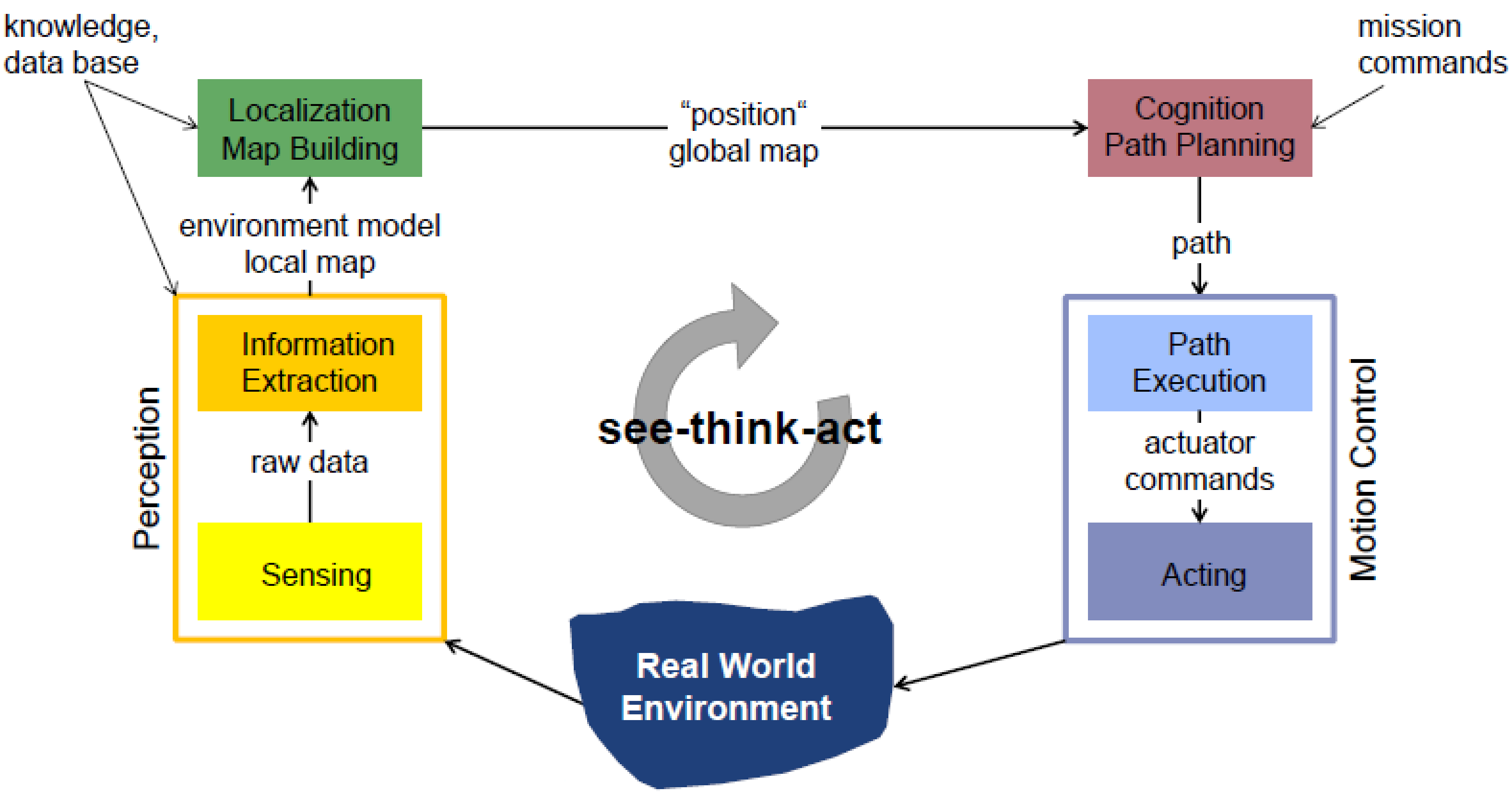
EE565:Mobile Robotics
Lecture 9

Welcome

Dr. Ahmad Kamal Nasir

Probabilistic Map-based Localization

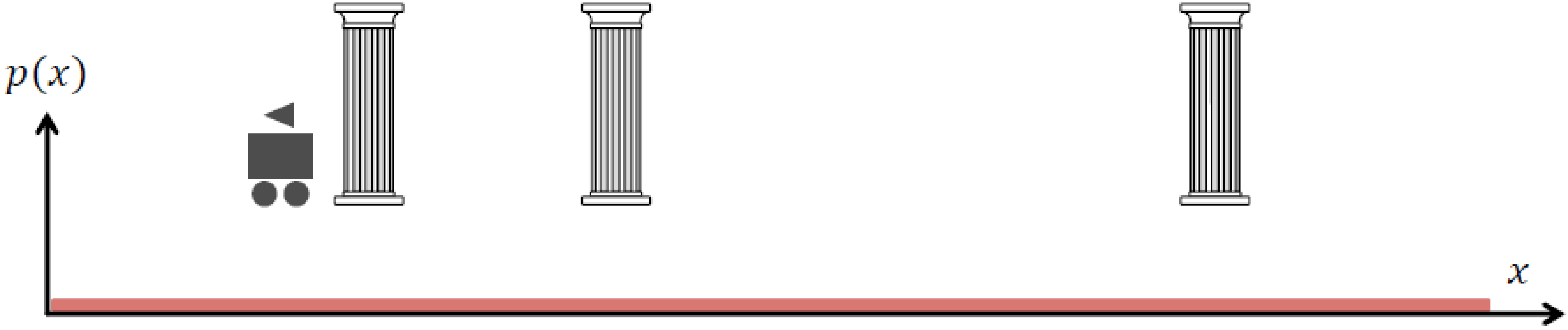


Definition, Challenges and Approach

- Map-based localization
  - The robot estimates its position using perceived information and a map
  - The map
    - might be known (localization)
    - Might be built in parallel (simultaneous localization and mapping - SLAM)
- Challenges
  - Measurements and the map are inherently error prone
  - Thus the robot has to deal with uncertain information
    - Probabilistic map-base localization
- Approach
  - The robot estimates the belief state about its position through an ACT and SEE cycle

SEE and ACT to improve belief state

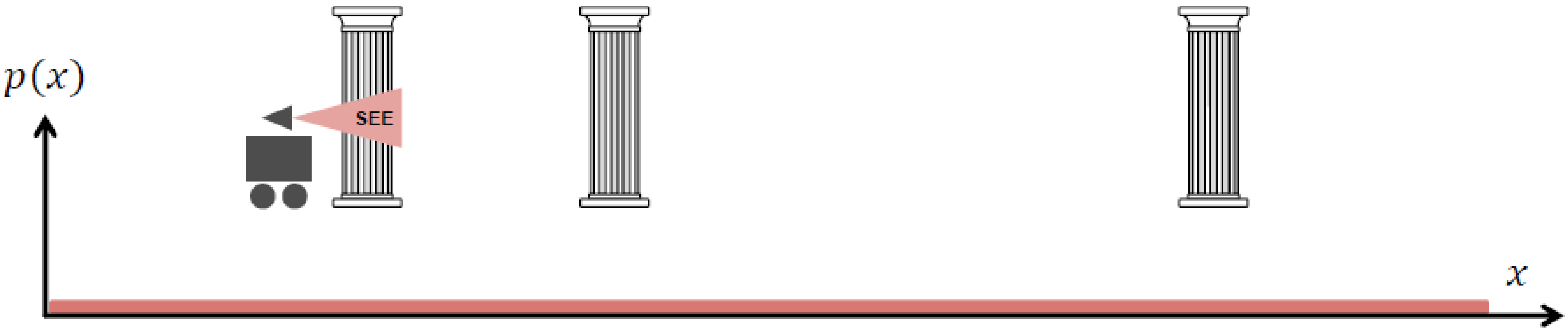
- Robot is placed somewhere in the environment → location unknown
- SEE: The robot queries its sensors → finds itself next to a pillar
- ACT: Robot moves one meter forward
  - motion estimated by wheel encoders
  - accumulation of uncertainty
- SEE: The robot queries its sensors again → finds itself next to a pillar
- Belief updates (information fusion)



SEE and ACT to improve belief state

- Robot is placed somewhere in the environment → location unknown
- SEE: The robot queries its sensors → finds itself next to a pillar

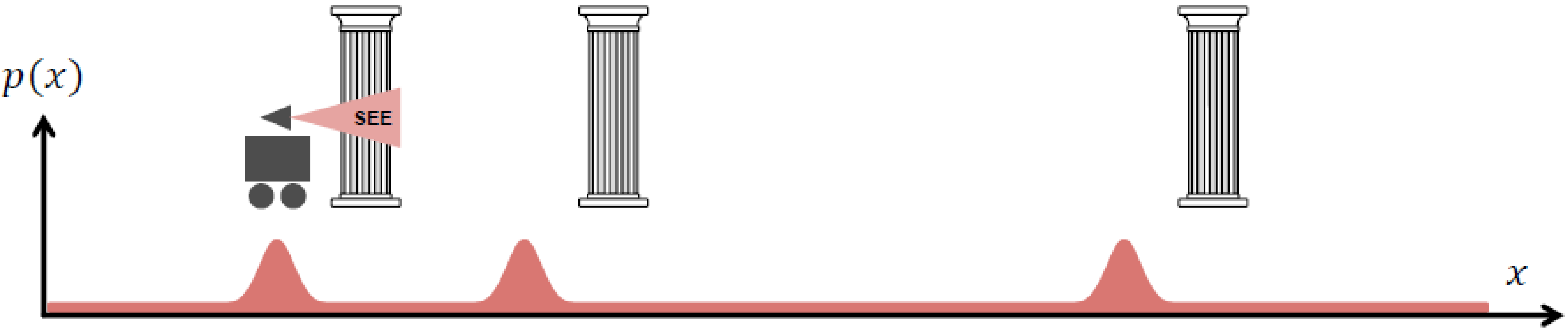
- ACT: Robot moves one meter forward
  - motion estimated by wheel encoders
  - accumulation of uncertainty
- SEE: The robot queries its sensors again → finds itself next to a pillar
- Belief updates (information fusion)



