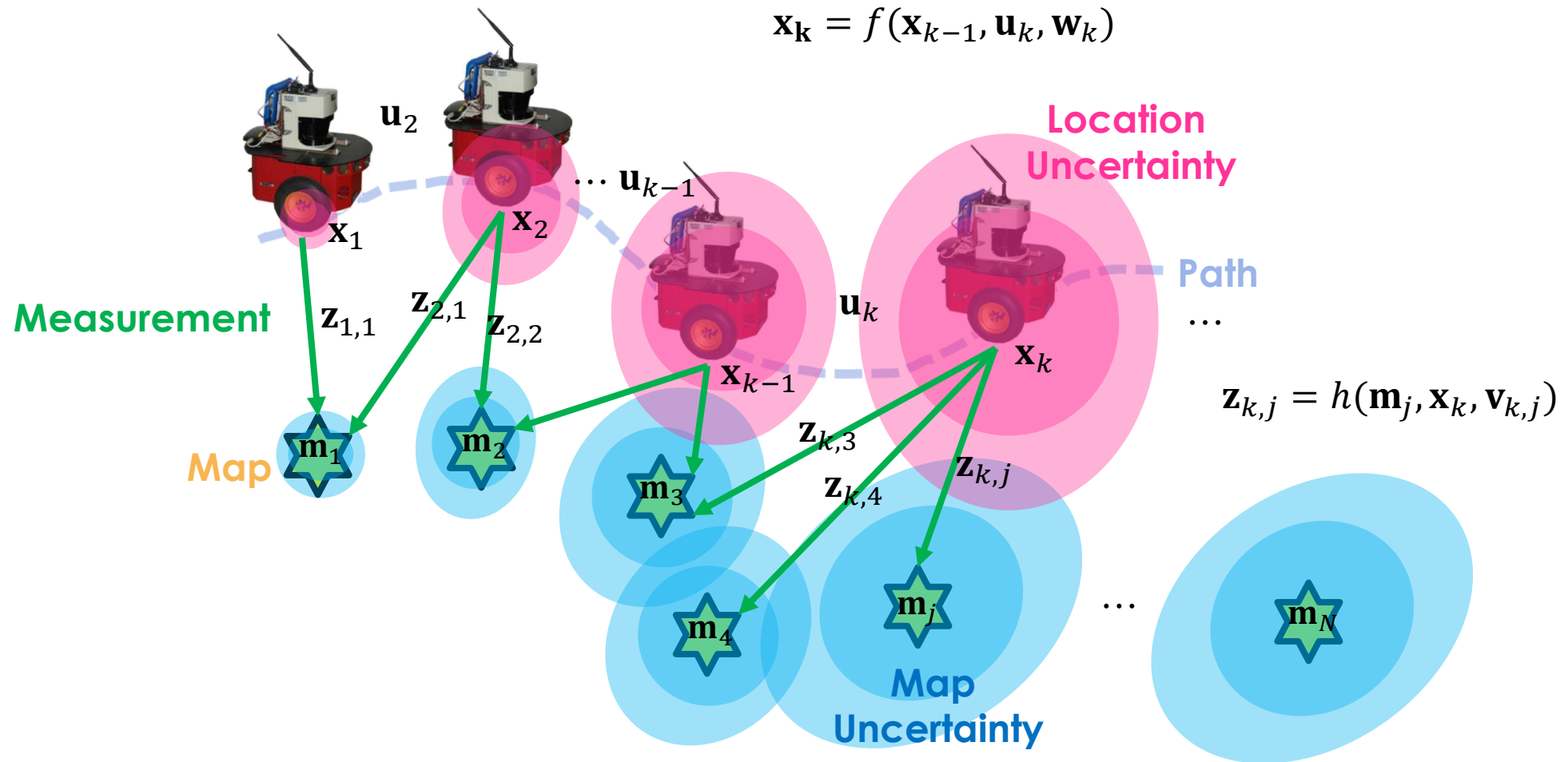
SLAM Problem



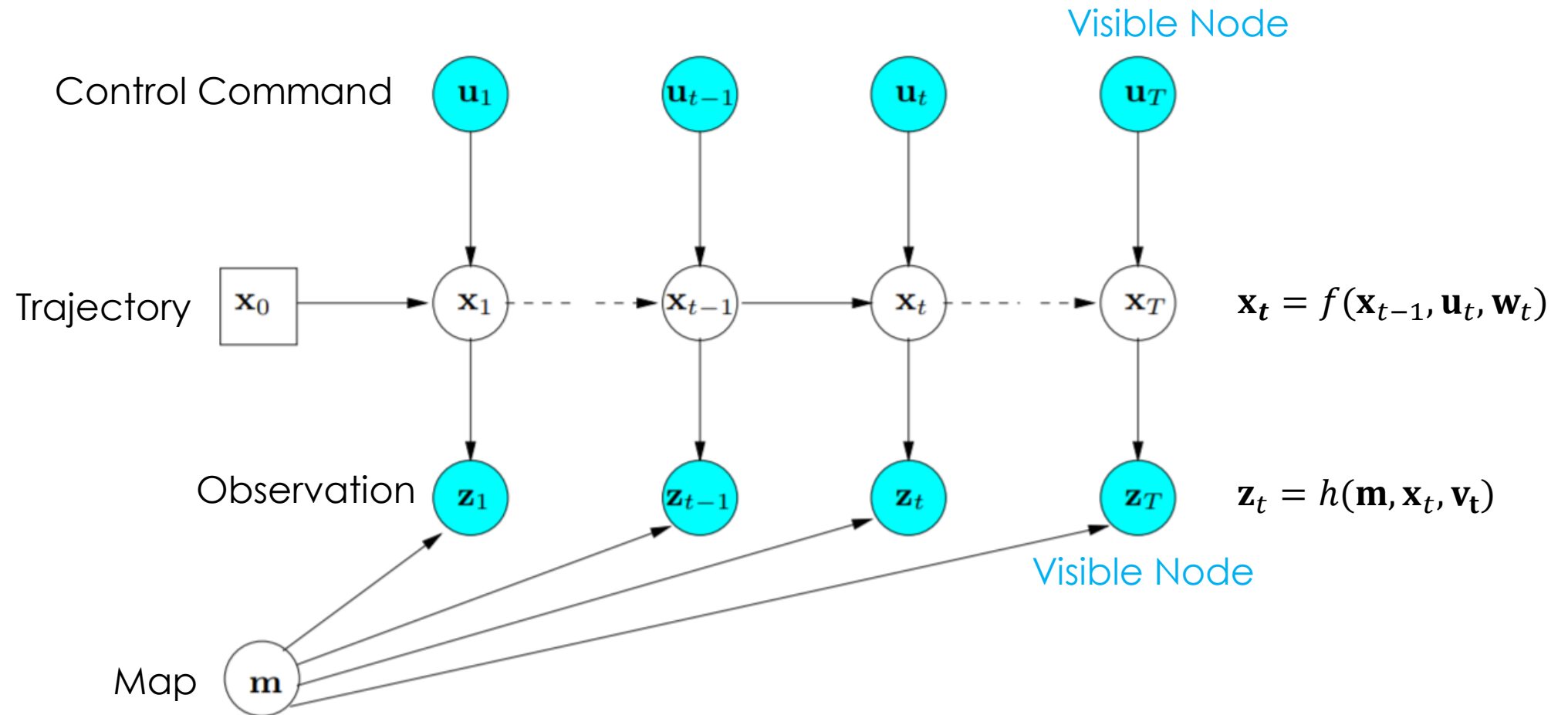
SLAM Problem



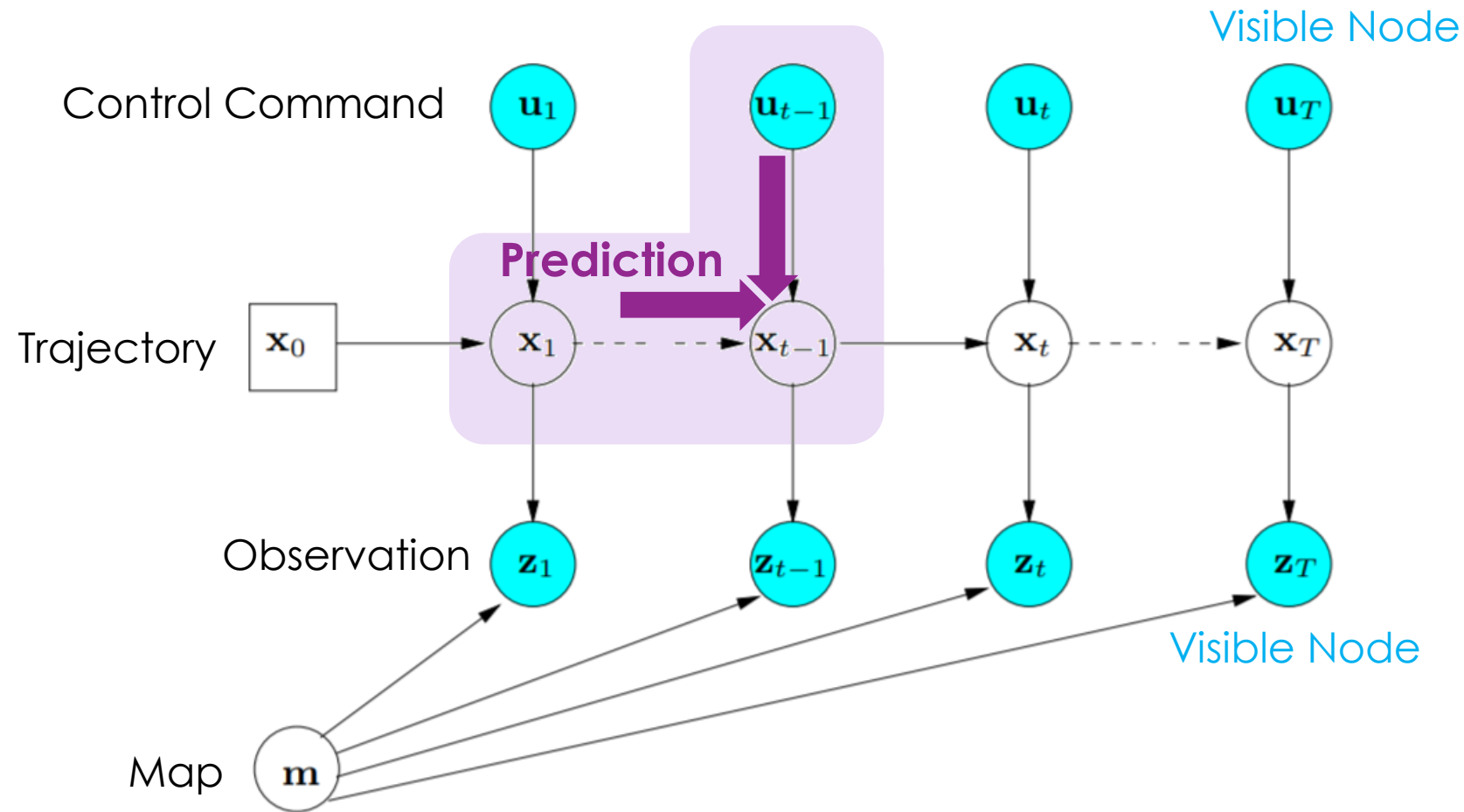
SLAM Problem



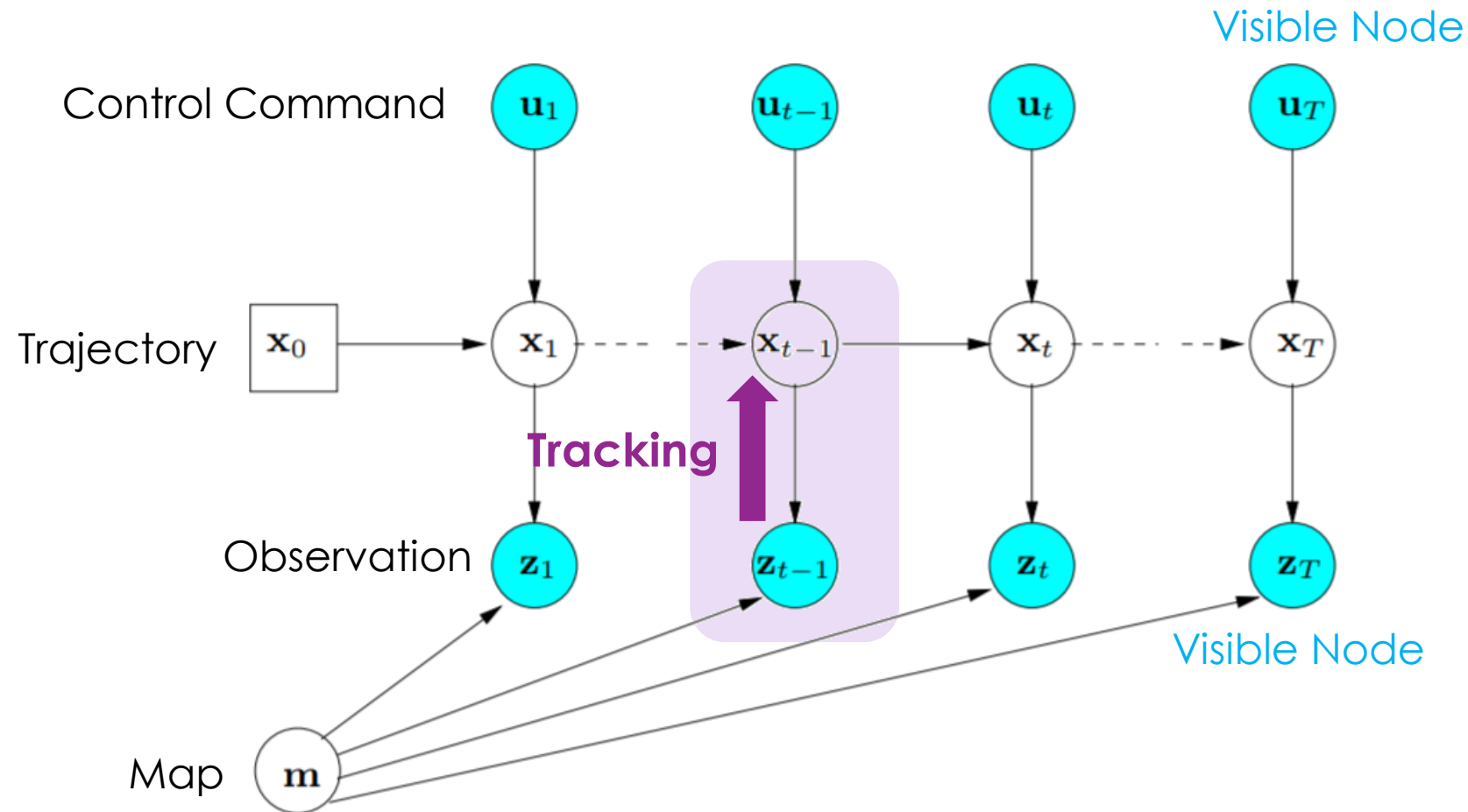
Probability Graphical Model for SLAM Problem



