CIE 482: Path Planning & Navigation for Mobile Robots

Lecture 15: Bayesian Filtering Theory

جامعة الملك فهد للبترول والمعادن
King Fahd University of Petroleum & Minerals

Ramy Rashad
Assistant Professor
CIE Department

Outline

- Recap last lecture
- Bayes Rule
- Bayesian Filtering

Outline

- Recap last lecture
- Bayes Rule
- Bayesian Filtering

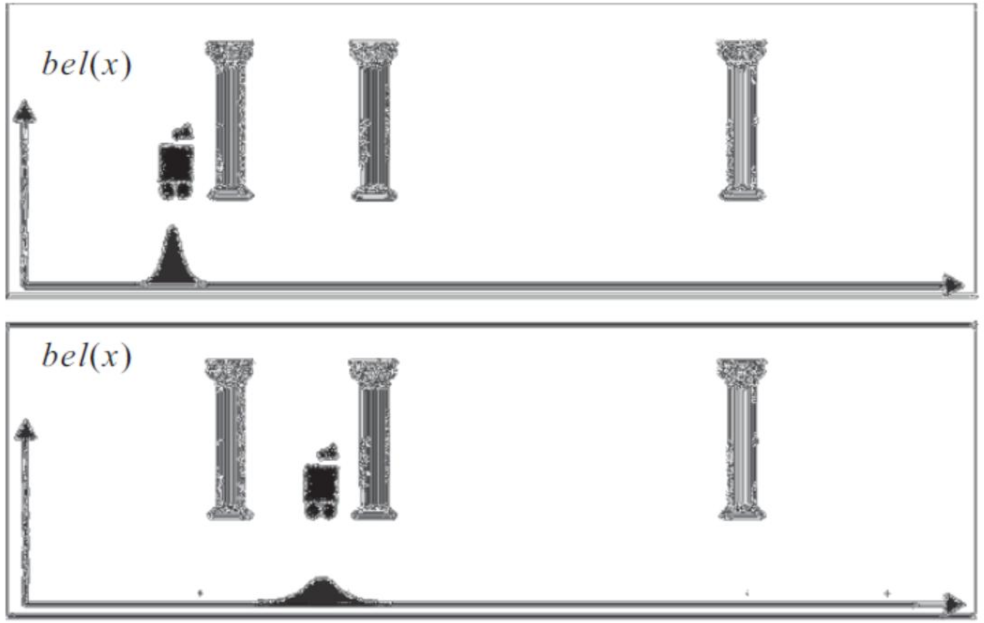
Recap: Belief Representation

- We represent the robot not just using a single possible position but also a possibly infinite set of positions.
- In this approach, the belief is modeled by a probability distribution function $bel(x)$.
- In this approach, beliefs are

represented by conditional

probability distributions:

$$bel(x) = p(x|y)$$



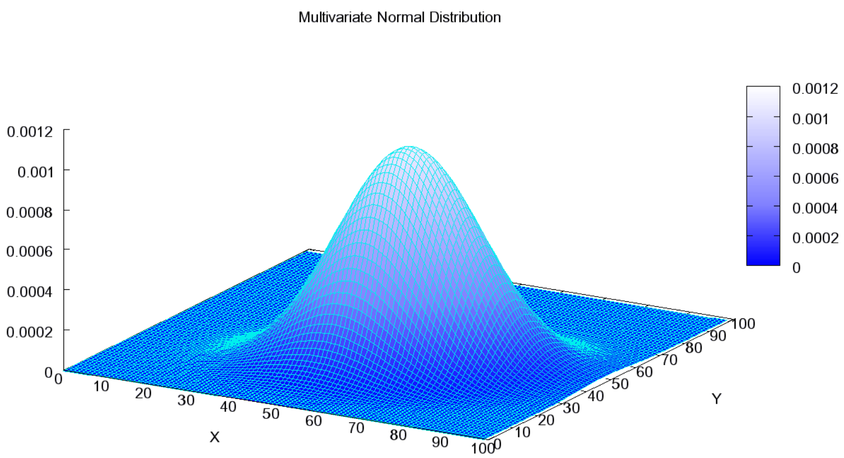
The belief function $bel(x)$ is our best guess of the robot's state x.

Recap: Multivariate Gaussian Distribution

- In case the random variable $x \in \mathbb{R}^k$ is a k-dimensional vector, we will have a multivariate normal distribution characterized by the density function:

$$p(x) = \frac{1}{(2\pi)^{k/2}\sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

where $\mu \in \mathbb{R}^k$ & $\Sigma \in \mathbb{R}^{k\times k}$ denote the mean & covariance matrix.



Recap: Joint Probability

- The joint distribution of two random variables X and Y is given by

$$p(X = x, Y = y) \quad \text{or simply} \quad \boldsymbol{p(x, y)}$$

which describes the probability that the random variable X takes on the value x and that Y takes on the value y.

- Discrete case, the probabilities must sum to 1:

$$\sum_x \sum_y p(x, y) = 1$$

- Continuous case, the density must integrate to 1 over all possible x and y:

$$\int \int p(x, y) dx\, dy = 1$$

Recap: Marginal Probability

- Marginal Probability refers to the probability of a single event occurring, without consideration of any other events.
- It is derived from a joint probability distribution and represents the likelihood of an event happening in isolation.

- Discrete case:

$$p(x) = \sum_y p(x,y) \qquad\qquad p(y) = \sum_x p(x,y)$$

- Continuous case:

$$p(x) = \int p(x,y)dy \qquad\qquad p(y) = \int p(x,y)dx$$

Recap: Conditional Probability

- The conditional probability describes the probability that the random variable X takes on the value x conditioned that Y takes on the value y (i.e., we already know that $Y = y$).
- We denote this conditional probability by

$$p(X = x|Y = y) \quad \text{or simply} \quad \boldsymbol{p(x|y)}$$

- Intuitively, conditional probability tells us how our belief about X changes once we have observed (or "condition on") Y.