SEE and ACT to improve belief state

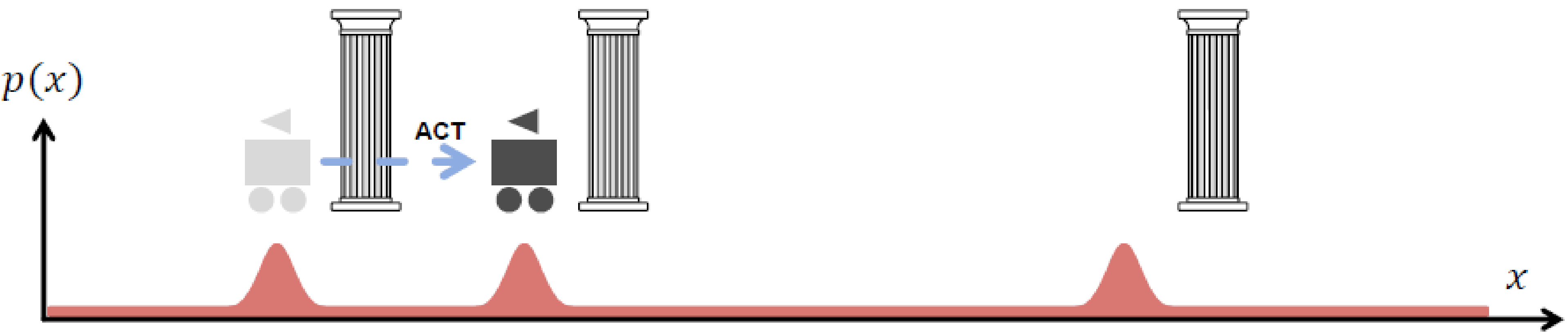
- Robot is placed somewhere in the environment → location unknown
- SEE: The robot queries its sensors → finds itself next to a pillar

- ACT: Robot moves one meter forward
  - motion estimated by wheel encoders
  - accumulation of uncertainty
- SEE: The robot queries its sensors again → finds itself next to a pillar
- Belief updates (information fusion)



SEE and ACT to improve belief state

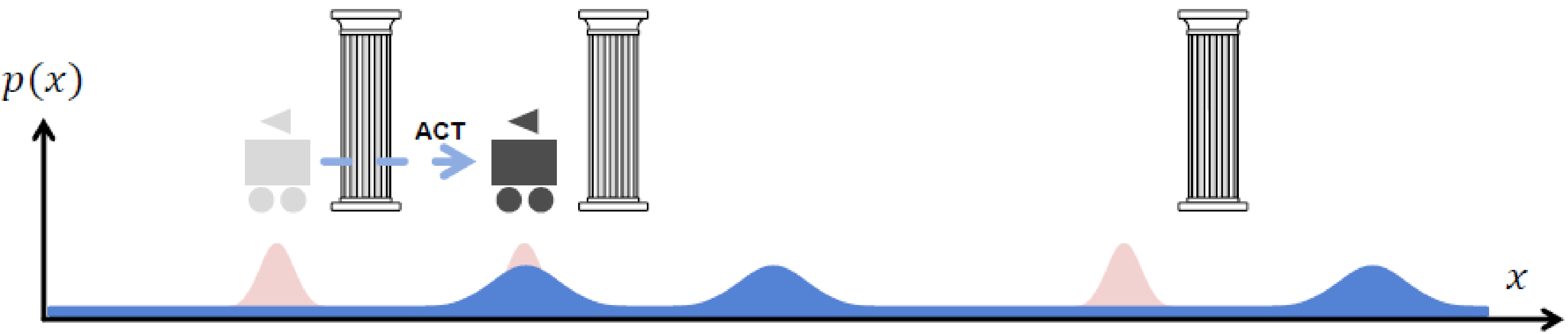
- Robot is placed somewhere in the environment → location unknown
- SEE: The robot queries its sensors → finds itself next to a pillar
- ACT: Robot moves one meter forward
  - motion estimated by wheel encoders
  - accumulation of uncertainty
- SEE: The robot queries its sensors again → finds itself next to a pillar
- Belief updates (information fusion)



SEE and ACT to improve belief state

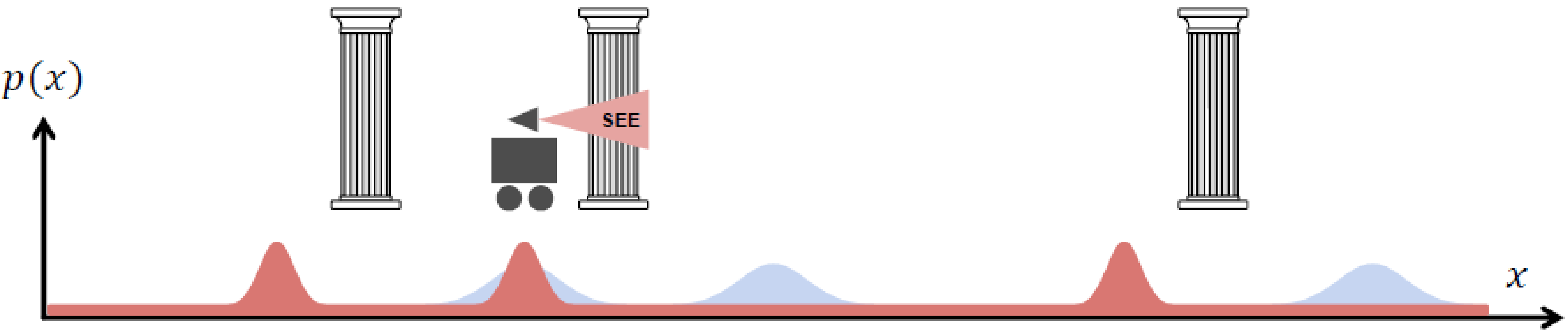
- Robot is placed somewhere in the environment → location unknown
- SEE: The robot queries its sensors → finds itself next to a pillar

- ACT: Robot moves one meter forward
  - motion estimated by wheel encoders
  - accumulation of uncertainty
- SEE: The robot queries its sensors again → finds itself next to a pillar
- Belief updates (information fusion)



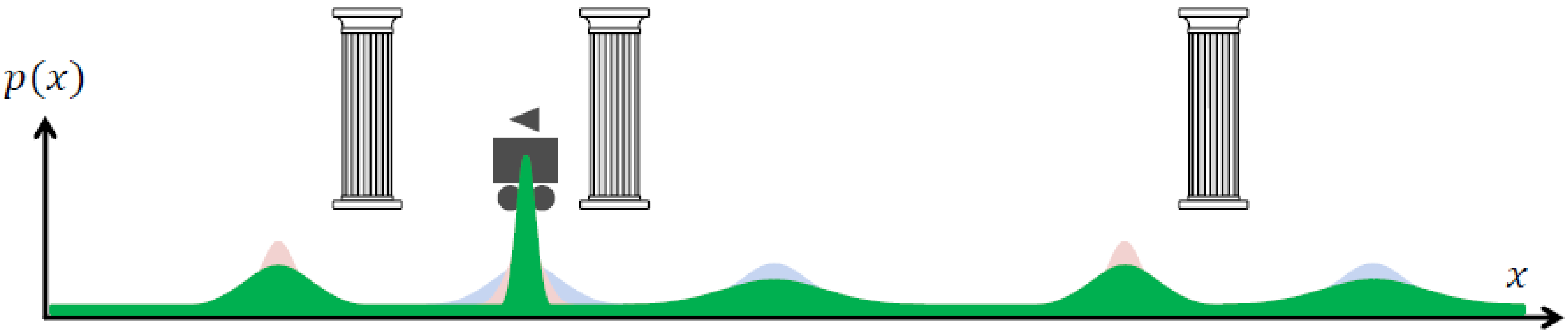
SEE and ACT to improve belief state

- Robot is placed somewhere in the environment → location unknown
- SEE: The robot queries its sensors → finds itself next to a pillar
- ACT: Robot moves one meter forward
  - motion estimated by wheel encoders
  - accumulation of uncertainty
- SEE: The robot queries its sensors again → finds itself next to a pillar
- Belief updates (information fusion)



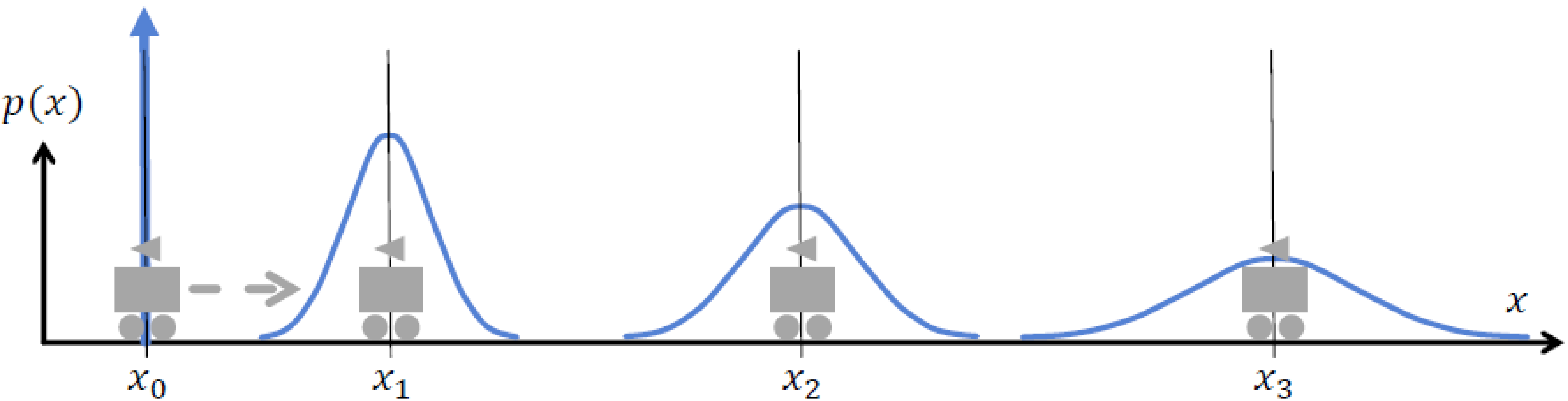
SEE and ACT to improve belief state

- Robot is placed somewhere in the environment → location unknown
- SEE: The robot queries its sensors → finds itself next to a pillar
- ACT: Robot moves one meter forward
  - motion estimated by wheel encoders
  - accumulation of uncertainty
- SEE: The robot queries its sensors again → finds itself next to a pillar
- Belief update (information fusion)



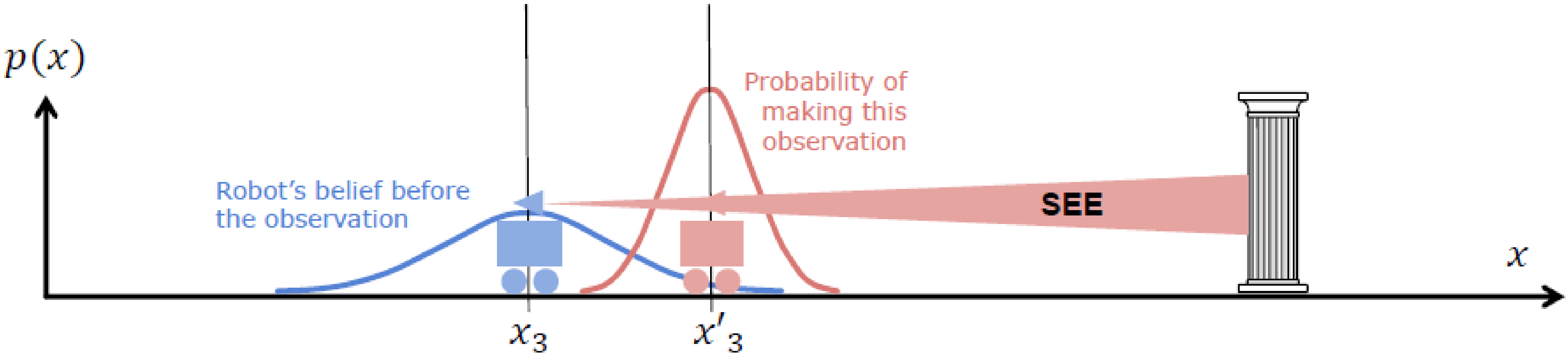
Using Motion Model and its Uncertainties