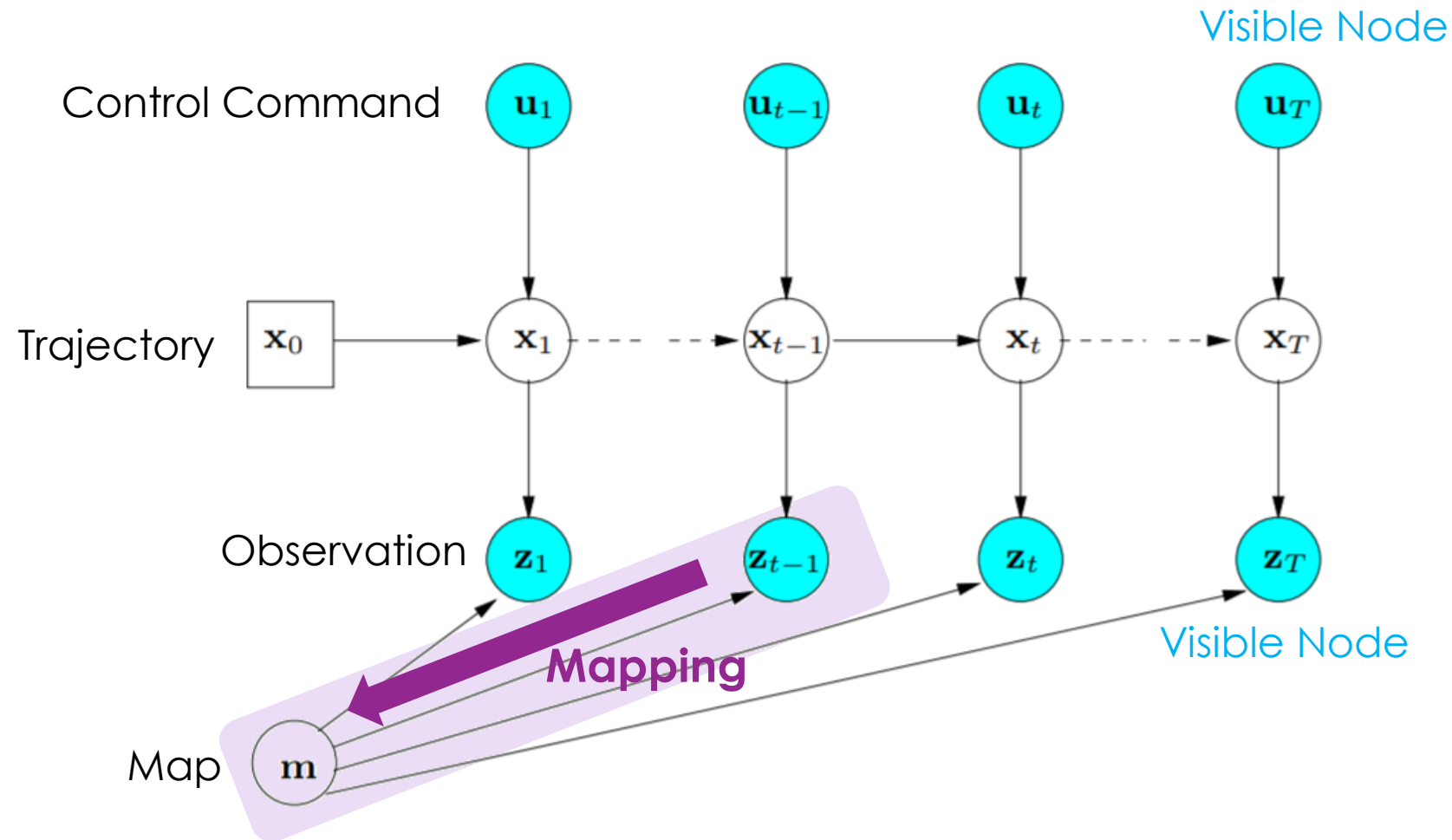
Probability Graphical Model for SLAM Problem



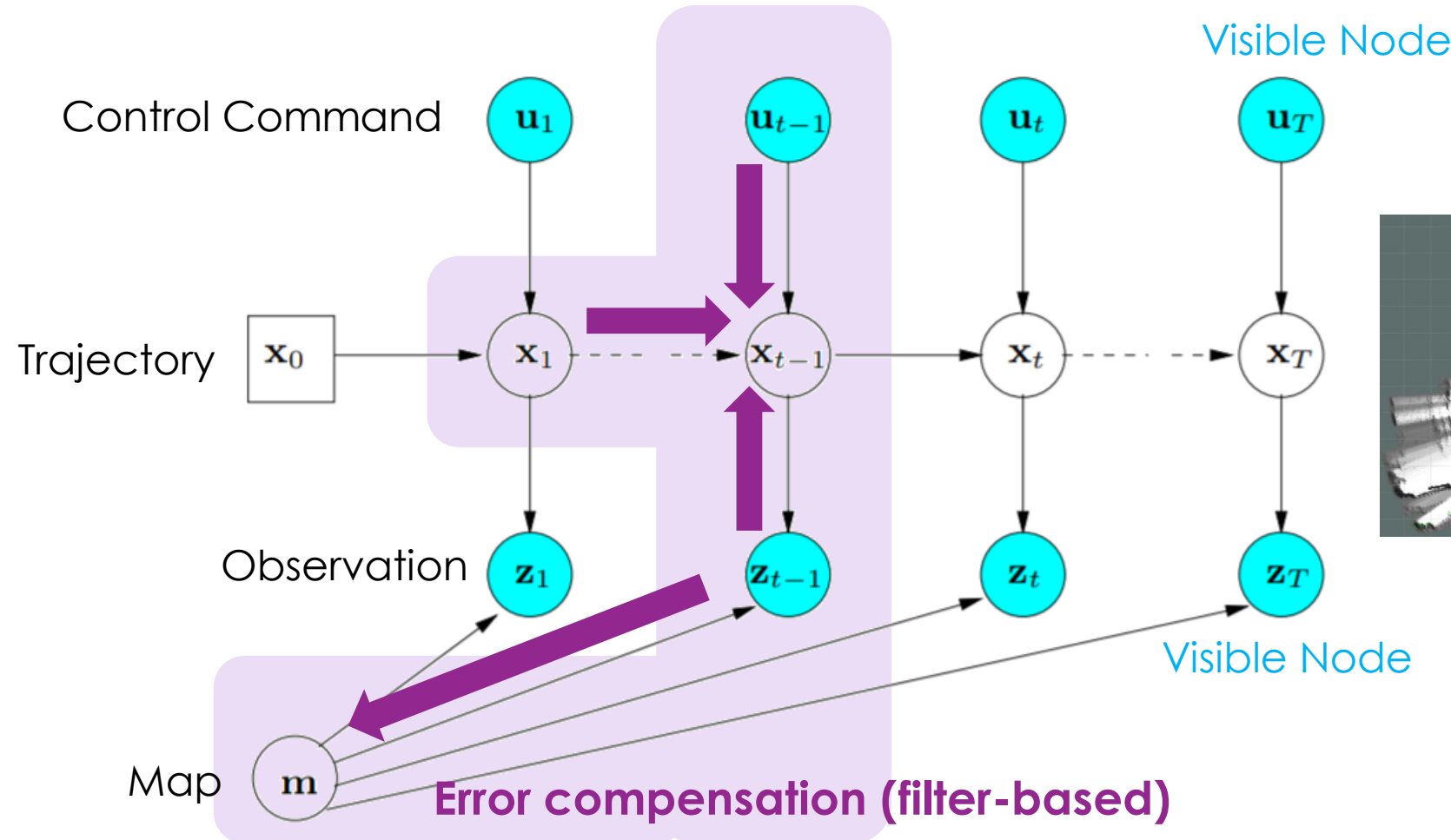
Probability Graphical Model for SLAM Problem



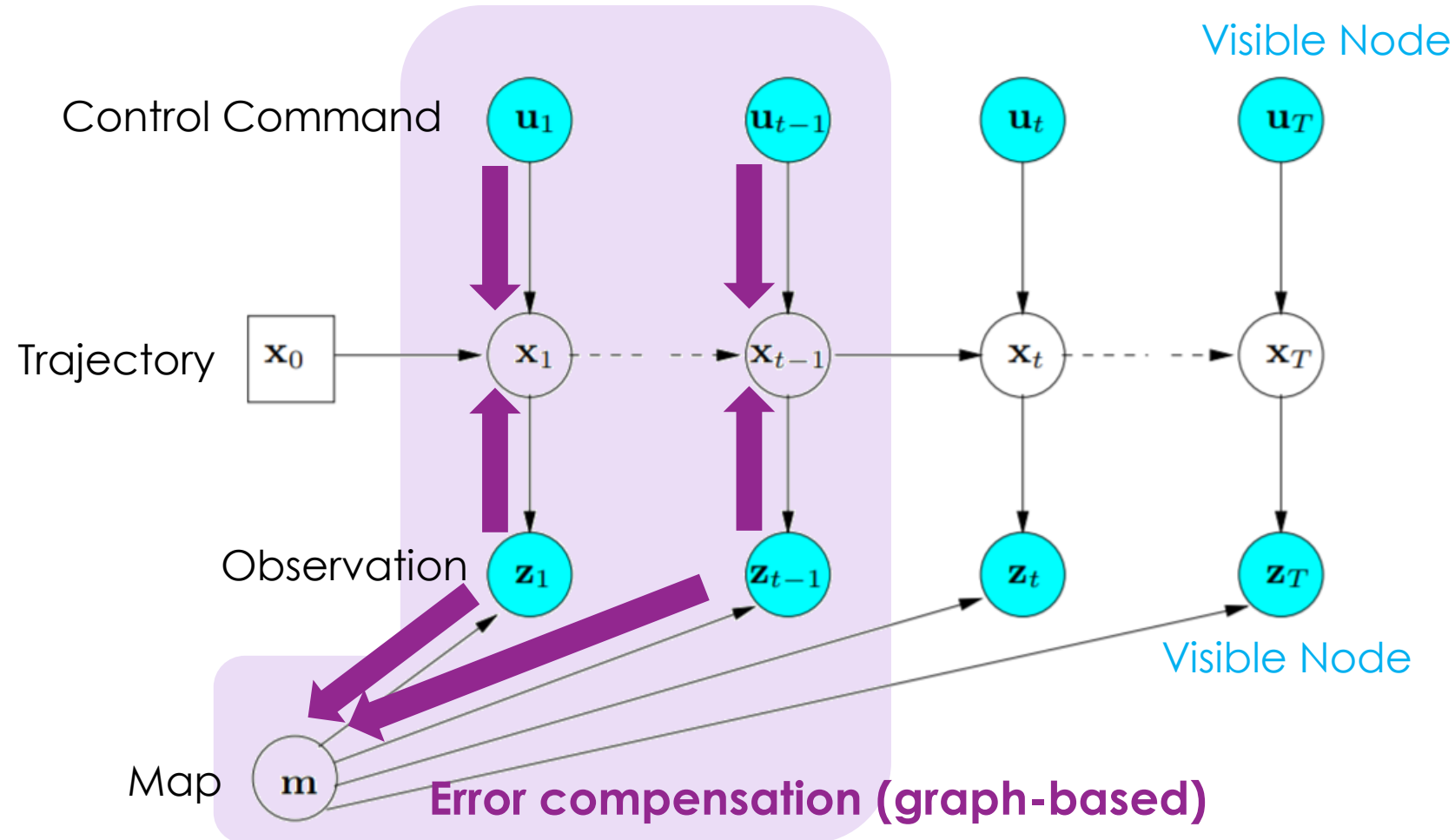
Probability Graphical Model for SLAM Problem



Probability Graphical Model for SLAM Problem



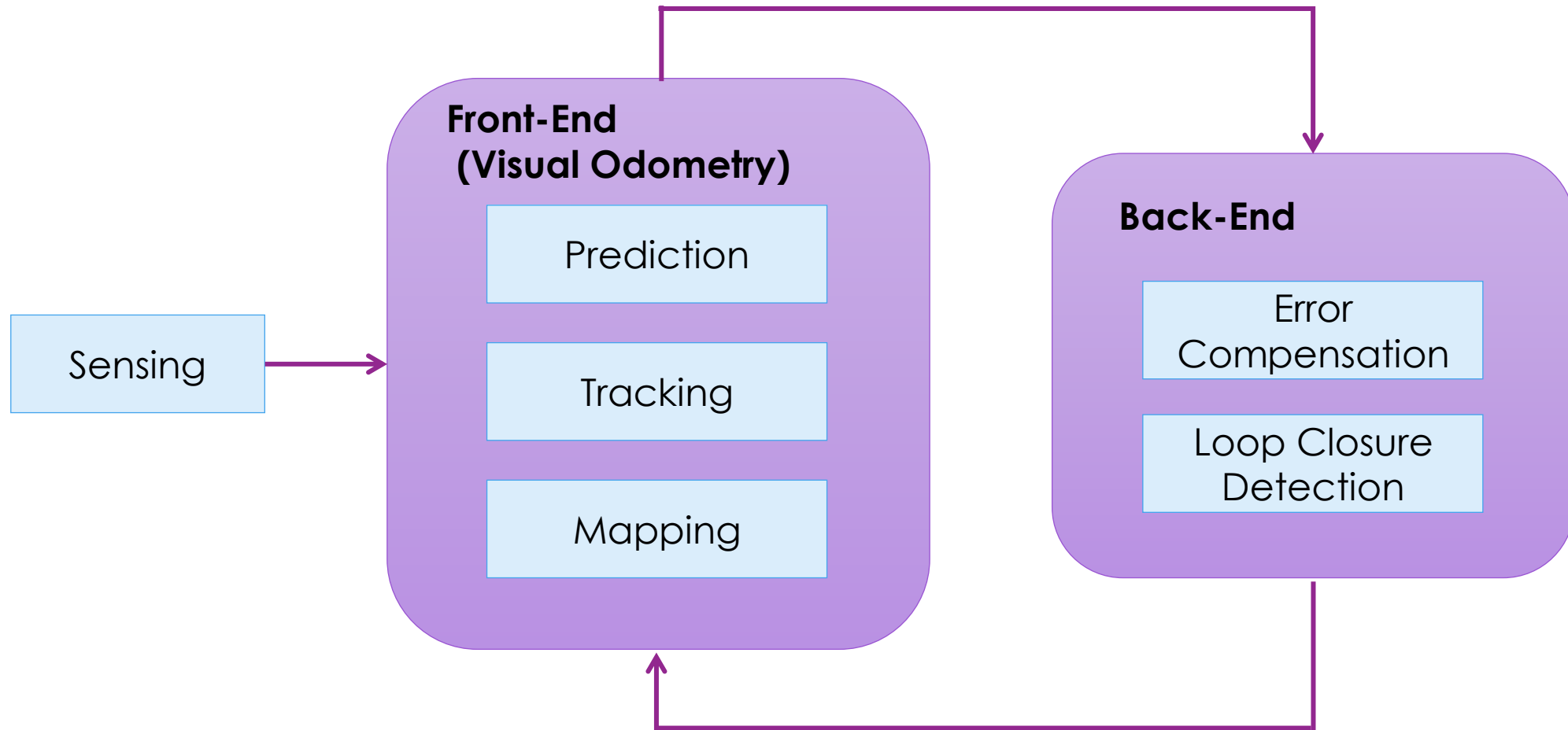
Probability Graphical Model for SLAM Problem