- The robot moves and estimates its position through its proprioceptive sensors
  - Wheel Encoder (Odometry)
- During this step, the robot's state uncertainty grows



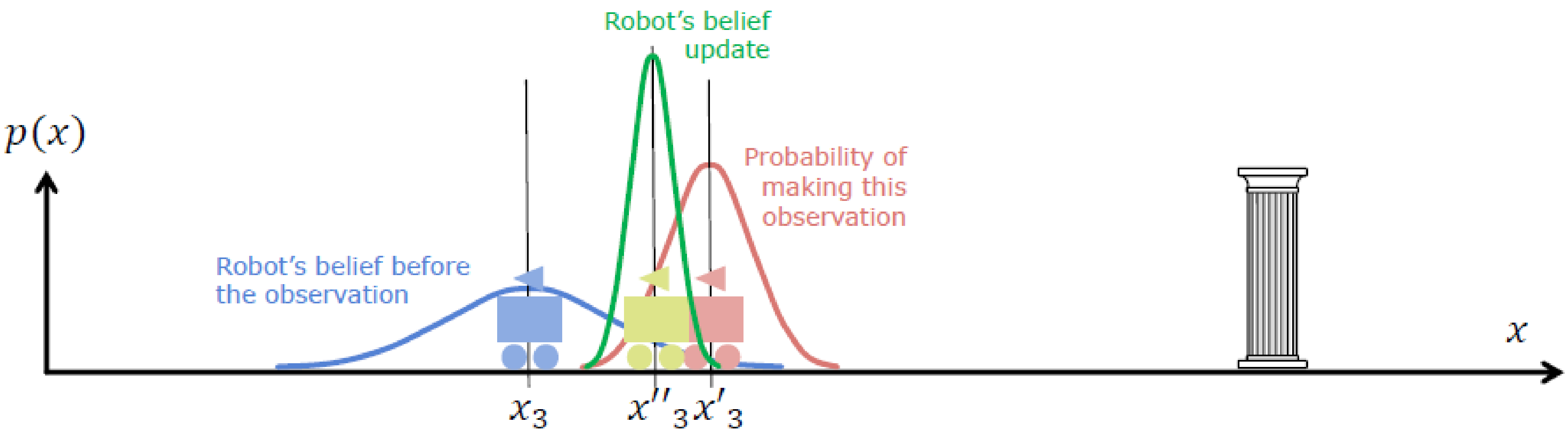
Estimation of Position based on Perception and Map

- The robot makes an observation using its exteroceptive sensors
- This results in a second estimation of the current position



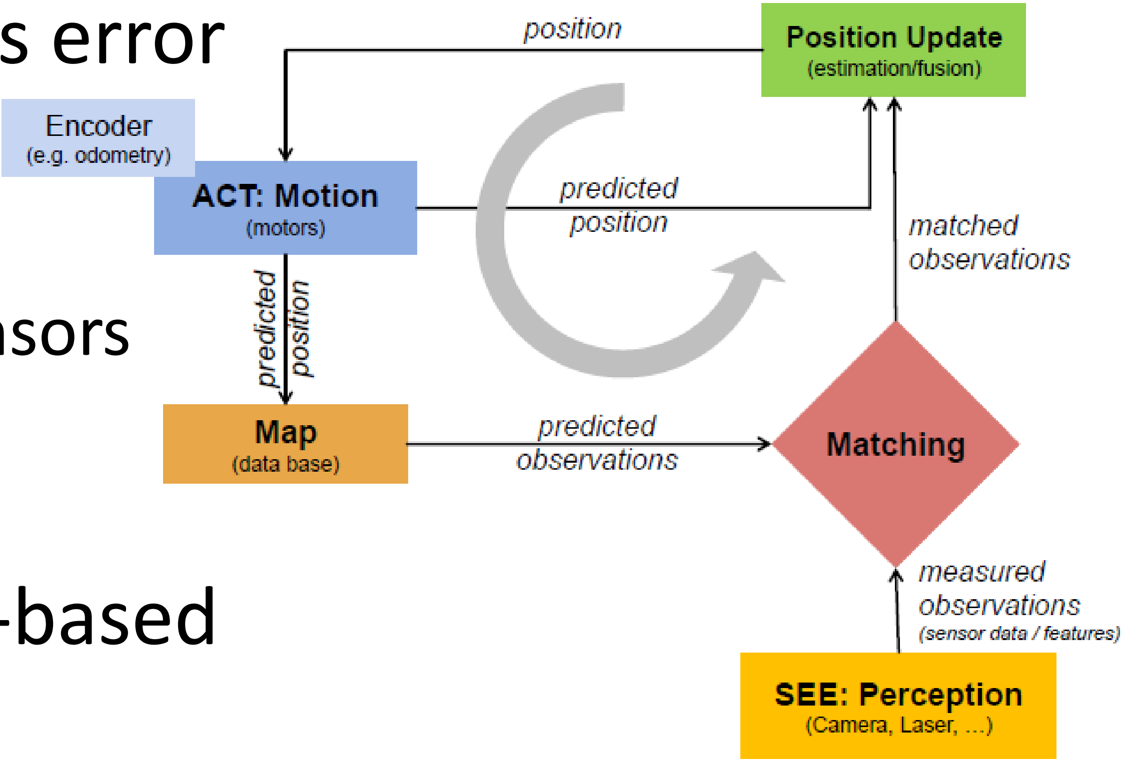
Fusion of Prior Belief with Observation

- The robot corrects its position by combining its belief before the observation with the probability of making exactly that observation
- During this step, the robot's state uncertainty shrinks



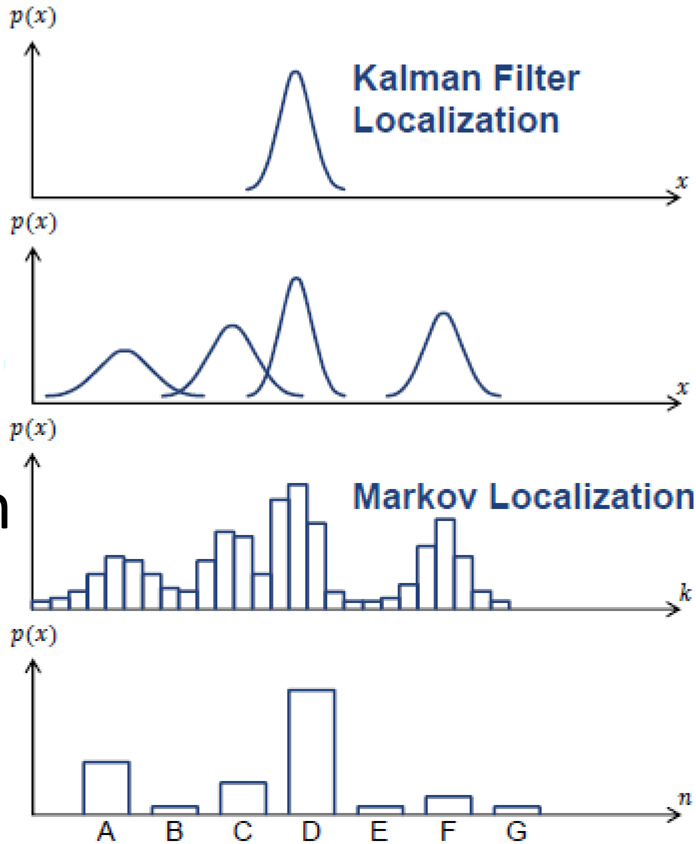
The Estimation Cycle (ACT-SEE)

- Information (measurements) is error prone (uncertain)
  - Odometry
  - Exteroceptive sensors (camera, laser, ...)
  - Map
- Probabilistic map-based localization



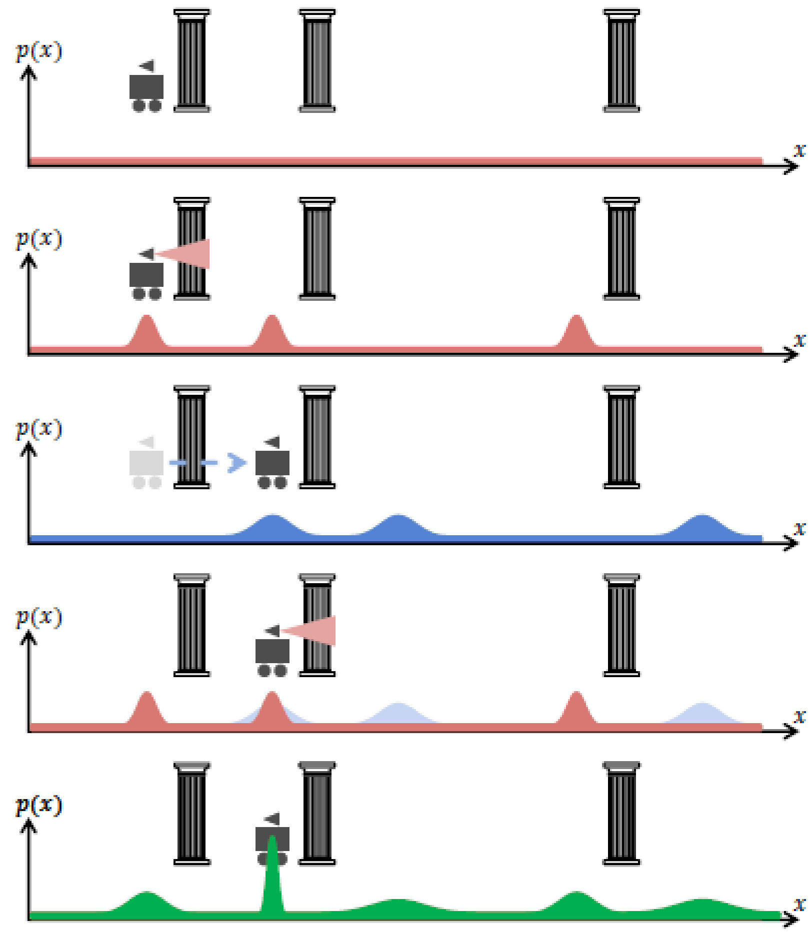
Belief Representation

- Continuous map with single hypothesis probability distribution $p(x)$
- Continuous map with multiple hypotheses probability distribution $p(x)$
- Discretized metric map (grid k) with probability distribution $p(k)$
- d) Discretized topological map (nodes n) with
- probability distribution $p(n)$



ACT - SEE Cycle for Localization

- SEE: The robot queries its sensors
  - finds itself next to a pillar
- ACT: Robot moves one meter forward
  - motion estimated by wheel encoders
  - accumulation of uncertainty
- SEE: The robot queries its sensors
  - again → finds itself next to a pillar
- Belief update (information fusion)



Refresher on Probability Theory: joint distribution

- $p(x, y)$: joint distribution representing the probability that the random variable X takes on the value x and that Y takes on the value y

- If X and Y are independent we can write:

$$p(x, y) = p(x)\, p(y)$$

Refresher on Probability Theory: conditional probability

- $p(x|y)$: conditional probability that describes the probability that the random variable X takes on the value x conditioned on the knowledge that Y for sure takes y.

$$p(x|y) = \frac{p(x,y)}{p(y)}$$

- and if X and Y are independent (uncorrelated) we can write:

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

Refresher on Probability Theory: theorem of total probability

- The **theorem of total probability** (*convolution*) originates from the axioms of probability theory and is written as:

$$p(x) = \sum_{y} p(x|y)p(y)$$ for discrete probabilities

$$p(x) = \int_{y} p(x|y)p(y)dy$$ for continuous probabilities

- This theorem is used by both Markov and Kalman-filter localization algorithms during the prediction update.

Refresher on Probability Theory: the Bayes rule

- The Bayes rule relates the conditional probability $p(x|y)$ to its inverse $p(y|x)$
- Under the condition that $p(y) > 0$, the Bayes rule is written as:

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

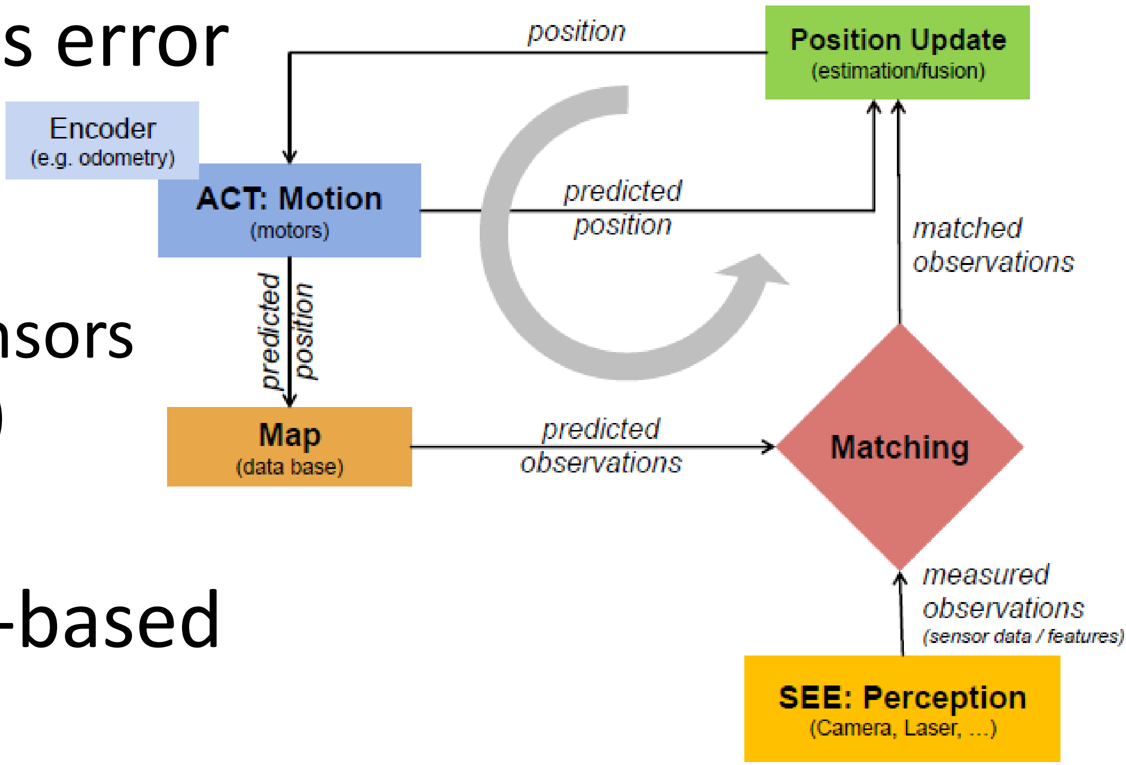
$$p(x|y) = \eta p(y|x)p(x)$$

$\eta = p(y)^{-1}$ normalization factor ($\int p = 1$)

- This theorem is used by both Markov and Kalman-filter localization algorithms during the measurement update.

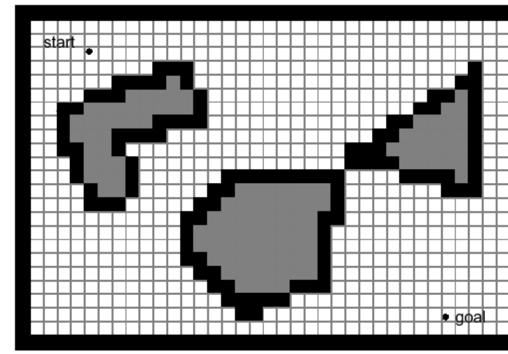
Markov localization: applying probability theory to localization

- Information (measurements) is error prone (uncertain)
  - Odometry
  - Exteroceptive sensors (camera, laser, …)
  - Map
- Probabilistic map-based localization



Basics and Assumption

- Discretized pose representation $x_t \rightarrow$ grid map
- Markov localization tracks the robot's belief state $bel(x_t)$ using an arbitrary probability density function to represent the robot's position



- *Markov assumption*: Formally, this means that the output of the estimation process is a function x_t only of the robot's previous state x_{t-1} and its most recent actions (odometry) u_t and perception z_t.

$$p(x_t | x_0, u_t \cdots u_0, z_t \cdots z_0) = p(x_t | x_{t-1}, u_t, z_t)$$

- Markov localization addresses the *global localization problem*, the *position tracking problem*, and the *kidnapped robot problem*.

Applying Probability Theory to Localization

- **ACT** | probabilistic estimation of the robot's new belief state $\overline{bel}(x_t)$ based on the previous location $\overline{bel}(x_{t-1})$ and the probabilistic motion model $p(x_t|u_t, x_{t-1})$ with action u_t (control input)
  - application of theorem of total probability / convolution

$$\overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1}) bel(x_{t-1})\, dx_{t-1}$$ for continuous probabilities

$$\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t|u_t, x_{t-1}) bel(x_{t-1})$$ for discrete probabilities

Applying Probability Theory to Localization

- **SEE** | probabilistic estimation of the robot's new belief state $bel(x_t)$ as a function of its measurement data z_t and its former belief state $\overline{bel}(x_t)$:
  - application of ***Bayes rule***

$$bel(x_t) = \eta p(z_t | x_t, M)\overline{bel}(x_t)$$

- where $p(z_t | x_t, m_t)$ is the probabilistic measurement model (SEE), that is, the probability of observing the measurement data z_t given the knowledge of the map m_t and the robot's position x_t. Thereby $\eta = p(y)^{-1}$is the normalization factor so that $\sum p = 1$.

The Basic Algorithms for Markov Localization

For all x_t do

$\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t|u_t, x_{t-1})bel(x_{t-1})$ (prediction update)

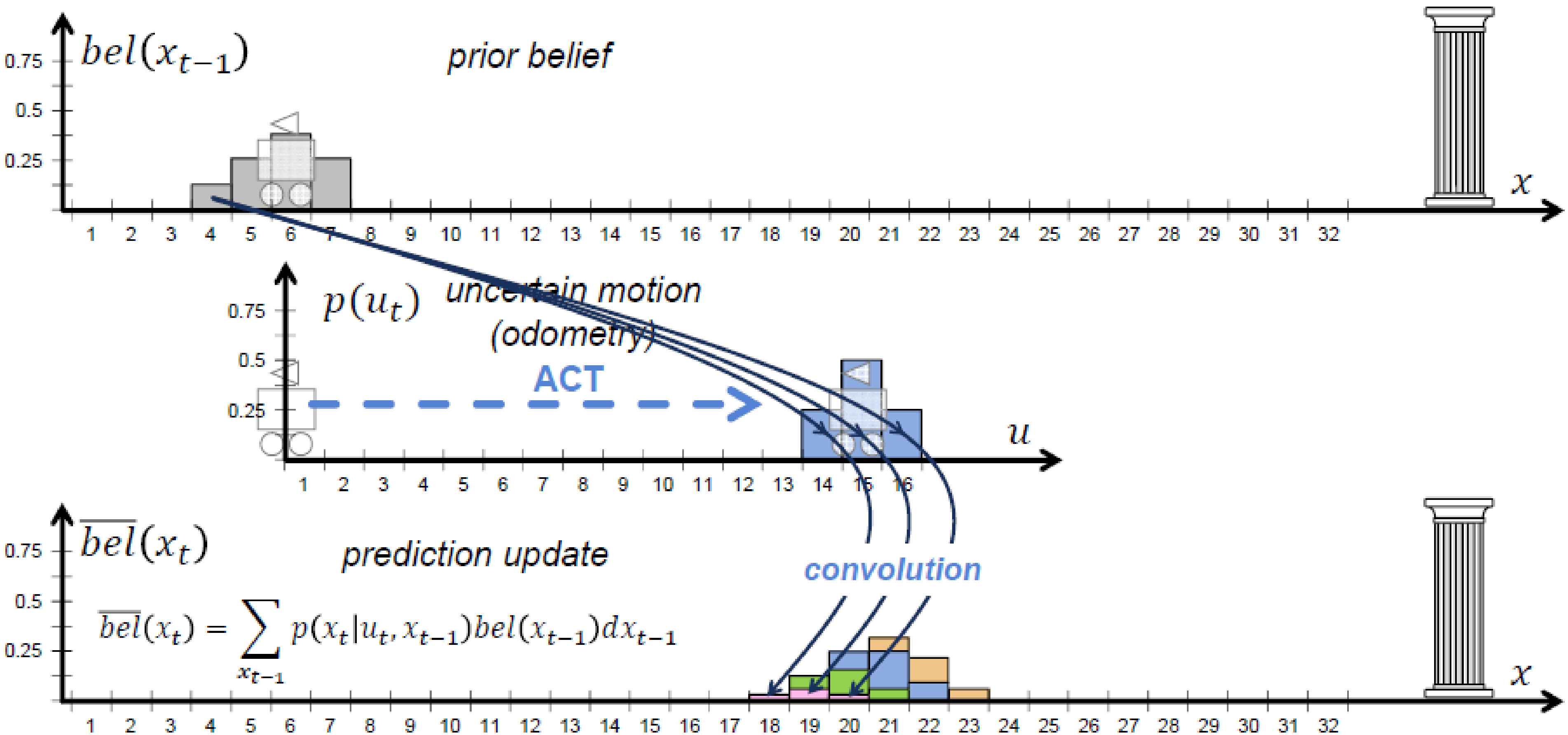
$bel(x_t) = \eta p(z_t|x_t, M)\overline{bel}(x_t)$ (measurement update)