Error Compensation Methods

- Filter-based
 - Small Computation
 - On-line Optimization
- Graph-based
 - Large computation
 - High Accuracy
 - Off-line Optimization

SLAM Architecture

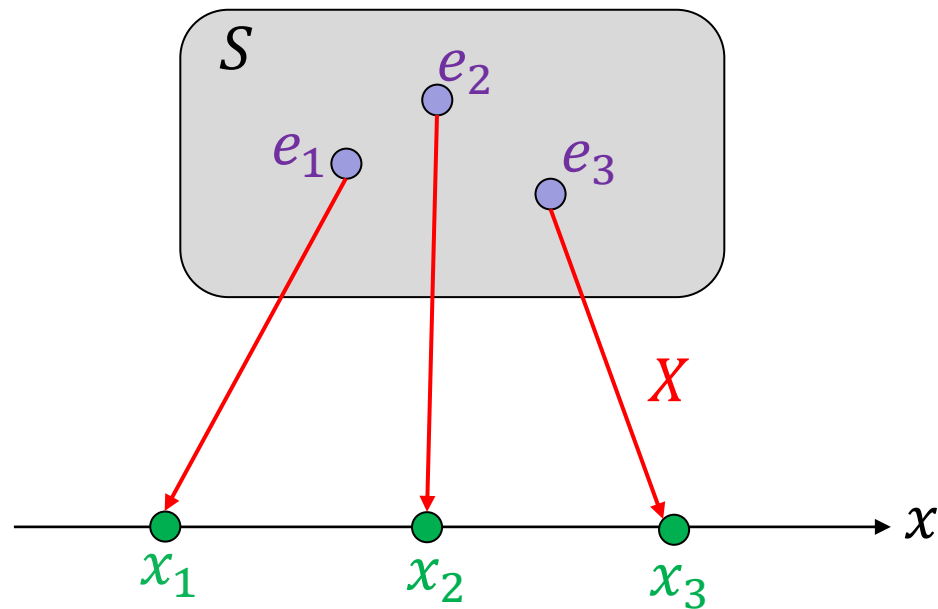


Outline

- 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

- 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 **red** ball

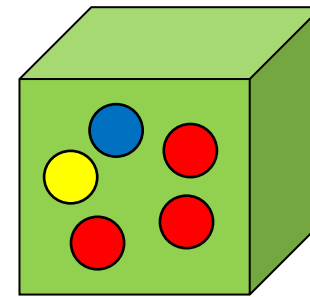
X = 2, if choose the **yellow** ball

X = 3, if choose the **blue** ball

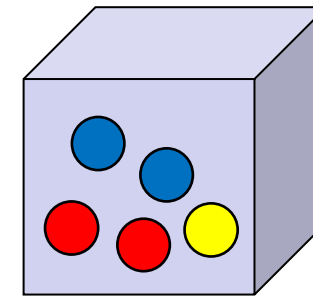
Y: The id of the box

Y = 1, if choose the **green** box

Y = 2, if choose the **purple** box

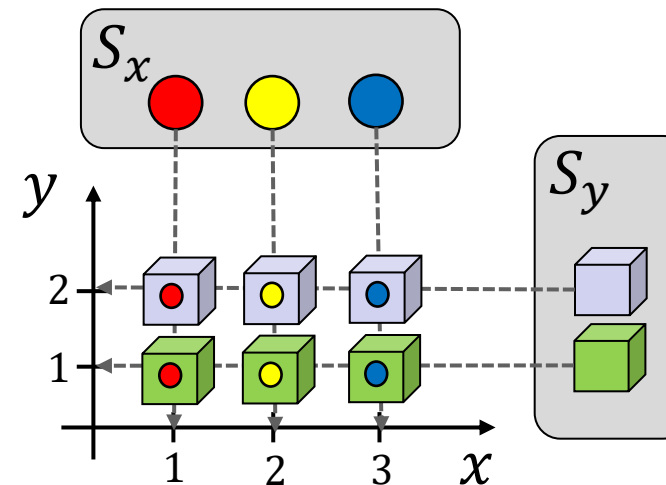


Y = 1



Y = 2

● X = 1 ● X = 2 ● X = 3



Different Types of Probability

- Joint Probability

$$\mathbf{P(X,Y)}$$

- Condition Probability

$$\mathbf{P(X|Y), P(Y|X)}$$

- Marginal Probability

$$\mathbf{P(X), P(Y)}$$

Sum / Product Rule

- Sum Rule

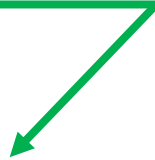
$$\boxed{P(X = x_i)} = \sum_Y P(X, Y)$$

Marginal Probability

- Product Rule

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

Joint Probability

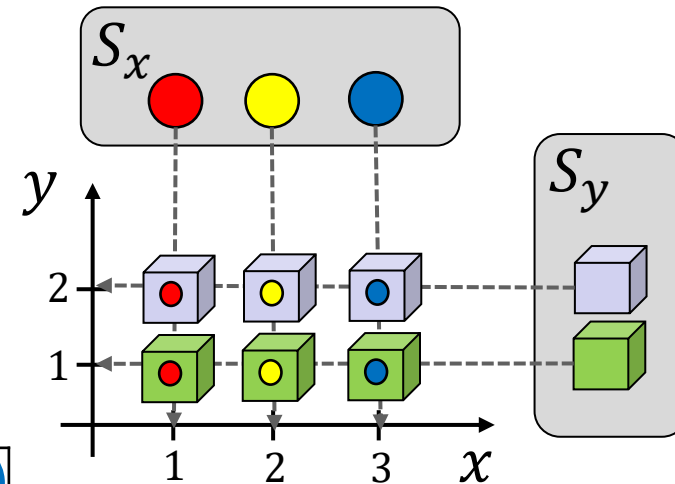
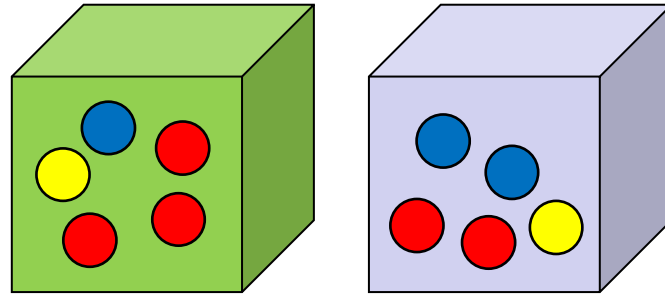


- Bayes Theorem

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

Sum / Product Rule Example

- Joint Probability and Marginal Probability



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

- Independent Event

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

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

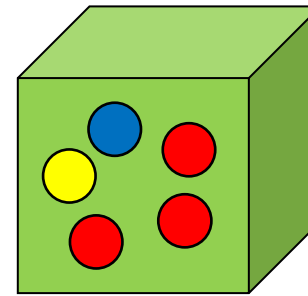
$$P(Y = 1) = P(Y = 1|X = 2)$$

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

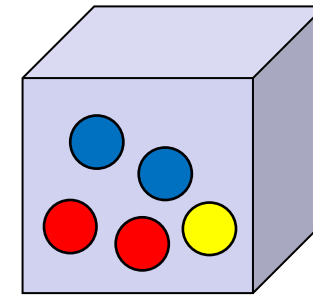
- Independent Random Variable

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

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

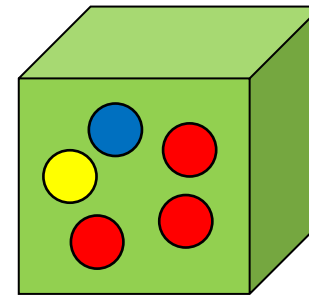


$Y = 1$

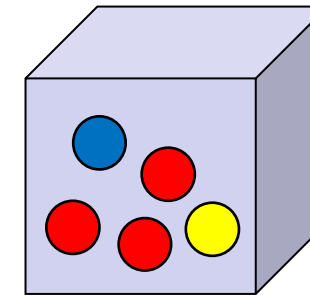


$Y = 2$

● $X = 1$ ● $X = 2$ ● $X = 3$



$Y = 1$

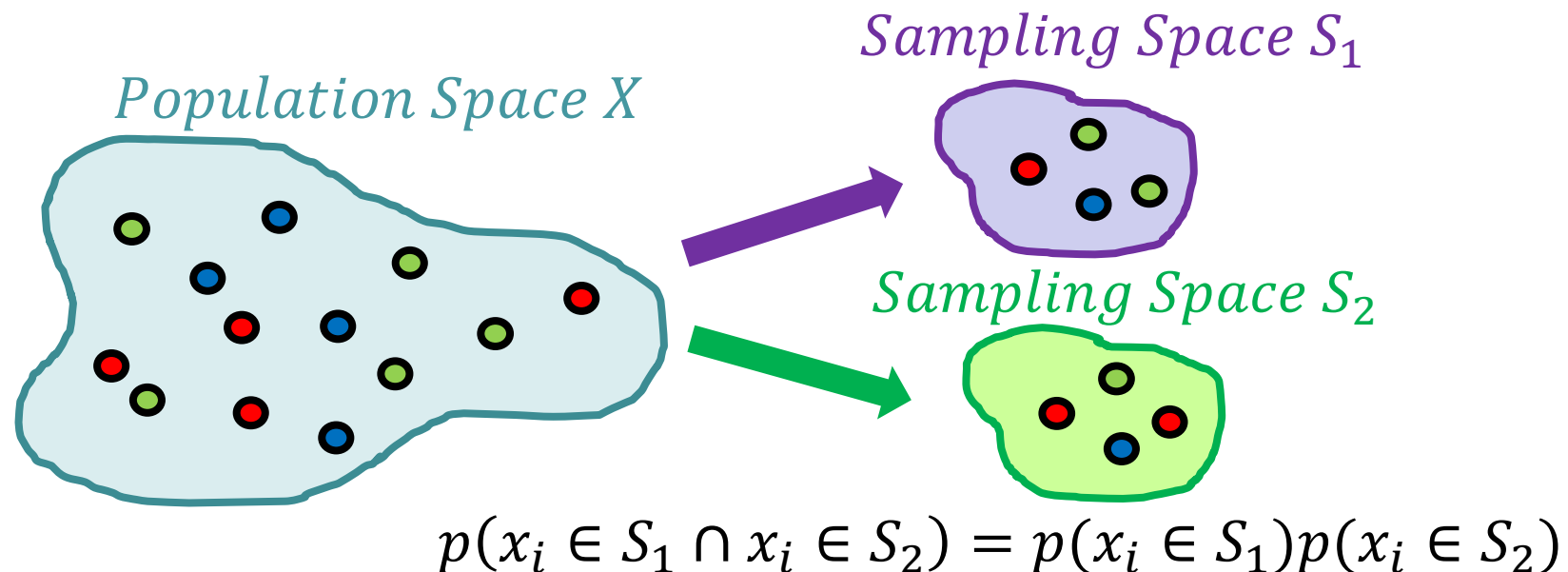


$Y = 2$

● $X = 1$ ● $X = 2$ ● $X = 3$

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

- 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

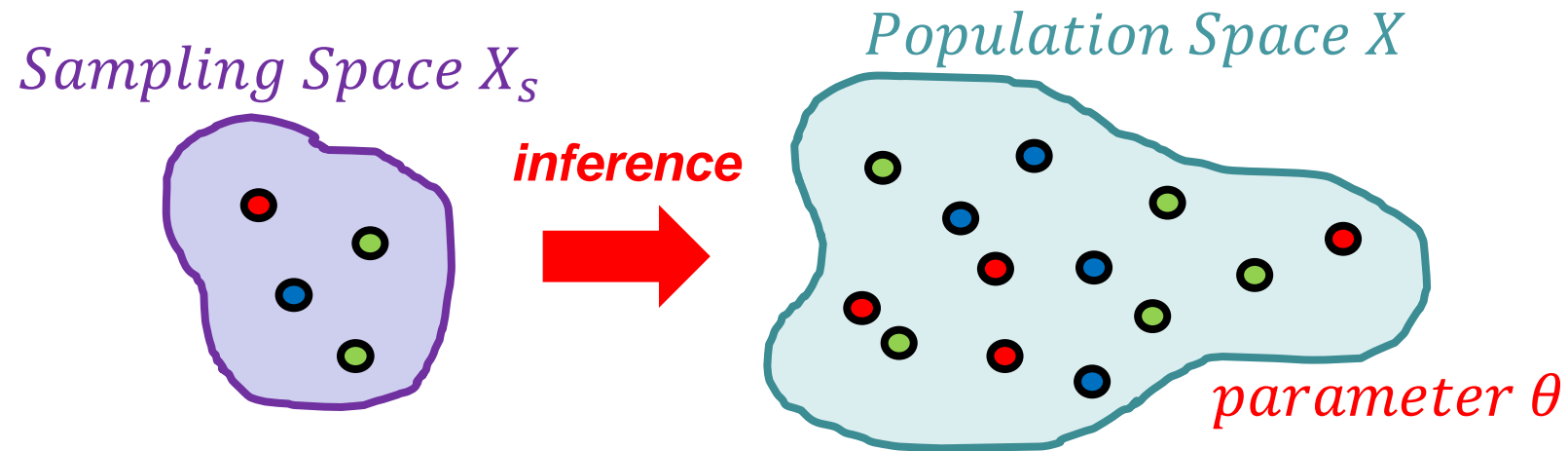


Inference

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

Statistical Inference

- A process to inference the parameters of population based on the information of sampling data
- x_s is sampled data, θ is the parameter of distribution over population, statistical inference is to inference $p(\theta | x_s)$