endfor

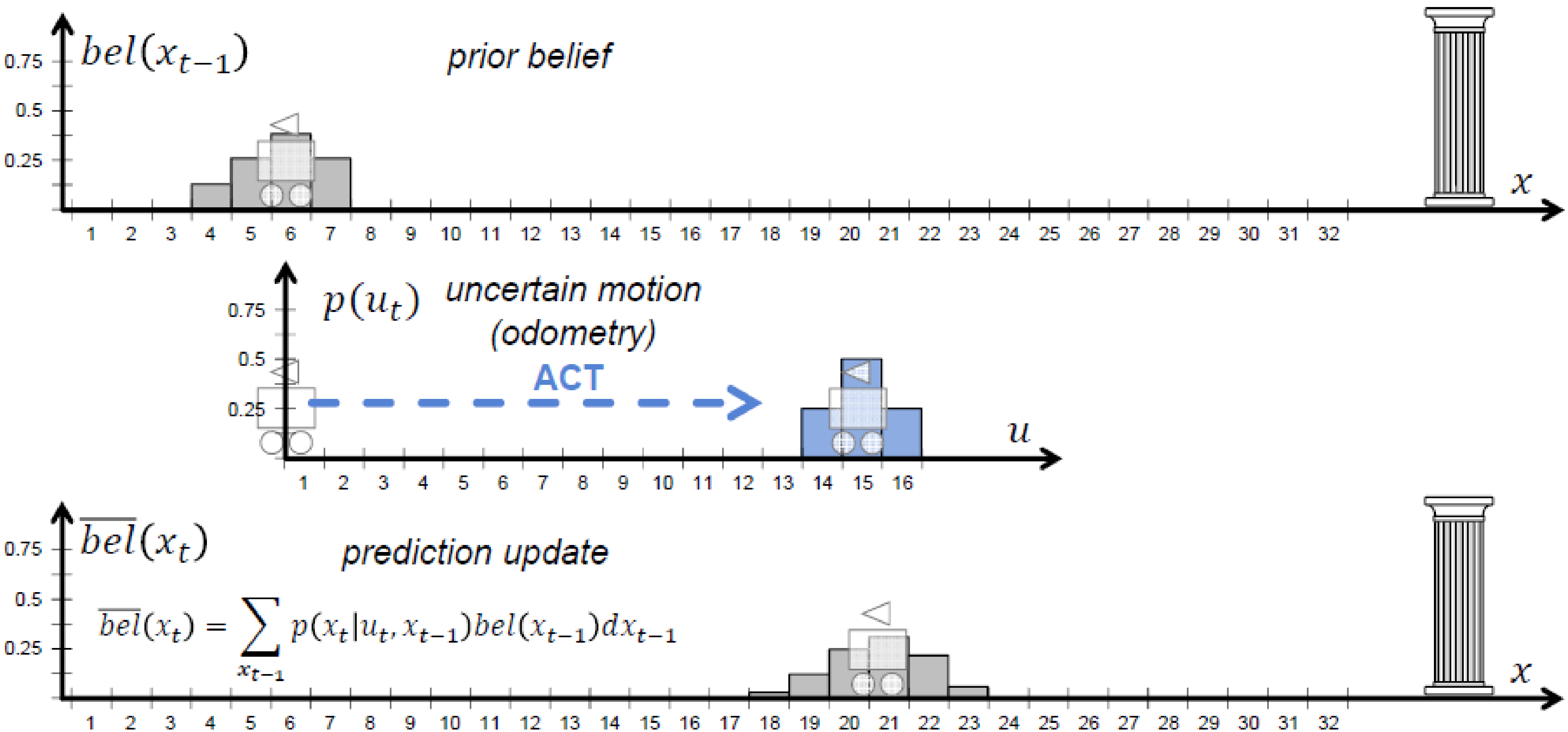
Return $bel(x_t)$

- ***Markov assumption***: Formally, this means that the output is a function x_t only of the robot's previous state x_{t-1} and its most recent actions (odometry) u_t and perception z_t.

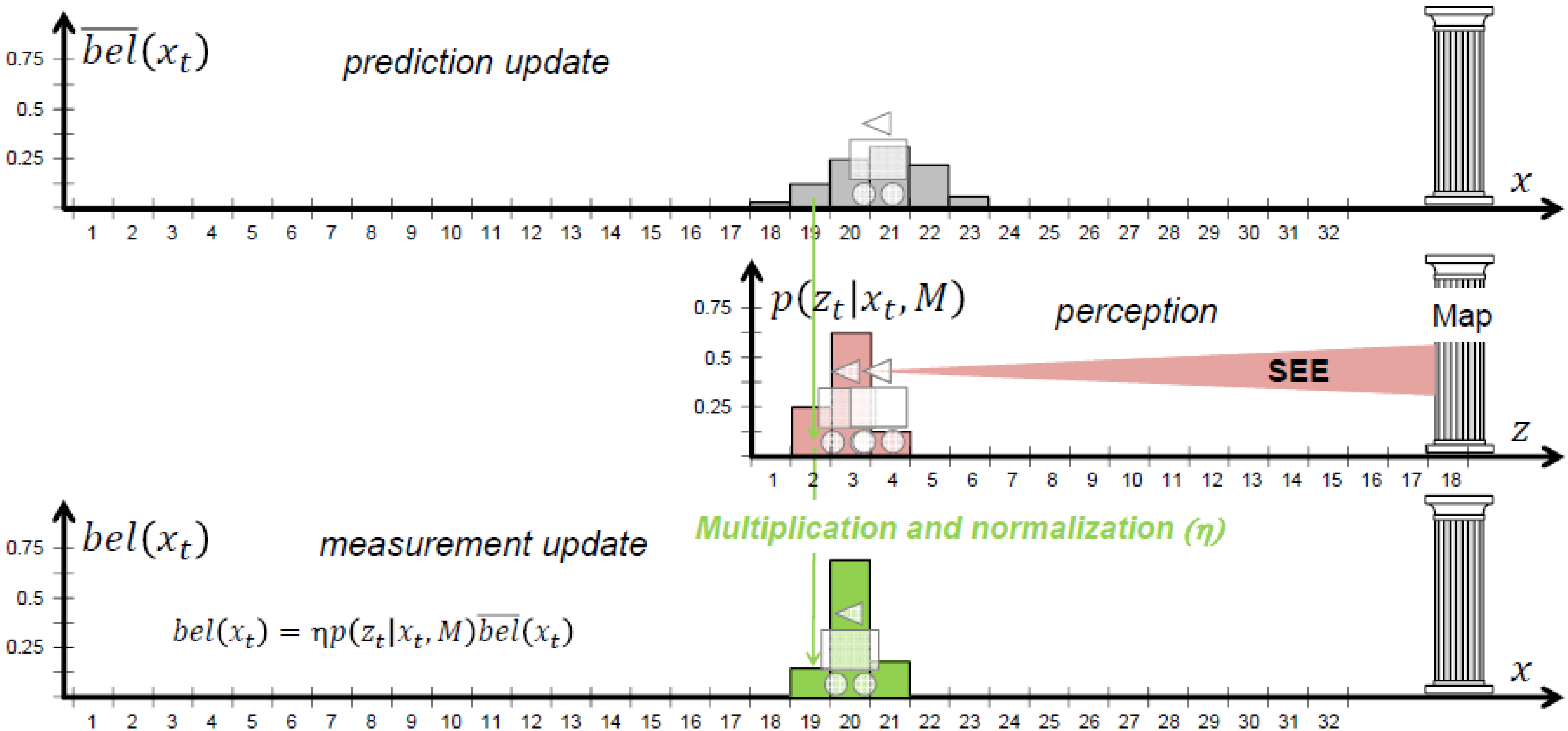
Using Motion Model and its Uncertainties



Using Motion Model and its Uncertainties

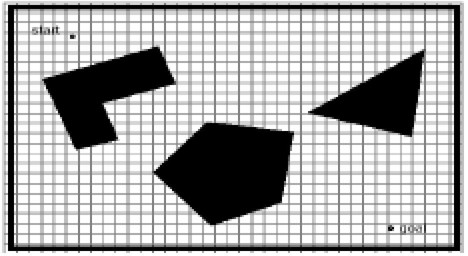


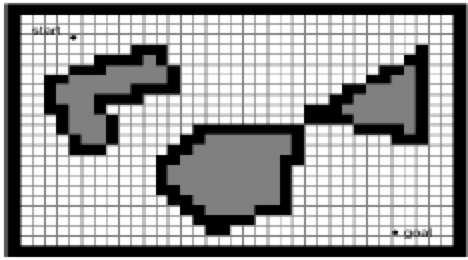
Estimation of Position based on Perception and Map



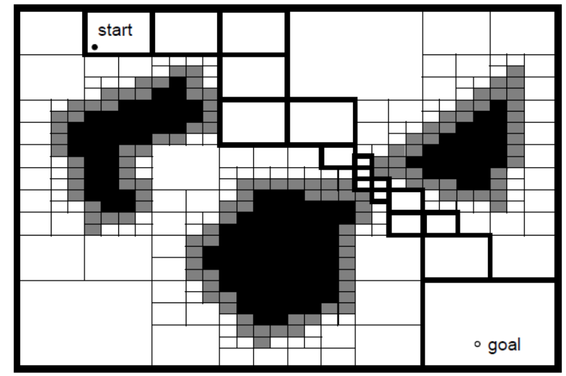
Extension to 2D

- The real world for mobile robot is at least 2D (moving in the plane)
  - discretized pose state space (grid) consists of x, y, θ
  - Markov Localization scales badly with the size of the environment
- Space: 10 m x 10 m with a grid size of 0.1 m and an angular resolution of $1°$
  - $100 \cdot 100 \cdot 360 \ = 3.6\ 10^6$ grid points (states)
  - prediction step requires in worst case $(3.6\ 10^6)^2$ multiplications and summations
- Fine fixed decomposition grids result in a huge state space
  - Very important processing power needed
  - Large memory requirement