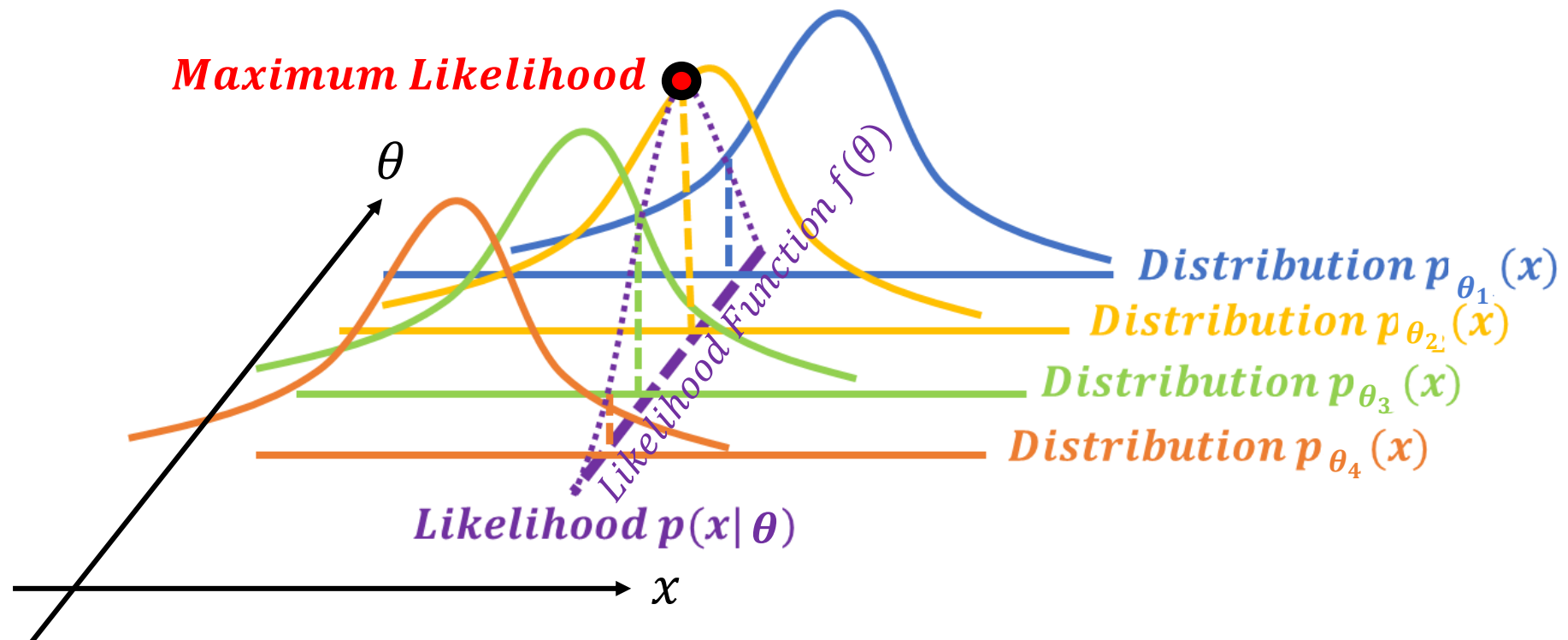


Statistical Inference

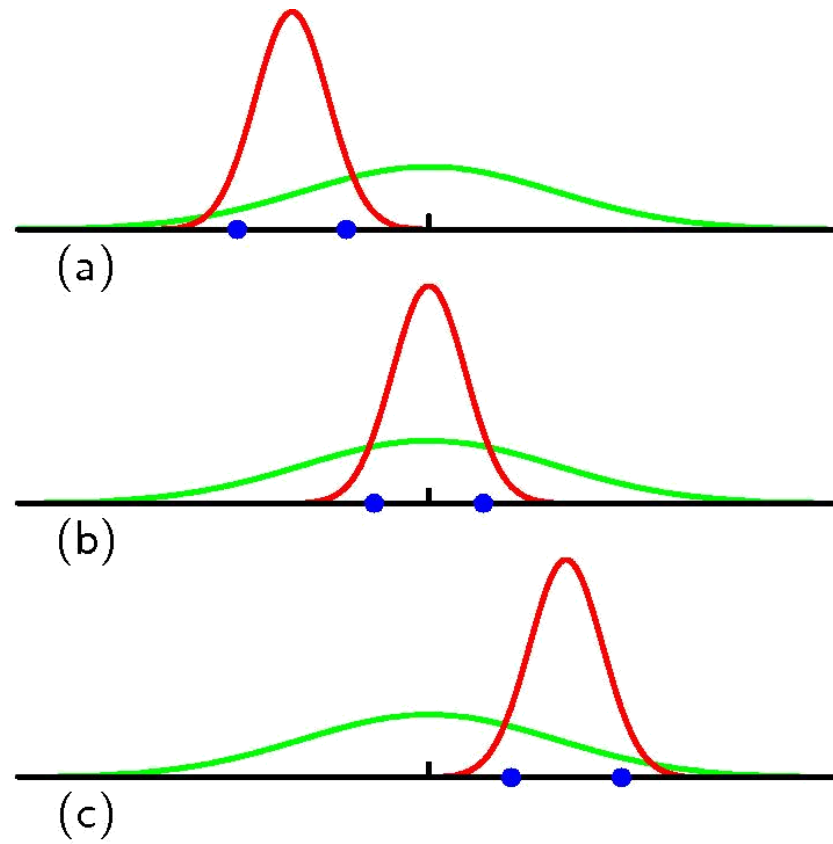
- Two approaches of statistical inference
 - Hypothesis Testing (Top-Down)
 - Given a hypothesis of parameters, evaluate the correctness from sampling data
 - Estimation (Bottom-Up)
 - Find the most likely parameters from sampling data

Maximum Likelihood Estimation (MLE)

- Visualization of likelihood function



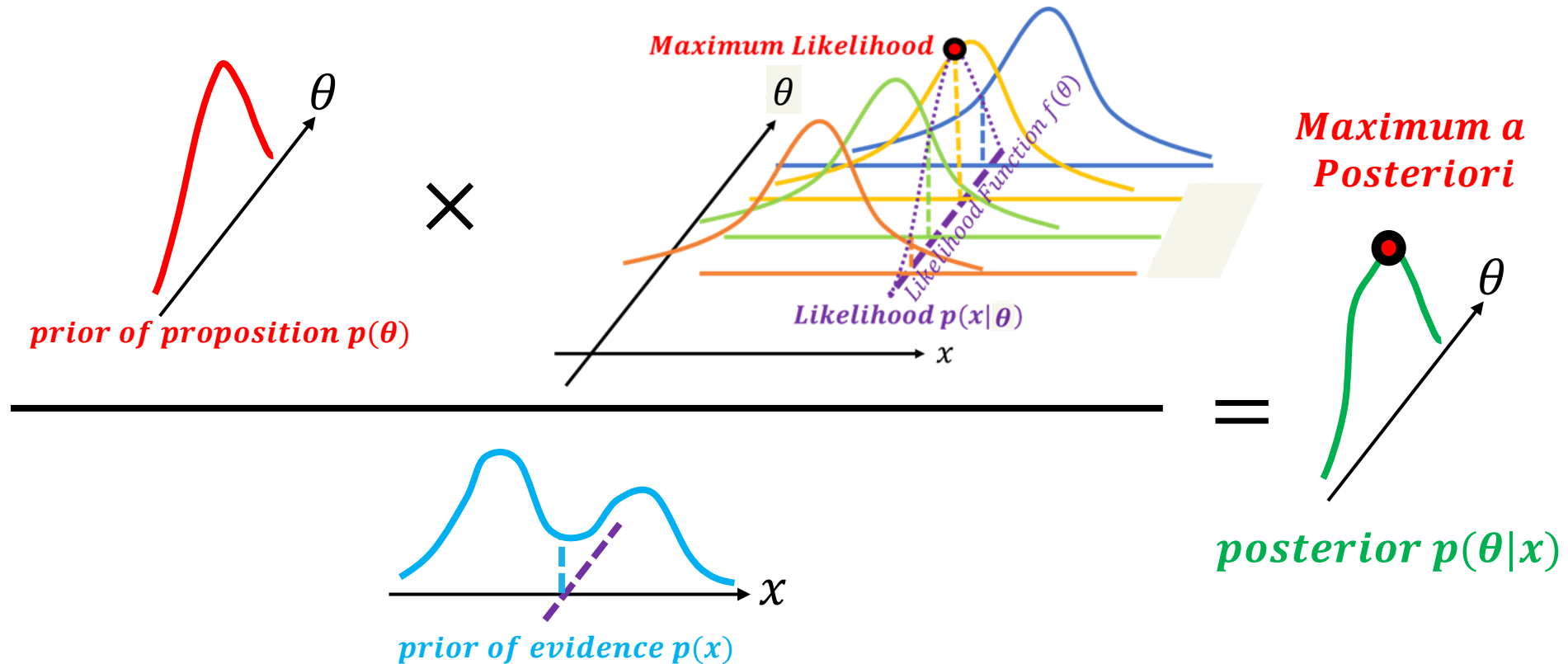
Problem of MLE



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 $P(\mathbf{x} | \theta)$:
 - *Bernoulli distribution*: $\theta^n(1 - \theta)^{m-n}$
 - *MLE Estimation*:

$$\rightarrow \max_p \theta(1 - \theta)^4$$

$$\theta = 0.2$$

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

Example: Coin Estimation

- MAP Estimation (Assume Discrete Uniform Prior)

Prior
(Discrete Uniform)

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

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

$$\begin{array}{l} \theta = 0.0 \\ \theta = 0.1 \\ \theta = 0.2 \\ \vdots \end{array} \begin{bmatrix} 1/11 \\ 1/11 \\ 1/11 \\ \vdots \end{bmatrix} \times \begin{bmatrix} (0)^1(1)^4 \\ (0.1)^1(0.9)^4 \\ (0.2)^1(0.8)^4 \\ \vdots \end{bmatrix}$$

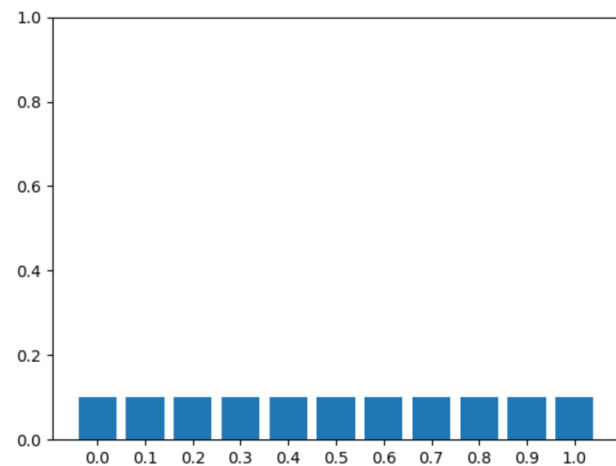
$$p(x) = \sum_{\theta} p(x, \theta) = \sum_{\theta} p(\theta)p(x|\theta)$$

$$= \begin{bmatrix} 0.000 \\ 0.213 \\ 0.333 \\ \vdots \end{bmatrix}$$

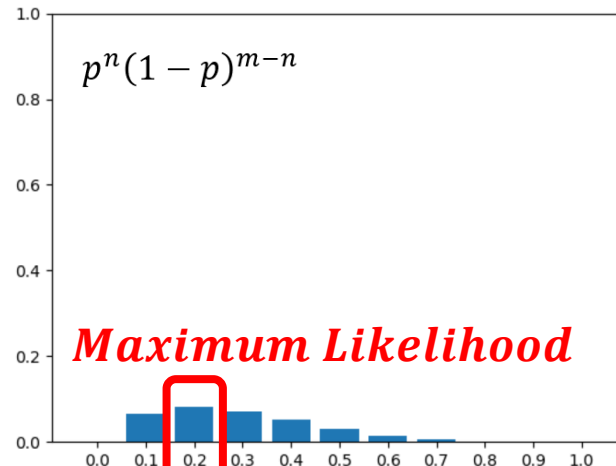
Marginal Probability

Example: Coin Estimation

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

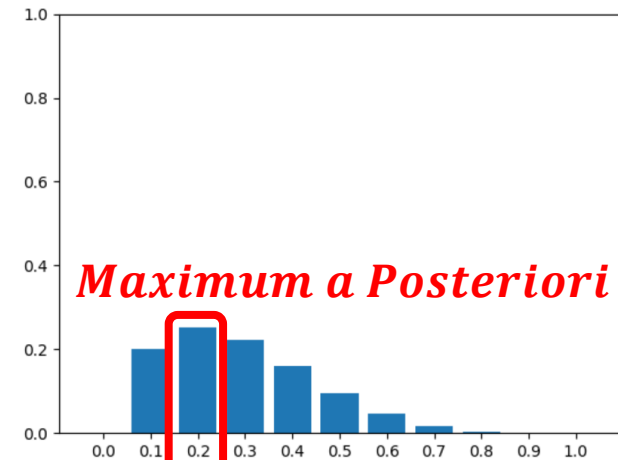


Prior: $P(\theta)$



Maximum Likelihood

Likelihood: $P(x|\theta)$



Maximum a Posteriori

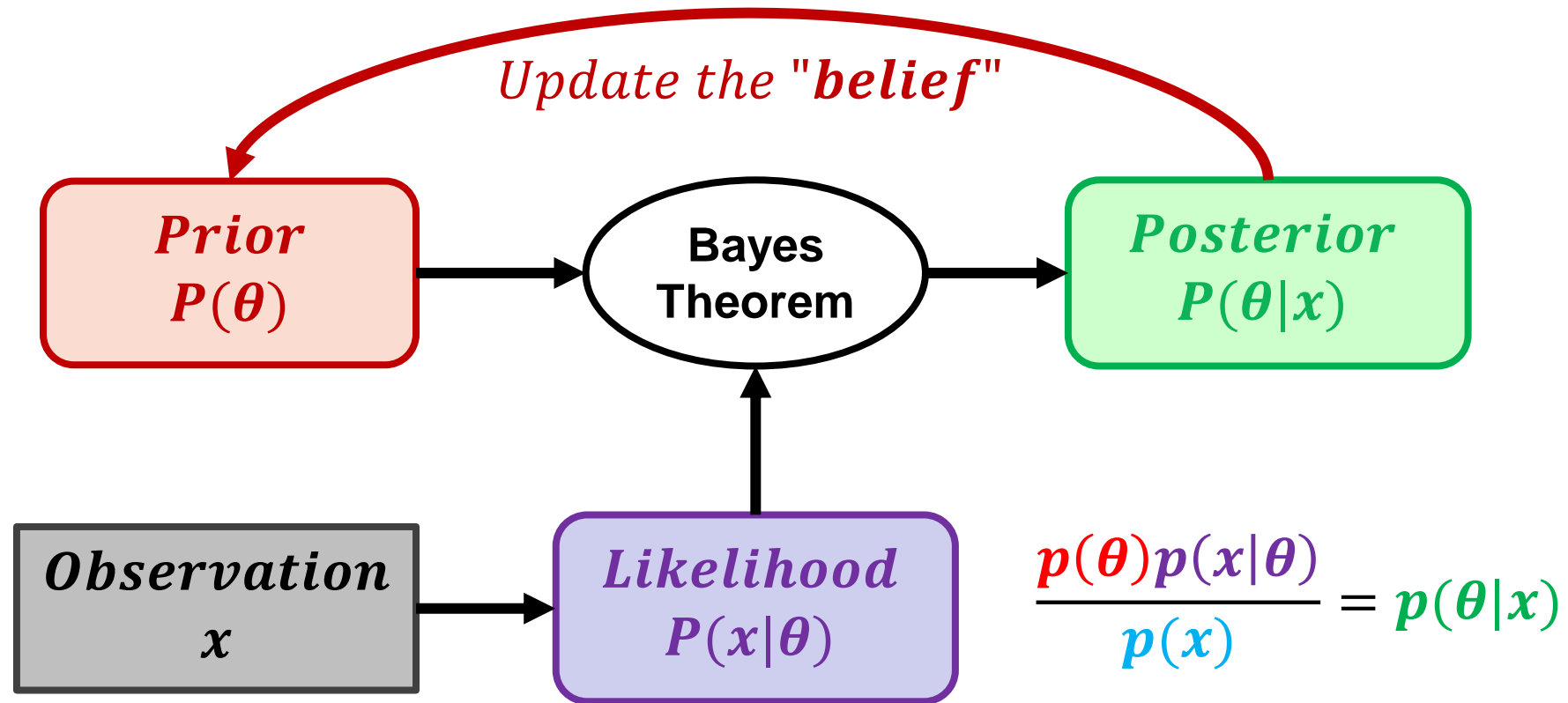
Posterior: $P(\theta|x)$

Bayesian Probability

- Classical Probability View
 - Model parameters have a **certain** value.
 - The goal of learning is to **infer the parameters** from sampling data which we call “**Estimation**”.
- Bayesian Probability View
 - Model parameters have **uncertainty**.
 - The goal of learning is to infer the probability over every possible parameters, or **infer the hyper-parameters**.

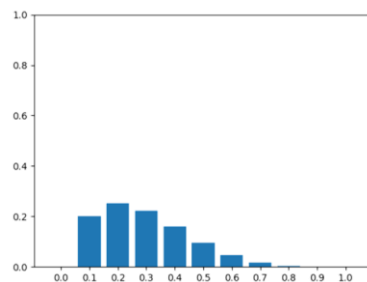
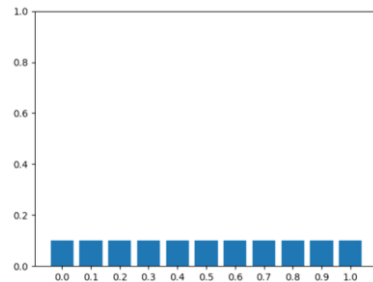
Bayesian Approach

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

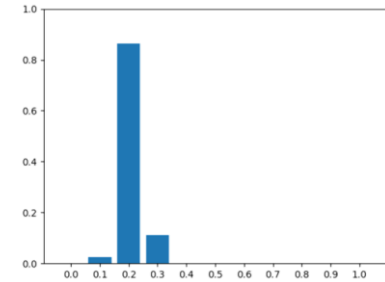
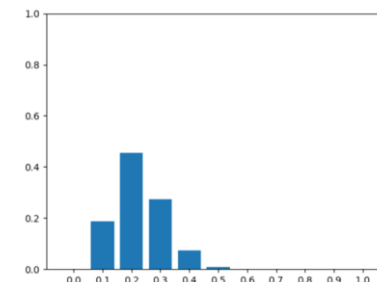


Bayesian Approach (Tossing Coins Example)

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



...



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

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

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

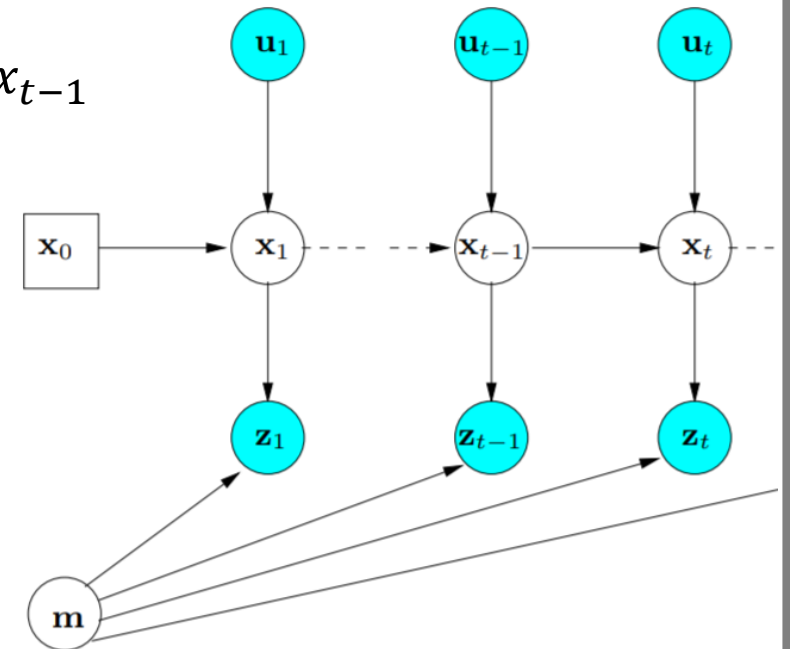
$bel(\theta)$

State Prediction:

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

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

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



Bayes Filter

$\overline{bel}(\theta)$

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

$bel(\theta)$

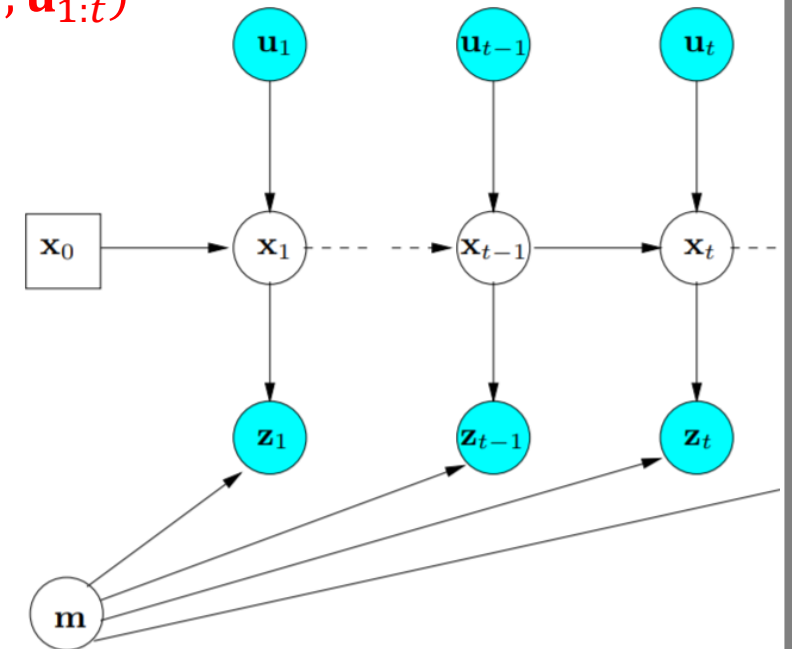
Measurement Update:

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

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


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

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



Bayes Filter

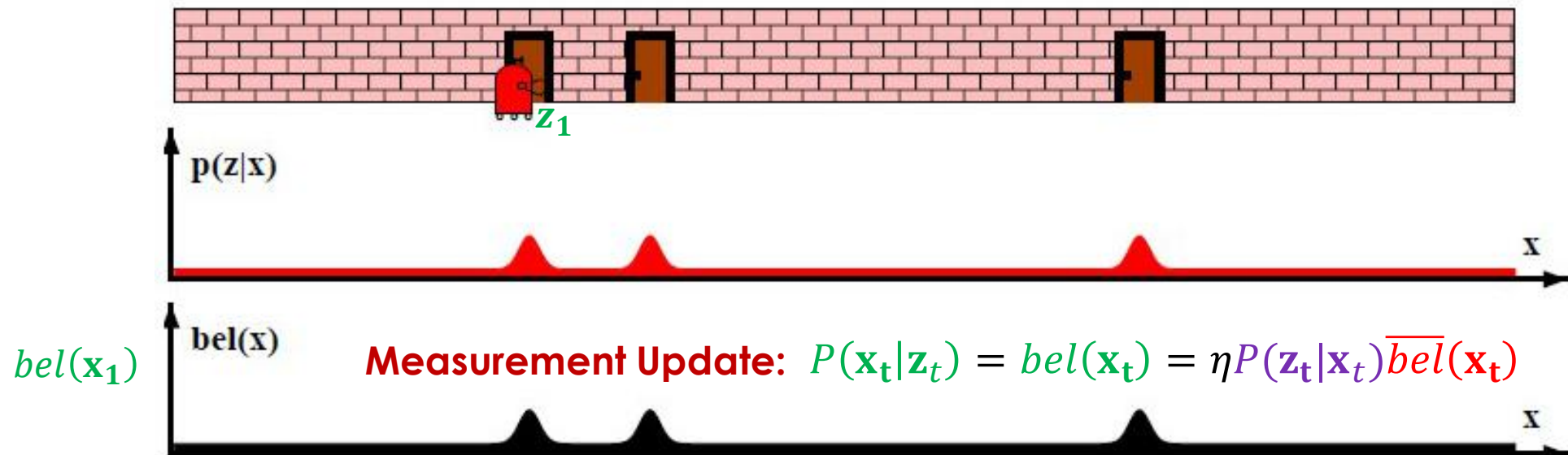
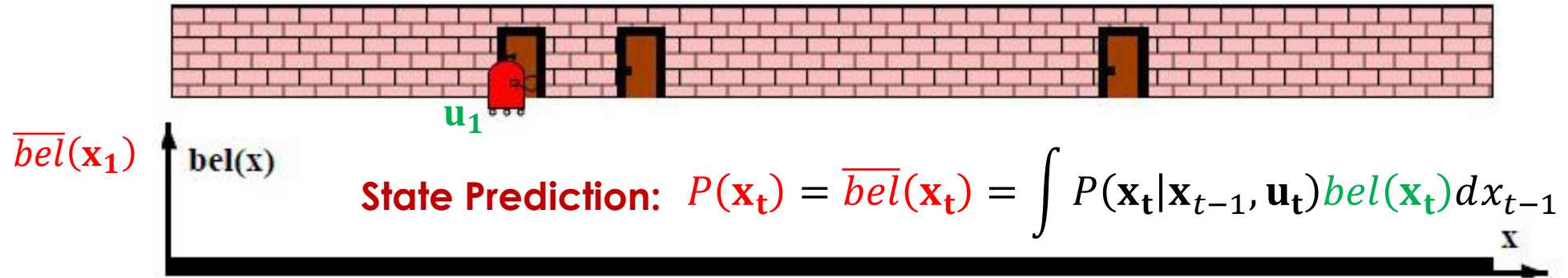
State Prediction: $P(\mathbf{x}_t) = \overline{bel}(\mathbf{x}_t) = \int P(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) bel(\mathbf{x}_t) d\mathbf{x}_{t-1}$



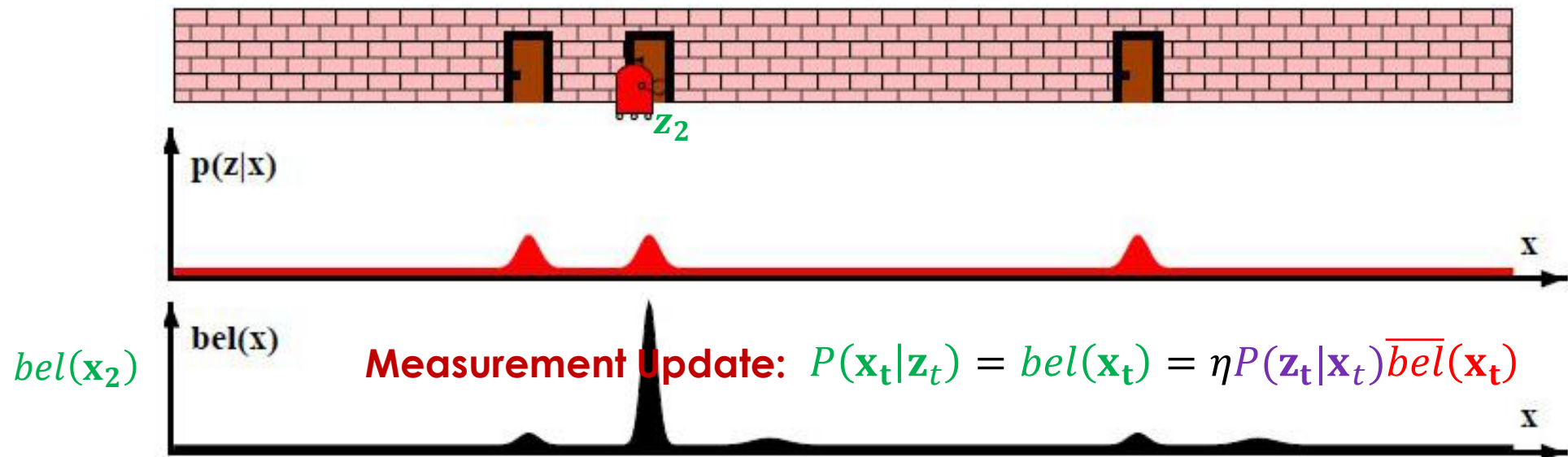
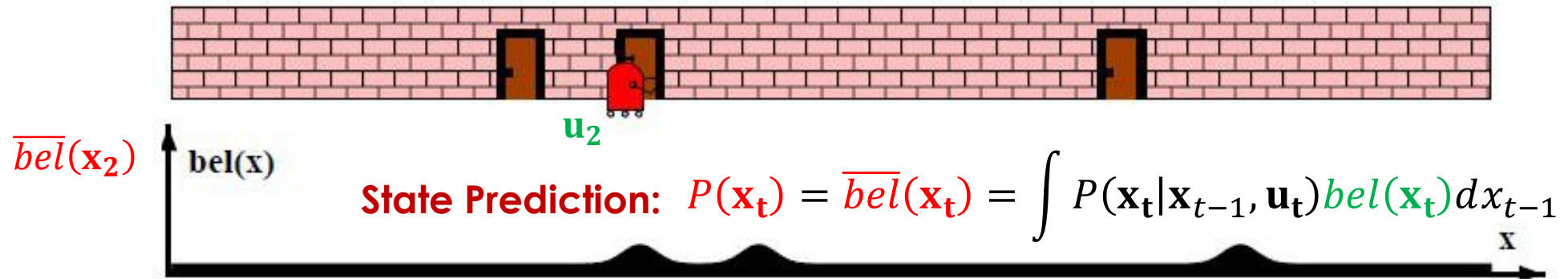
Measurement Update: $P(\mathbf{x}_t | \mathbf{z}_t) = bel(\mathbf{x}_t) = \eta P(\mathbf{z}_t | \mathbf{x}_t) \overline{bel}(\mathbf{x}_t)$

```
1:  Algorithm Bayes_filter( $bel(x_{t-1}), u_t, z_t$ ):  
2:    for all  $x_t$  do  
3:       $\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx$   
4:       $bel(x_t) = \eta p(z_t | x_t) \overline{bel}(x_t)$   
5:    endfor  
6:    return  $bel(x_t)$ 
```