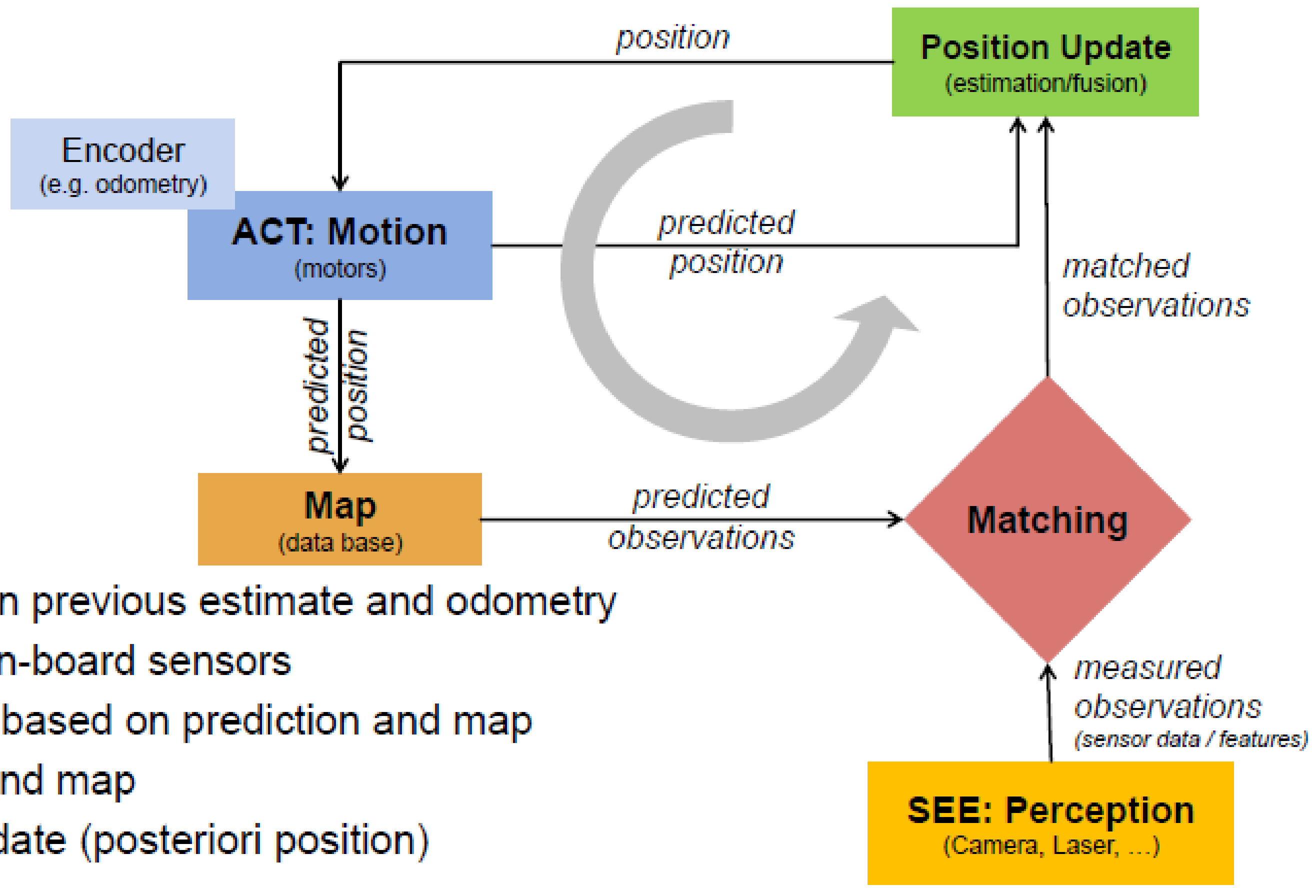


Reducing Computational Complexity



- Adaptive cell decomposition
- Motion model (Odomety) limited to a small number of grid points
- Randomized sampling
  - Approximation of belief state by a representative subset of possible locations
  - weighting the sampling process with the probability values
  - Injection of some randomized (not weighted) samples
  - randomized sampling methods are also known as particle filter algorithms, condensation algorithms, and Monte Carlo algorithms.

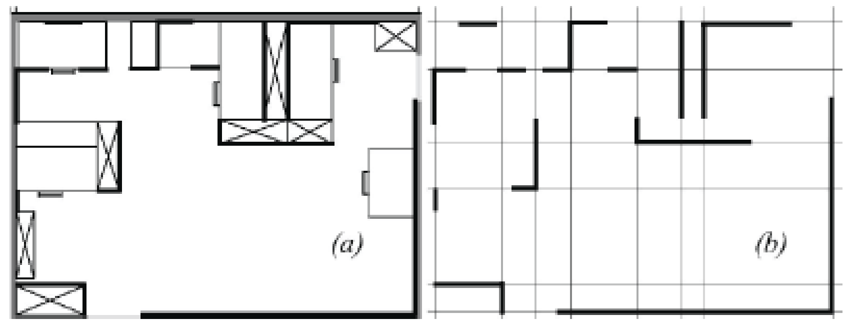
Kalman Filter Localization: applying probability theory to localization

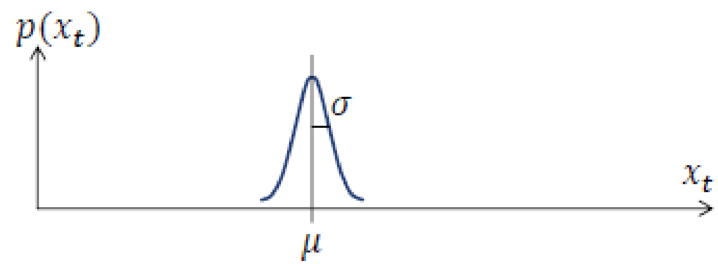


1. **Prediction (ACT)** based on previous estimate and odometry
2. **Observation (SEE)** with on-board sensors
3. **Measurement prediction** based on prediction and map
4. **Matching** of observation and map
5. **Estimation** → position update (posteriori position)

Basics and assumption

- Continuous pose representation x_t
- Kalman Filter Assumptions:
  - Error approximation with normal distribution: $x = N(\mu, \sigma^2)$ (Gaussian model)
  - Output y_t distribution is a linear (or linearized) function of the input distribution: $y = Ax_1 + Bx_2$
- Kalman filter localization tracks the robot's belief state $p(x_t)$ typically as a single hypothesis with normal distribution.
- Kalman localization thus addresses the *position tracking problem*, but ***not*** the *global localization* or the *kidnapped robot problem*.





prediction (odometry) - ACT

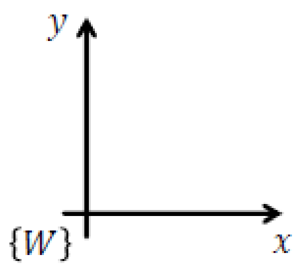
$$\hat{x}_t = f(x_{t-1}, u_t) = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin\theta + \frac{\Delta s_r - \Delta s_l}{2b} \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix}$$

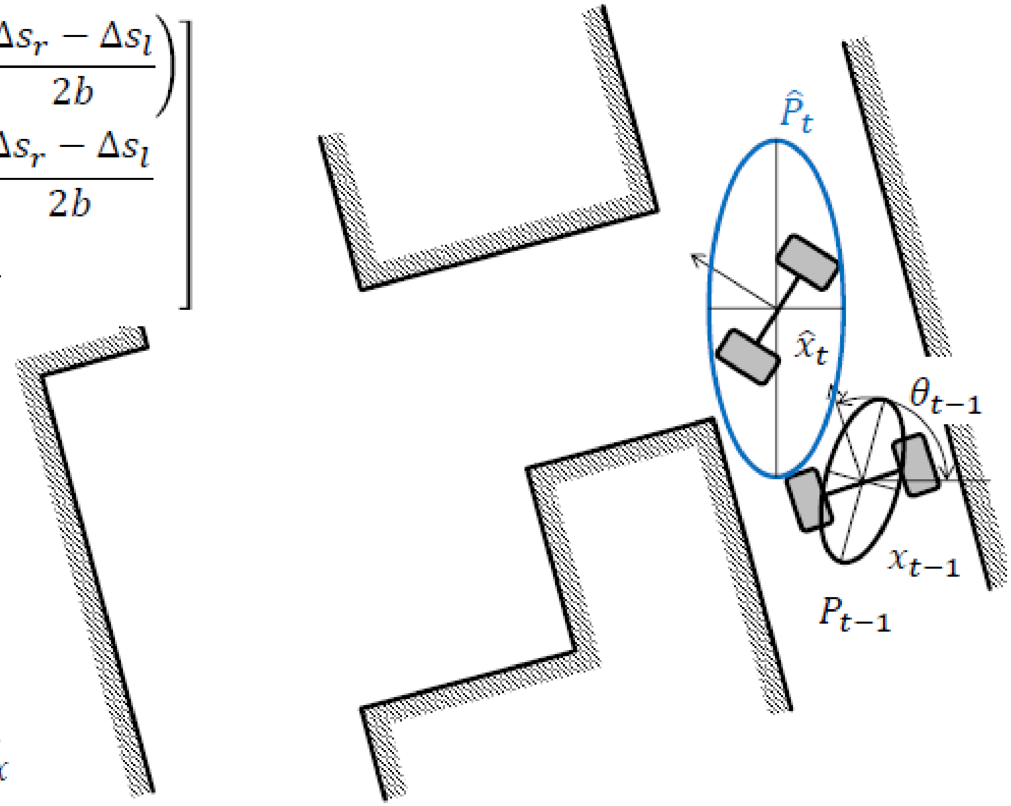
motion model

$$\hat{P}_t = F_x P_{t-1} {F_x}^T + F_u Q_t {F_u}^T$$

$$Q_t = \begin{bmatrix} k_r|\Delta s_r| & 0 \\ 0 & k_r|\Delta s_l| \end{bmatrix}$$

$F_x = \nabla_x f$ and $F_u = \nabla_x f$ represent the Jacobians of the function f with respect to x_t and u_t

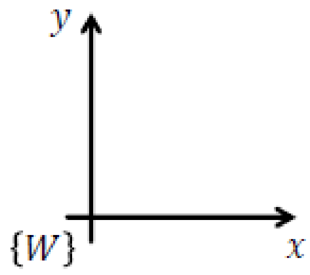


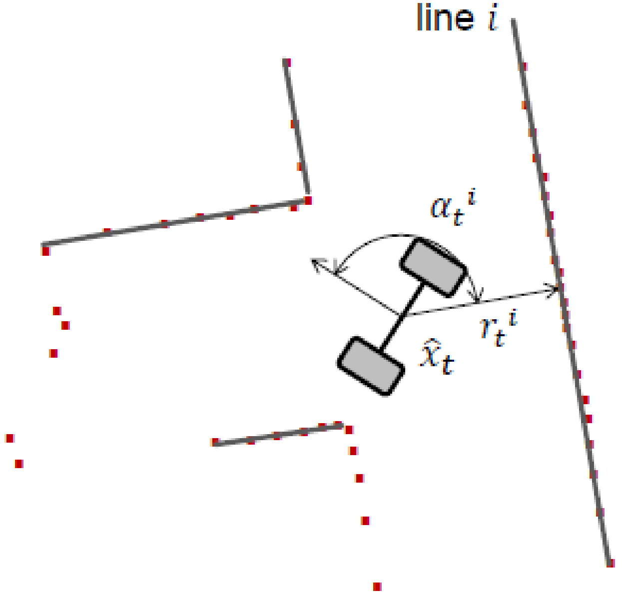


Observation - SEE

$$z_t{}^i = \begin{bmatrix} \alpha_t{}^i \\ r_t{}^i \end{bmatrix}$$

$$R_t{}^i = \begin{bmatrix} \sigma_{\alpha\alpha} & \sigma_{\alpha r} \\ \sigma_{r\alpha} & \sigma_{rr} \end{bmatrix}_t^i$$