Localization



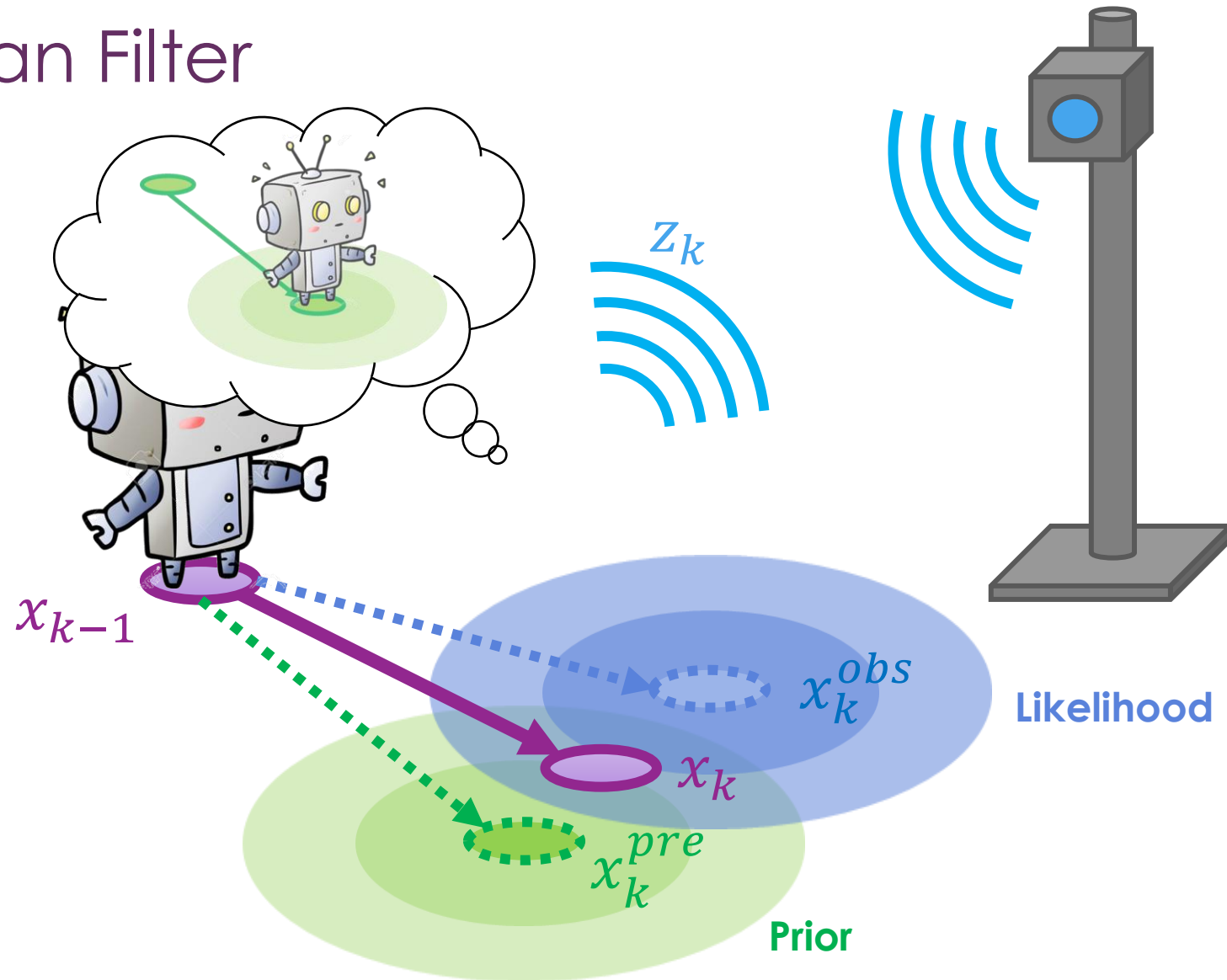
Localization



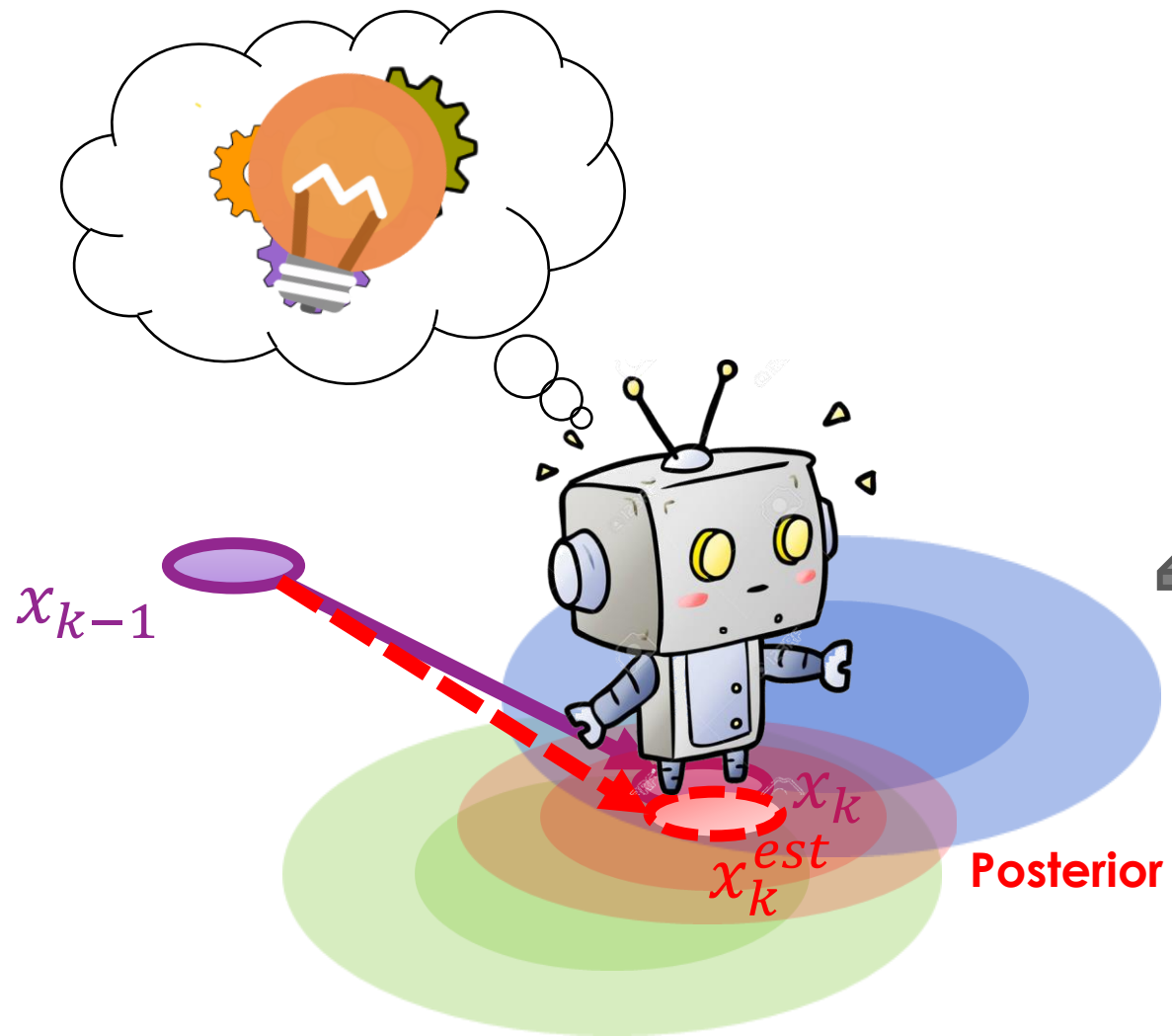
Outline

- 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

Kalman Filter



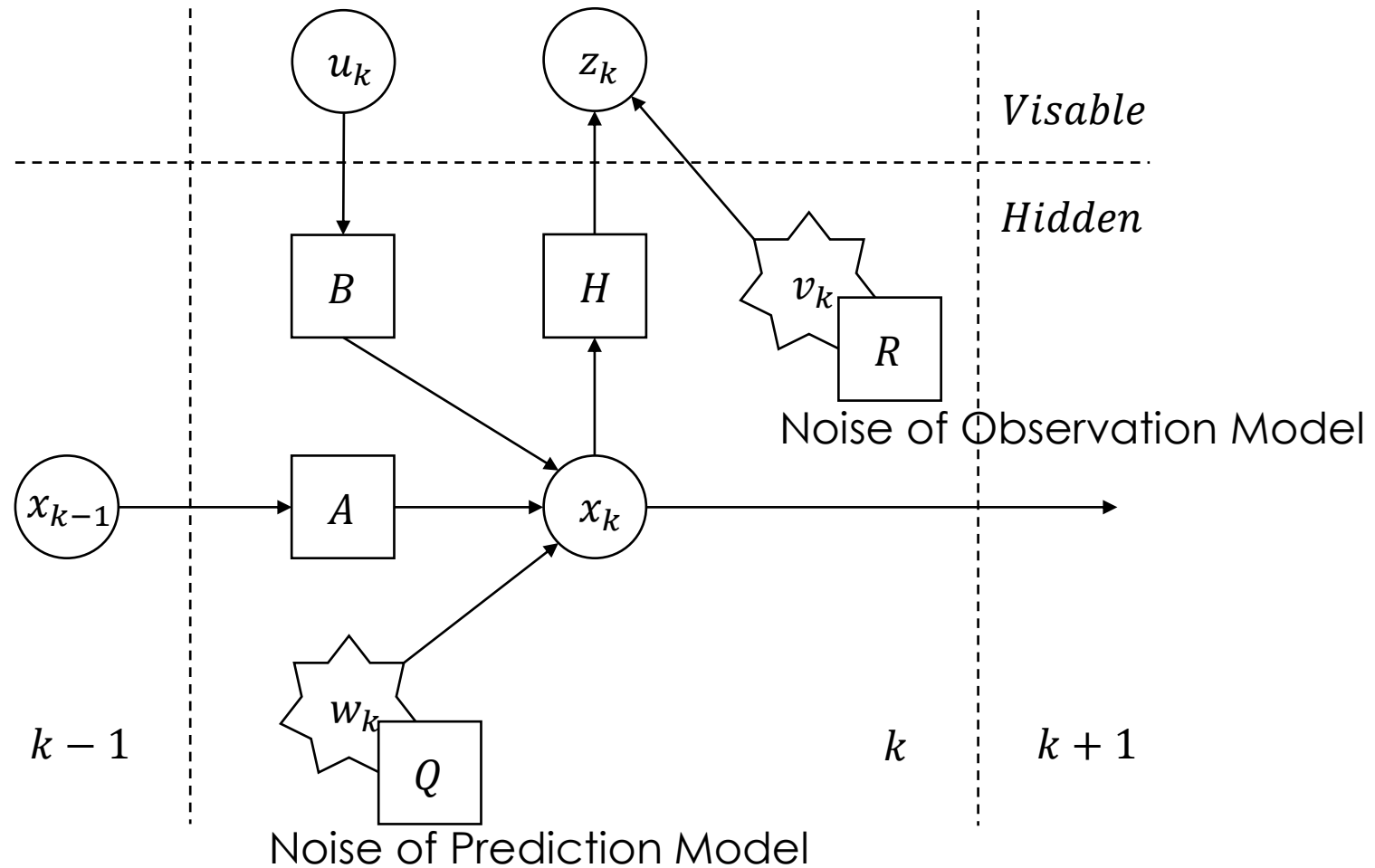
Kalman Filter



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

- Prediction

- ✓ State: x_k^{pre}

- ✓ Error: $e_k^{pre} = x_k - x_k^{pre}$,

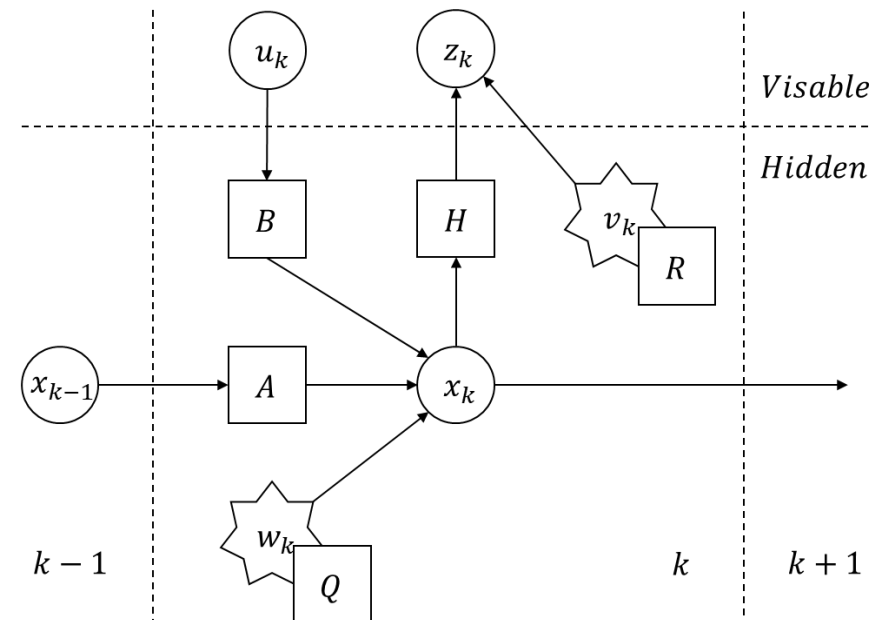
- ✓ Covariance: $P_k^{pre} = E[e_k^{pre} e_k^{preT}]$

- Estimation

- ✓ State: x_k^{est}

- ✓ Error: $e_k^{est} = x_k - x_k^{est}$

- ✓ Covariance: $P_k^{est} = E[e_k^{est} e_k^{estT}]$



Kalman Filter

- The prediction of the state:

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

- Define the feedback equation:

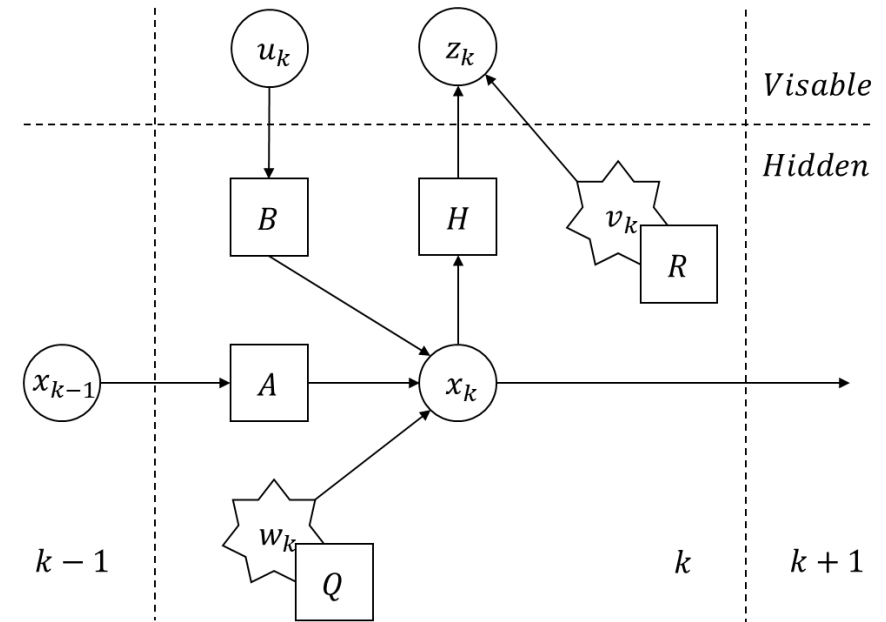
$$\triangleright x_k^{est} = x_k^{pre} + \underset{\text{Kalman Gain}}{\mathbf{K}}(z_k - \underset{\text{Observation}}{z_k^{pre}})$$

- Substitute the observation term of the feedback equation:

$$\begin{aligned} z_k &= Hx_k + v_k, z_k^{pre} = Hx_k^{pre} \\ x_k^{est} &= x_k^{pre} + K(Hx_k + v_k - Hx_k^{pre}) \\ &= x_k^{pre} + KH(x_k - x_k^{pre}) + Kv_k \end{aligned}$$

- The object is to find the optimal Kalman Gain \mathbf{K} to minimize the covariance of the estimation :

$$\triangleright J = \sum_{min} P_k^{est}$$



Kalman Filter

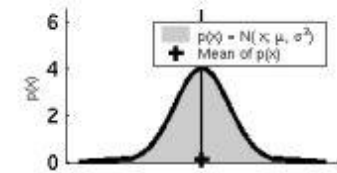
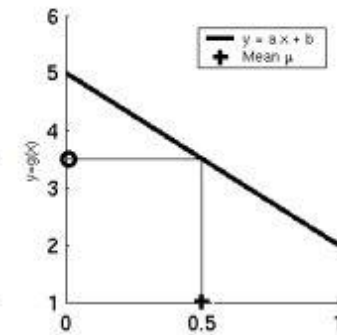
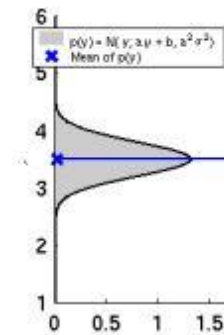
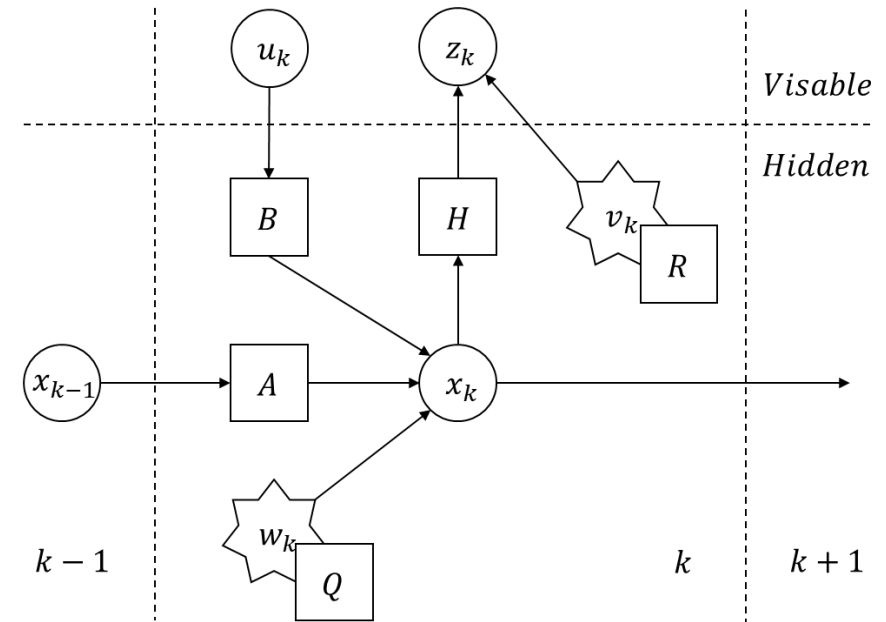
- Propagate the error along the system.
- Compute the covariance of prediction

➤ Prediction error:

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

➤ Covariance:

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



Kalman Filter

- Estimate the covariance of posterior

➤ Estimation error:

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

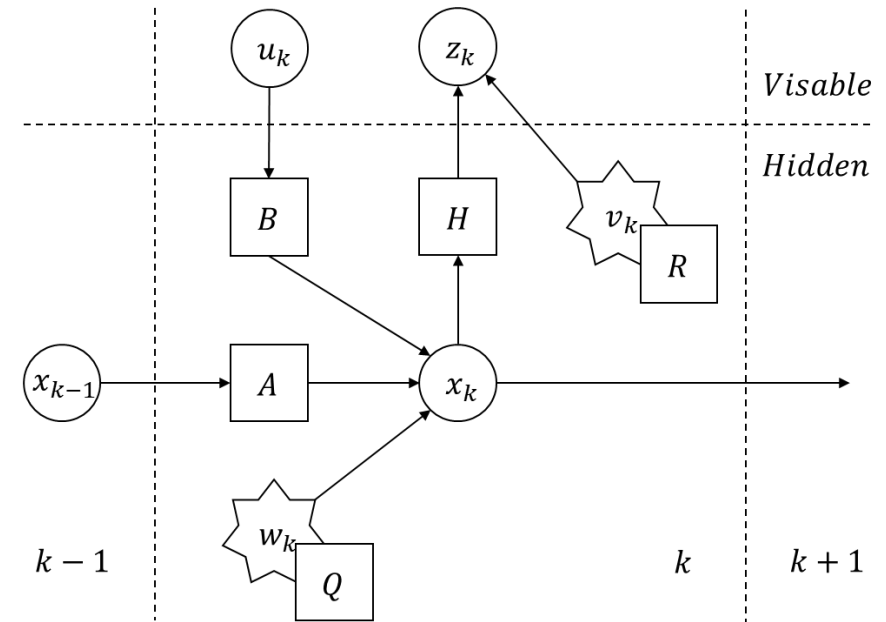
$$x_k^{est} = x_k^{pre} + KH(x_k - x_k^{pre}) + Kv_k$$

➤ Covariance:

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

- Optimize the objective function

$$\begin{aligned} \frac{\partial P_k^{est}}{\partial K} &= -2(P_k^{pre} H^T) + 2K(HP_k^{pre} H^T + R) = 0 \\ K &= P_k^{pre} H^T (HP_k^{pre} H^T + R)^{-1} \end{aligned}$$



Kalman Filter

- Kalman Filter Computation Steps:

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

- Predict the next state

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

- Compute the prediction covariance

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

- Compute Kalman-gain

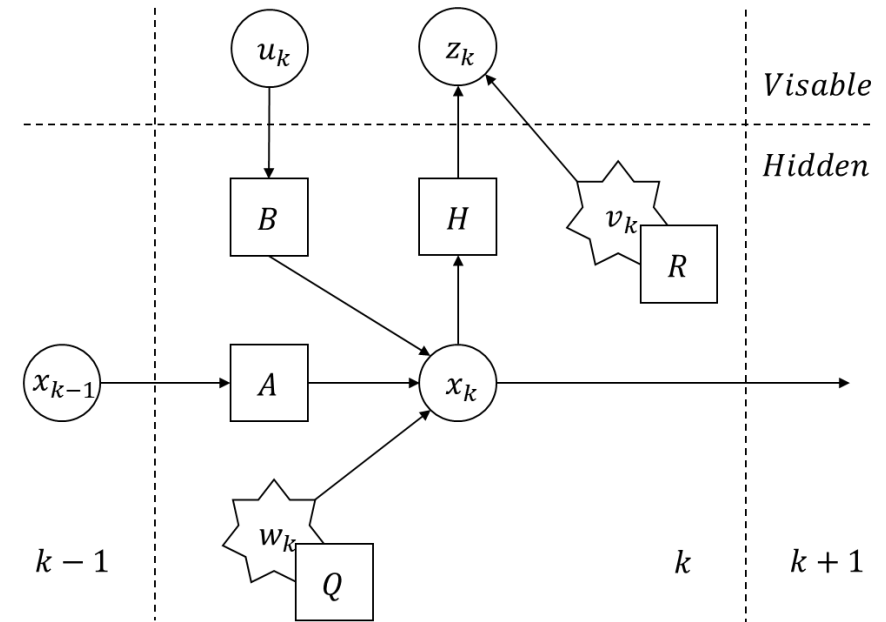
$$K_k = P_k^{pre}H^T(HP_k^{pre}H^T + R)^{-1}$$

- Estimate the mean of the state

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

- Estimate the covariance of the state

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



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

Kalman Filter

- Probabilistic view of Kalman filter
- Prediction Model & Observation Model

➤ $x_k = Ax_{k-1} + Bu_k + w_k$

➤ $z_k = Hx_k + v_k$

$$x_k = H^{-1}(z_k - v_k)$$

- Probability distribution of the prediction and observation

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

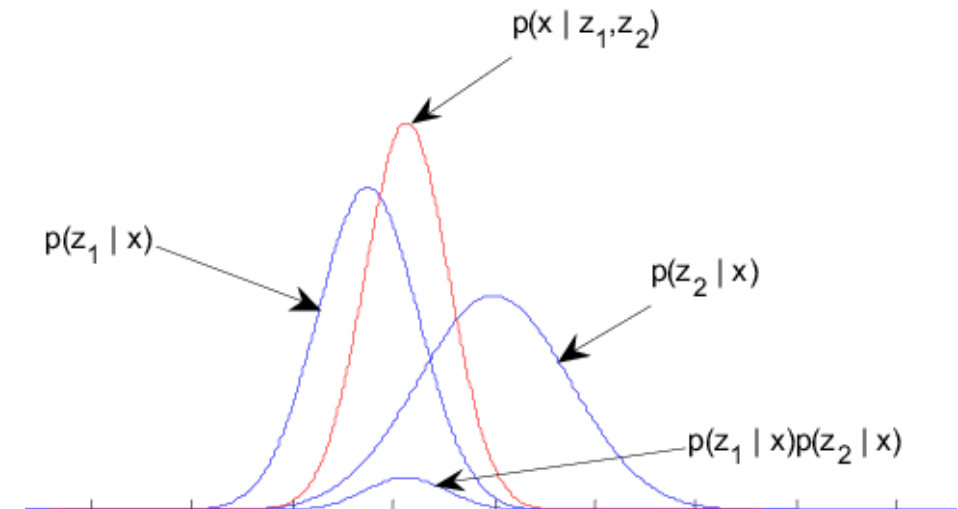
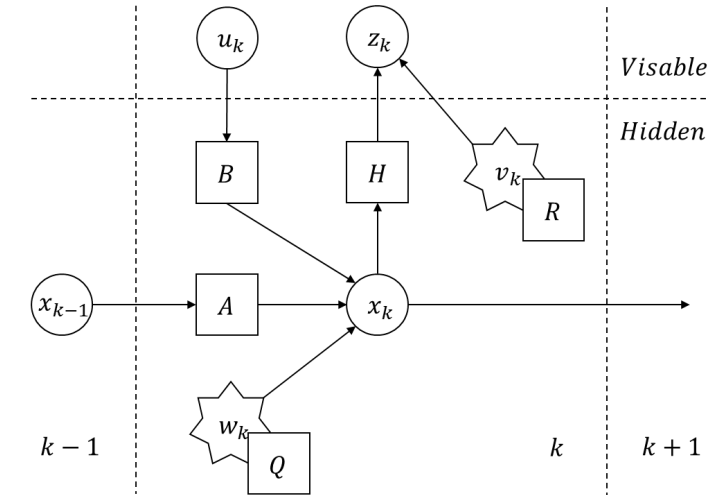
➤ $p(x_k^{obs}) = \mathcal{N}(H^{-1}z_k, H^{-1}RH^{-1})$