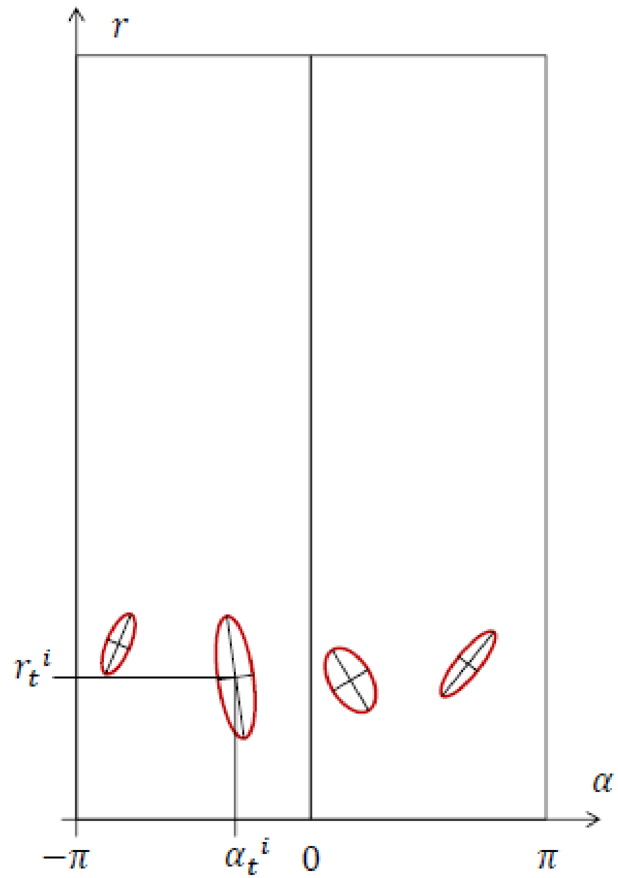


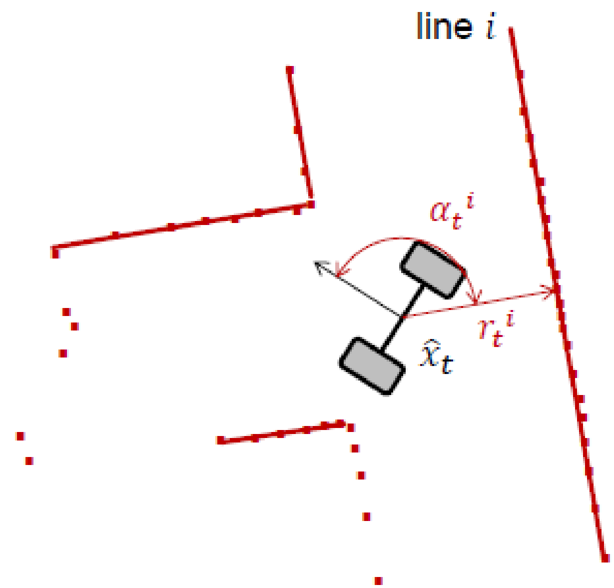


Observations in Sensor Model Space

$$z_t{}^i = \begin{bmatrix} \alpha_t{}^i \\ r_t{}^i \end{bmatrix}$$

$$R_t{}^i = \begin{bmatrix} \sigma_{\alpha\alpha} & \sigma_{\alpha r} \\ \sigma_{r\alpha} & \sigma_{rr} \end{bmatrix}_t^i$$

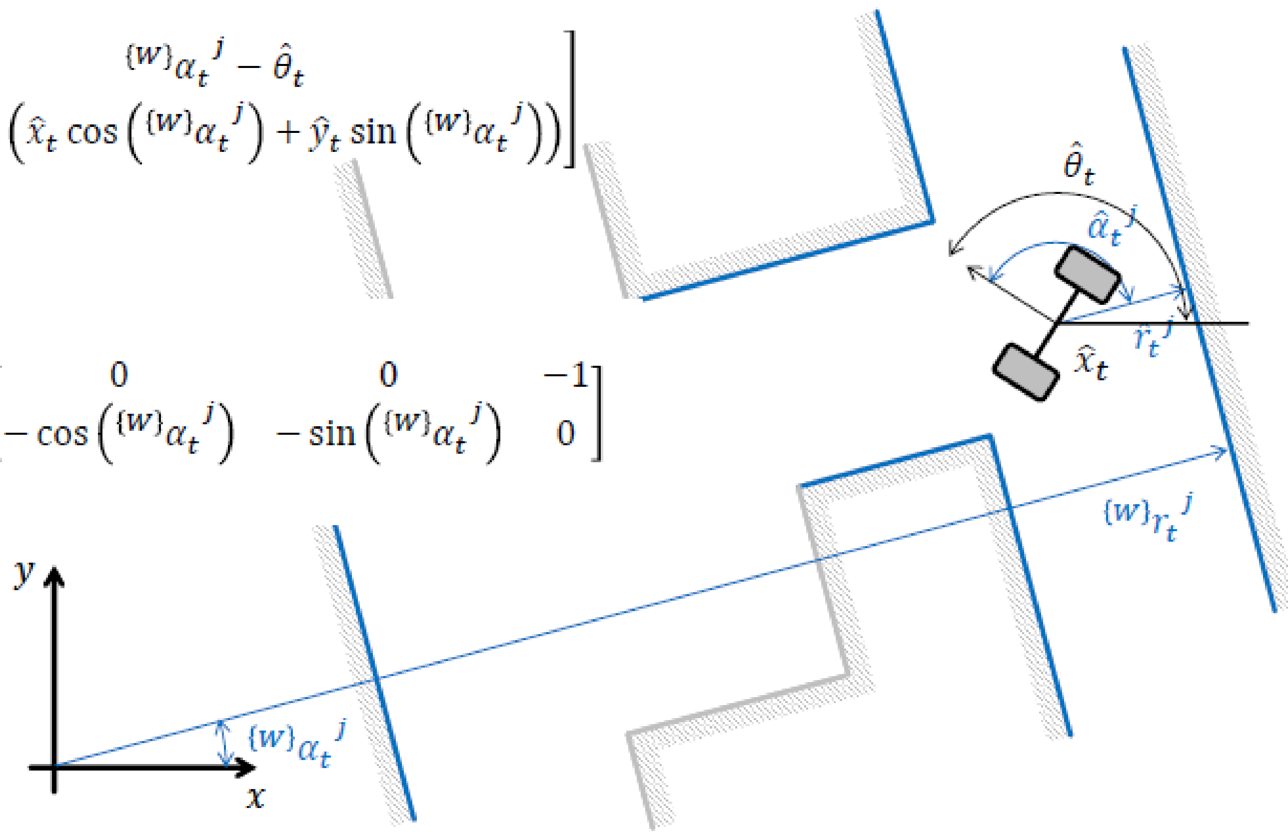




Measurement Prediction

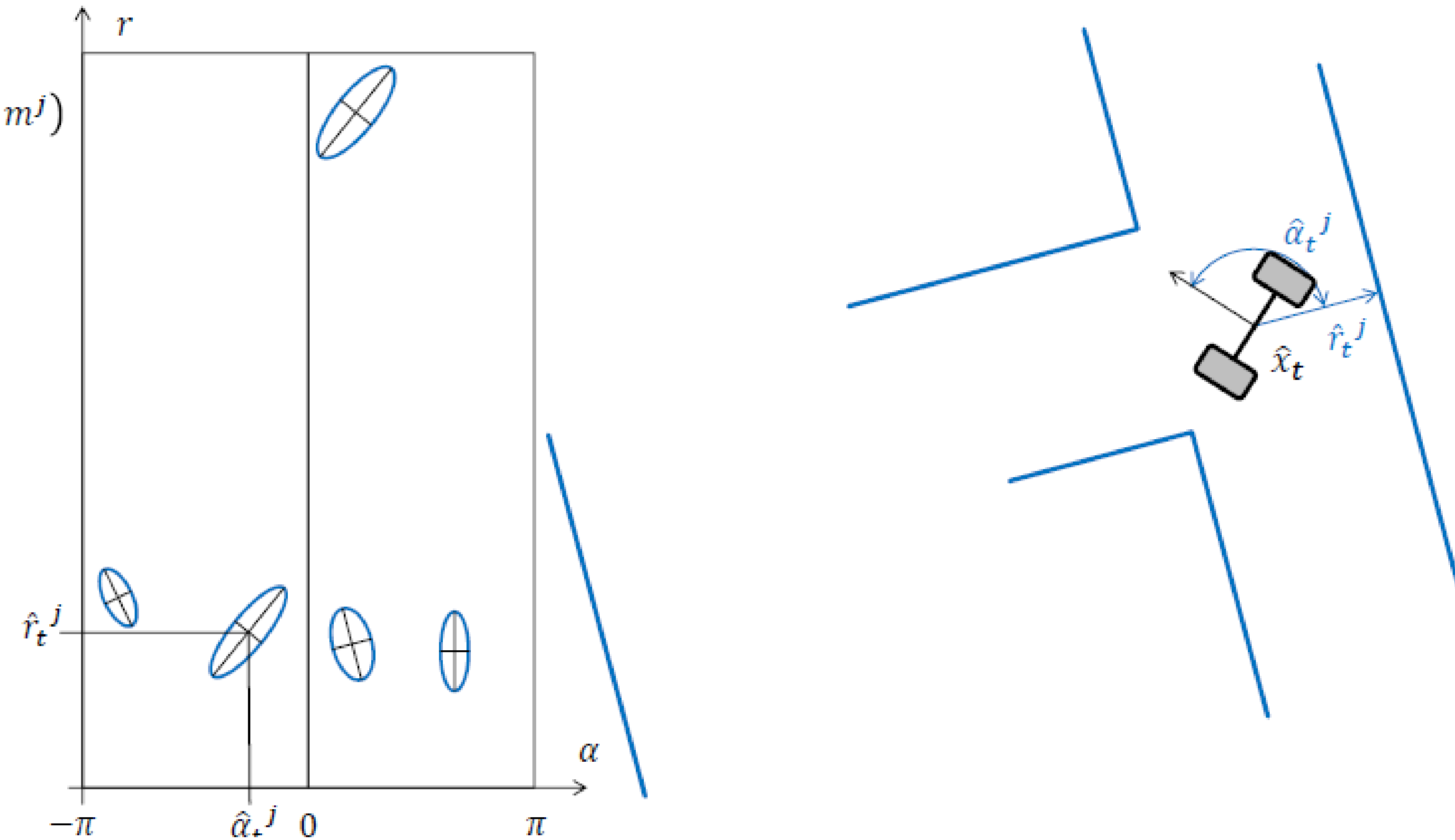
$$\hat{z}_t^{\,j} = \begin{bmatrix}\hat{\alpha}_t^{\,j}\\ \hat{r}_t^{\,j}\end{bmatrix} = h^j(\hat{x}_t, m^j) = \begin{bmatrix} {}^{\{W\}}\alpha_t^{\,j} - \hat{\theta}_t \\ {}^{\{W\}}r_t^{\,j} - \left(\hat{x}_t \cos\left({}^{\{W\}}\alpha_t^{\,j}\right) + \hat{y}_t \sin\left({}^{\{W\}}\alpha_t^{\,j}\right)\right)\end{bmatrix}$$

$$H^j = \nabla h^j = \begin{bmatrix} \frac{\partial \alpha_t^{\,i}}{\partial \hat{x}} & \frac{\partial \alpha_t^{\,i}}{\partial \hat{y}} & \frac{\partial \alpha_t^{\,i}}{\partial \hat{\theta}} \\ \frac{\partial r_t^{\,i}}{\partial \hat{x}} & \frac{\partial r_t^{\,i}}{\partial \hat{y}} & \frac{\partial r_t^{\,i}}{\partial \hat{\theta}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ -\cos\left({}^{\{W\}}\alpha_t^{\,j}\right) & -\sin\left({}^{\{W\}}\alpha_t^{\,j}\right) & 0 \end{bmatrix}$$

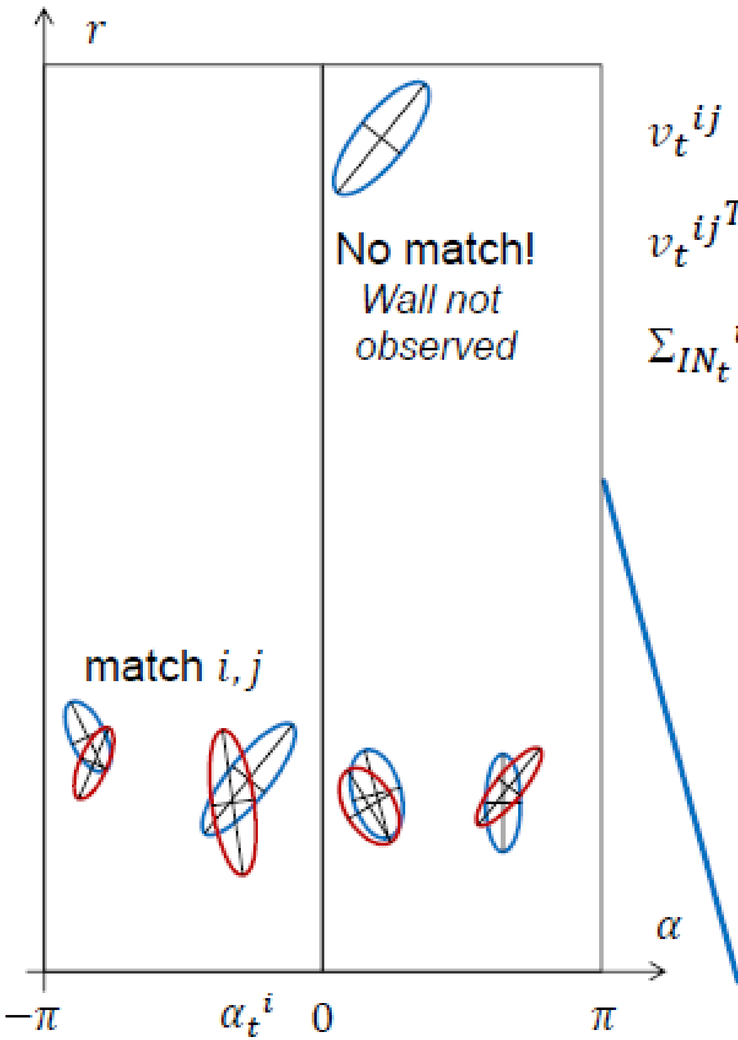


Measurement Prediction in Model Space

$$\hat{z}_t{}^j = \begin{bmatrix} \hat{\alpha}_t{}^j \\ \hat{r}_t{}^j \end{bmatrix} = h^j(\hat{x}_t, m^j)$$



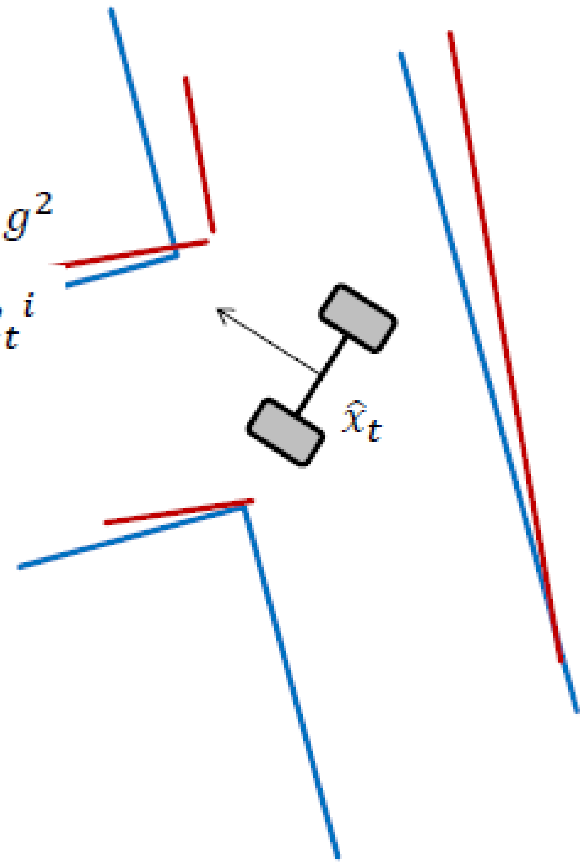
Matching in Sensor Model Space



$$v_t^{ij} = [z_t^i - \hat{z}_t^j]$$

$${v_t^{ij}}^T \cdot \left(\Sigma_{IN_t}^{ij}\right)^{-1} \cdot v_t^{ij} \le g^2$$

$$\Sigma_{IN_t}^{ij} = H^j \cdot \hat{P}_t \cdot {H^j}^T + R_t^i$$



Estimation

- For each found match we can now estimate an position update:

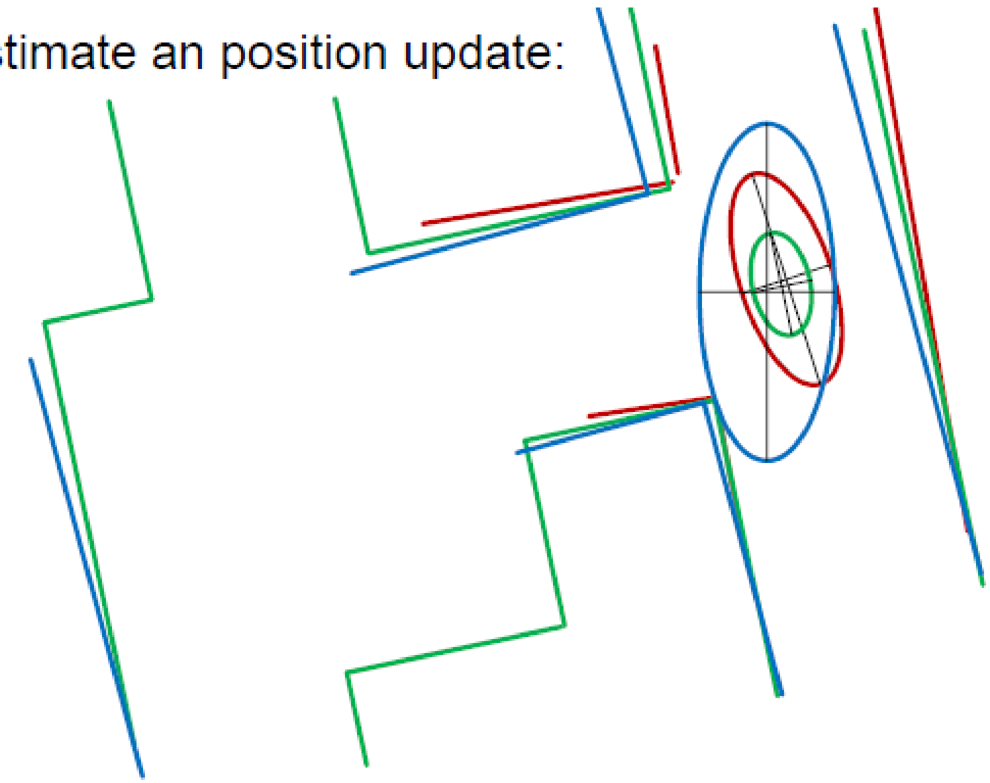
$$x_t = \hat{x}_t + K_t v_t$$

- where $K_t = \hat{P}_t H_t^{\,T} (\Sigma_{IN_t})^{-1}$

  is the Kalman gain

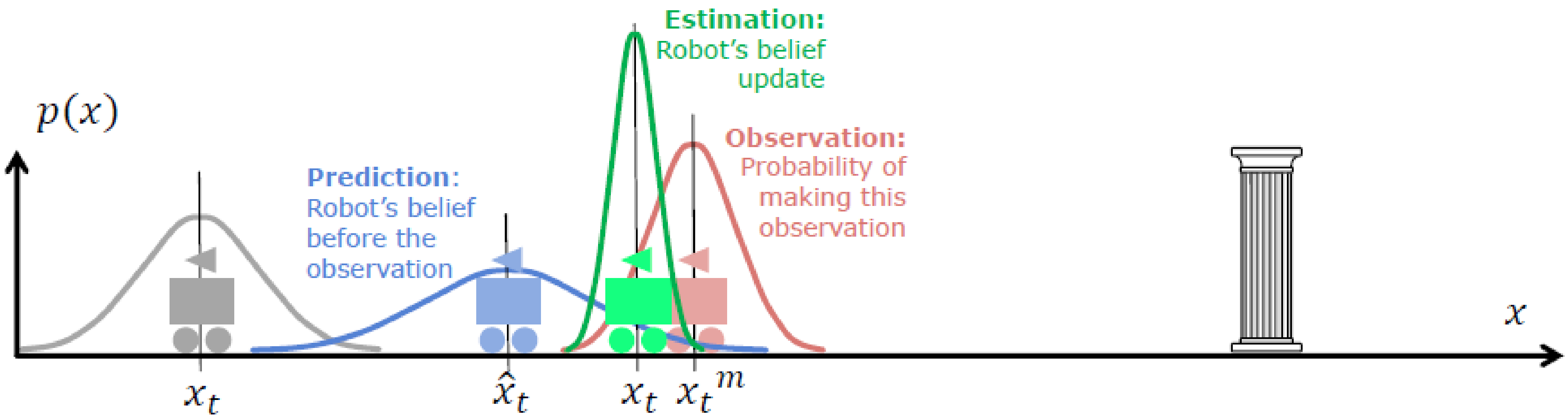
- and the corresponding position covariance P_t :

$$P_t = \hat{P}_t - K_t \Sigma_{IN_t} K_t^{\,T}$$



Kalman Filter Localization in summery

1. **Prediction (ACT)** based on previous estimate and odometry
2. **Observation (SEE)** with on-board sensors
3. **Measurement prediction** based on prediction and map
4. **Matching** of observation and map
5. **Estimation** → position update (posteriori position)



Questions