- Fusion of Gaussian Distribution

➤ $S = \Sigma_0(\Sigma_0 + \Sigma_1)^{-1}$

➤ $\mu = \mu_0 + S(\mu_1 - \mu_0)$

➤ $\Sigma = \Sigma_0 - S\Sigma_0$



Kalman Filter

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

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

- Fusion the distribution of prediction and observation

➤ Mean: x_k^{est}

$$\begin{aligned} &= x_{k-1}^{pre} + P_k^{pre}(P_k^{pre} + H^{-1}RH^{-T})^{-1}(x_k^{pre} - H^{-1}z_k) \\ &= x_{k-1}^{pre} + P_k^{pre} \mathbf{H^T H^{-T}} (P_k^{pre} + H^{-1}RH^{-T})^{-1} \mathbf{H^{-1}H} (x_k^{pre} - H^{-1}z_k) \\ &= x_{k-1}^{est} + Bu_k + P_k^{pre} \mathbf{H^T (HP_k^{pre}H^T + R)^{-1}} (Hx_k^{pre} - z_k) \\ &= x_{k-1}^{est} + Bu_k + K_k(Hx_k^{pre} - z_k) \end{aligned}$$

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

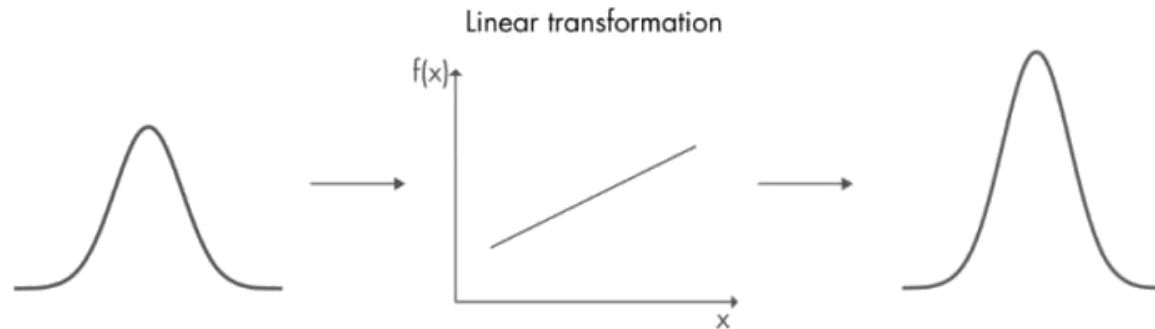
➤ Covariance: P_k^{est}

$$\begin{aligned} &= P_k^{pre} - P_k^{pre}(P_k^{pre} + H^{-1}RH^{-T})^{-1}P_k^{pre} \\ &= P_k^{pre} - P_k^{pre} \mathbf{H^T H^{-T}} (P_k^{pre} + H^{-1}RH^{-T})^{-1} \mathbf{H^{-1}H} P_k^{pre} \\ &= P_k^{pre} - P_k^{pre} \mathbf{H^T (HP_k^{pre}H^T + R)^{-1}} H P_k^{pre} \\ &= P_k^{pre} - K_k H P_k^{pre} = (I - K_k H) P_k^{pre} \end{aligned}$$

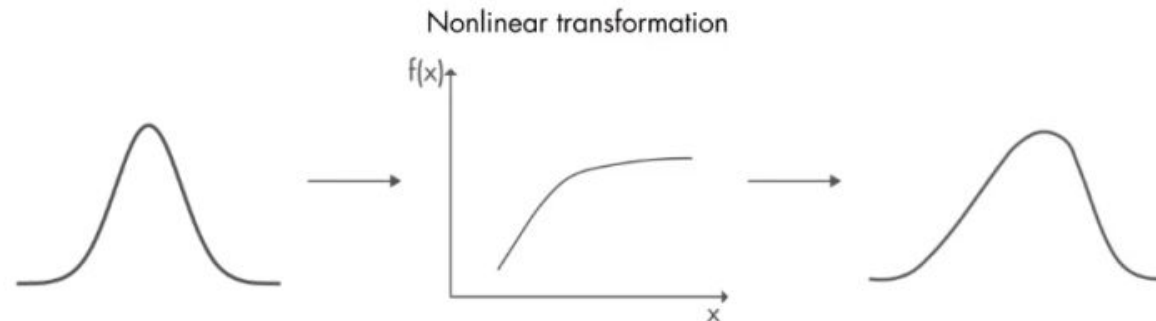
$$\begin{aligned} x_k^{pre} &= Ax_{k-1}^{est} + Bu_k \\ P_k^{pre} &= AP_{k-1}^{est}A^T + Q \\ K_k &= P_k^{pre} H^T (HP_k^{pre}H^T + R)^{-1} \\ x_k^{est} &= x_k^{pre} + K_k(z_k - Hx_k^{pre}) \\ P_k^{est} &= (I - K_k H) P_k^{pre} \end{aligned}$$

Extended Kalman Filter (EKF)

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

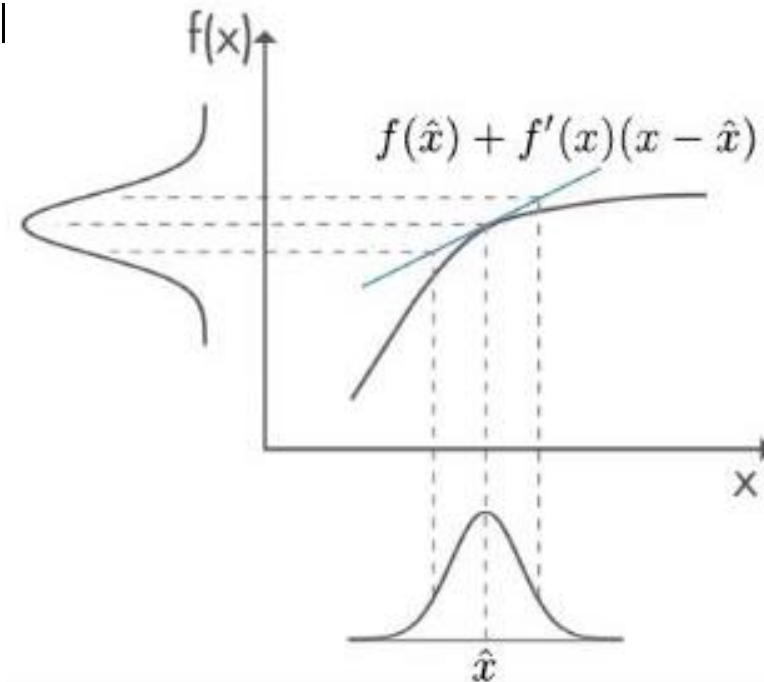


- However, the prediction model is usually nonlinear, the state distribution after transformation will not be a Gaussian.



Extended Kalman Filter (EKF)

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



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

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

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

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

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

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

Kalman-Filter

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

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

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

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

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

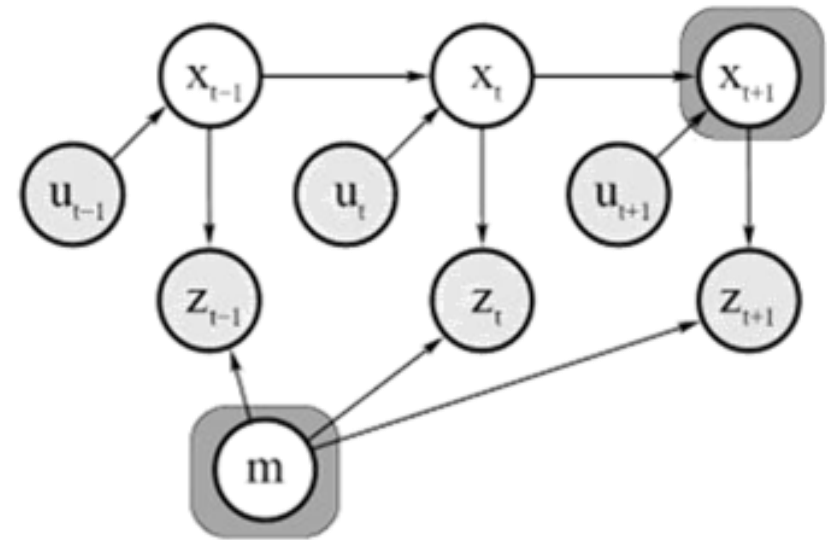
EKF-SLAM

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

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

- Probability distribution of the state:

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



Extended Kalman-Filter

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

EKF-SLAM (Prediction Model)

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

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos\theta \\ v \sin\theta \\ \omega \end{bmatrix}$$

- 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:

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

EKF-SLAM (Prediction Model)

- First, we integrate the angle:
 - $\theta(t) = \int \omega dt, \quad \theta(t) = \omega t + C$
- Consider the initial condition of angle
 - $\theta(0) = \hat{\theta}$, we can get the scalar term $C = \hat{\theta}$
- Then we can substitute the angle term for integral of x and y
 - $x(t) = \int v \cos(\hat{\theta} + \omega t) dt = \frac{v}{\omega} \sin(\hat{\theta} + \omega t) + C$
 $y(t) = \int v \sin(\hat{\theta} + \omega t) dt = -\frac{v}{\omega} \cos(\hat{\theta} + \omega t) + C$
- Consider the initial condition of position
 - $x(0) = \hat{x}, y(0) = \hat{y}$, we can get
 $x(t) = \int v \cos(\hat{\theta} + \omega t) dt = \frac{v}{\omega} \sin(\hat{\theta} + \omega t) - \frac{v}{\omega} \sin(\hat{\theta}) + \hat{x}$
 $y(t) = \int v \sin(\hat{\theta} + \omega t) dt = -\frac{v}{\omega} \cos(\hat{\theta} + \omega t) + \frac{v}{\omega} \cos(\hat{\theta}) + \hat{y}$

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:

$$\begin{aligned} \triangleright F_t^x &= \frac{\partial f}{\partial (x,y,\theta)^T} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} -\frac{v_t}{\omega_t} \sin(\theta) + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos(\theta) - \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ \omega \Delta t + \theta \end{bmatrix} = I + \frac{\partial f}{\partial (x,y,\theta)^T} \begin{bmatrix} -\frac{v_t}{\omega_t} \sin(\theta) + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos(\theta) - \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \end{bmatrix} \\ &= I + \begin{bmatrix} 0 & 0 & -\frac{v_t}{\omega_t} \cos(\theta) + \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ 0 & 0 & -\frac{v_t}{\omega_t} \sin(\theta) + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\frac{v_t}{\omega_t} \cos(\theta) + \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ 0 & 1 & -\frac{v_t}{\omega_t} \sin(\theta) + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

EKF-SLAM (Observation Model)

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

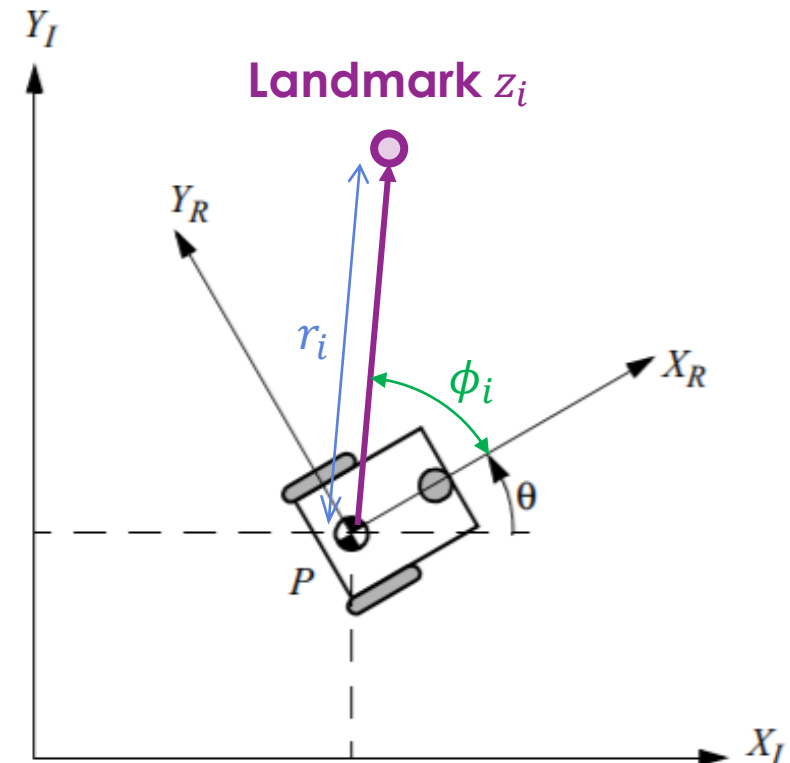
➤
$$\begin{bmatrix} m_{i,x} \\ m_{i,y} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} r_i \cos(\phi_i + \theta) \\ r_i \sin(\phi_i + \theta) \end{bmatrix}$$

- Define the following term:

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

- The observation can be represented as:

➤
$$z_i = \begin{bmatrix} \sqrt{q} \\ \text{atan2}(\delta_x, \delta_y) - \theta \end{bmatrix}$$



EKF-SLAM (Observation Model)

- Given observation model

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

- Linearized the observation model :

$$\begin{aligned} \triangleright H^i &= \frac{\partial z_i}{\partial (x, y, \theta, m_{i,x}, m_{i,y})} = \begin{bmatrix} \frac{\partial \sqrt{q}}{\partial x} & \frac{\partial \sqrt{q}}{\partial y} & \dots \\ \frac{\partial \text{atan2}(\delta_x, \delta_y)}{\partial x} & \frac{\partial \text{atan2}(\delta_x, \delta_y)}{\partial y} & \dots \end{bmatrix} \\ &= \frac{1}{q} \begin{bmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & \sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{bmatrix} \end{aligned}$$

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

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

EKF-SLAM

- Prediction Model

$$\triangleright F_t = \begin{bmatrix} 1 & 0 & 0 & 0 \dots 0 \\ 0 & 1 & 0 & 0 \dots 0 \\ 0 & 0 & 1 & 0 \dots 0 \end{bmatrix}^T * F_t^x, \text{ in which } F_t^x = \begin{bmatrix} 1 & 0 & -\frac{v_t}{\omega_t} \cos(\theta) + \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ 0 & 1 & -\frac{v_t}{\omega_t} \sin(\theta) + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ 0 & 0 & 1 \end{bmatrix}$$

- Observation Model

$$\triangleright H_t = \begin{bmatrix} 1 & 0 & 0 & 0 \dots 0 & 0 & 0 & 0 \dots 0 \\ 0 & 1 & 0 & 0 \dots 0 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 1 & 0 \dots 0 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 1 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 0 & 1 & 0 \dots 0 \end{bmatrix}^T * H_t^i$$

$x \quad y \quad \theta \quad m_{i,x} \quad m_{i,y}$

Extended Kalman-Filter

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

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