Recap: Conditional Probability

- It is related to the joint probability by the formula:

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

which reads as:

The conditional probability of x given y, is equal to their joint probability divided by the marginal probability of y.

Outline

- Recap last lecture
- Bayes Rule
- Bayesian Filtering

Bayes rule

- Consider the conditional probability $p(x|y) = \frac{p(x,y)}{p(y)}$
- We also have that $p(y|x) = \frac{p(x,y)}{p(x)}$
- Therefore, we have that

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

- This simple, yet powerful result is a fundamental result in probability theory called Bayes' rule/theorem.

Bayes rule

- Bayes' rule provides a mathematically rigorous way to update our belief in X once we have observed some new information Y.
- It underpins many applications in statistics, machine learning, and robotics where we use data (evidence) to refine our uncertainty about unknown states of the world.

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

Bayes rule

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

- $p(x)$ is our prior probability of $X = x$ (before incorporating any external data).
- $p(x|y)$ is our posterior probability of $X = x$ after considering the data $Y = y$.
- $p(y|x)$ is the likelihood that we see the data $Y = y$ if $X = x$ is true.
- $p(y)$ is the probability of the evidence

Bayes rule

- It is common to write the Bayes rule as

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

where η is a normalization factor (equal to $\frac{1}{p(y)}$).

Note that we can directly compute η by remembering that the integral of a probability distribution should be always 1.

Bayes rule

- Therefore, in conclusion,

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

- Bayes' rule relates the posterior probability $p(x|y)$ of an event $X = x$ after observing some evidence $Y = y$ to:
  1. The prior probability $p(x)$ of that event,
  2. The likelihood $p(y|x)$ of the evidence under that hypothesis.

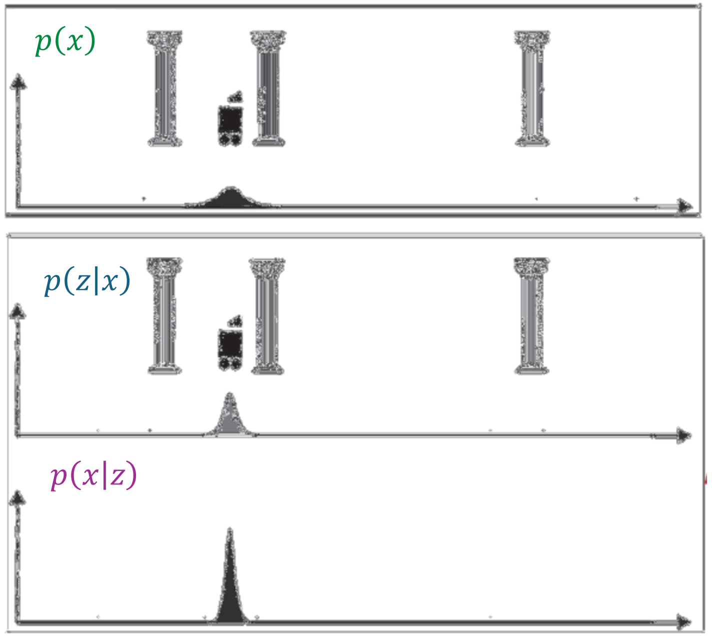
Robotics Example: Bayesian Localization

- Hypothesis X: The robot is at position x.
- Evidence Z: The robot observes sensor reading z.

Sensor Model

- Prior belief $p(x)$ is our prior belief about the robot's position.
- Likelihood $p(z|x)$ is the likelihood of measuring z if the robot really is at x.

- How to combine both information ?



Robotics Example: Bayesian Localization

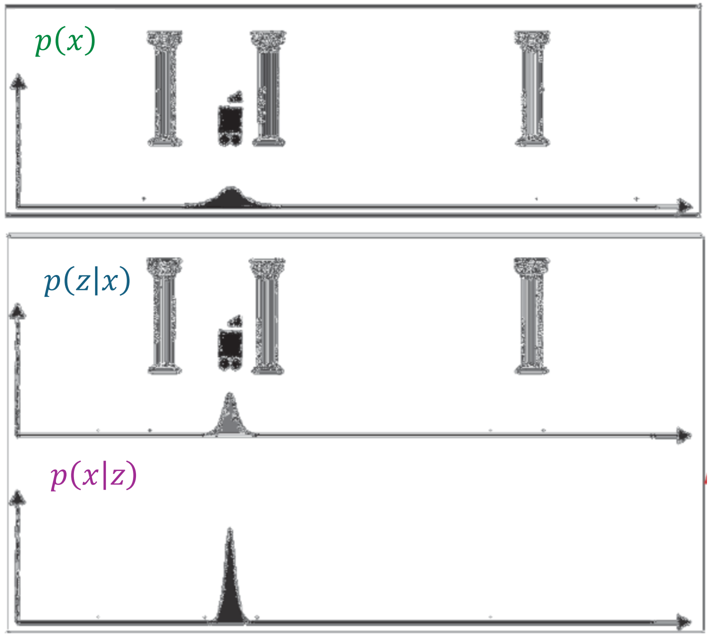
- Hypothesis X: The robot is at position x.
- Evidence Z: The robot observes sensor reading z.

- Prior belief $p(x)$ is our prior belief about the robot's position.
- Likelihood $p(z|x)$ is the likelihood of measuring z if the robot really is at x.

- Bayes rule combines both information using

$$\boxed{p(x|z) = \eta\, p(z|x)p(x)}$$

where $p(x|z)$ is our updated (posterior) belief about the robot's position after considering the measurement.

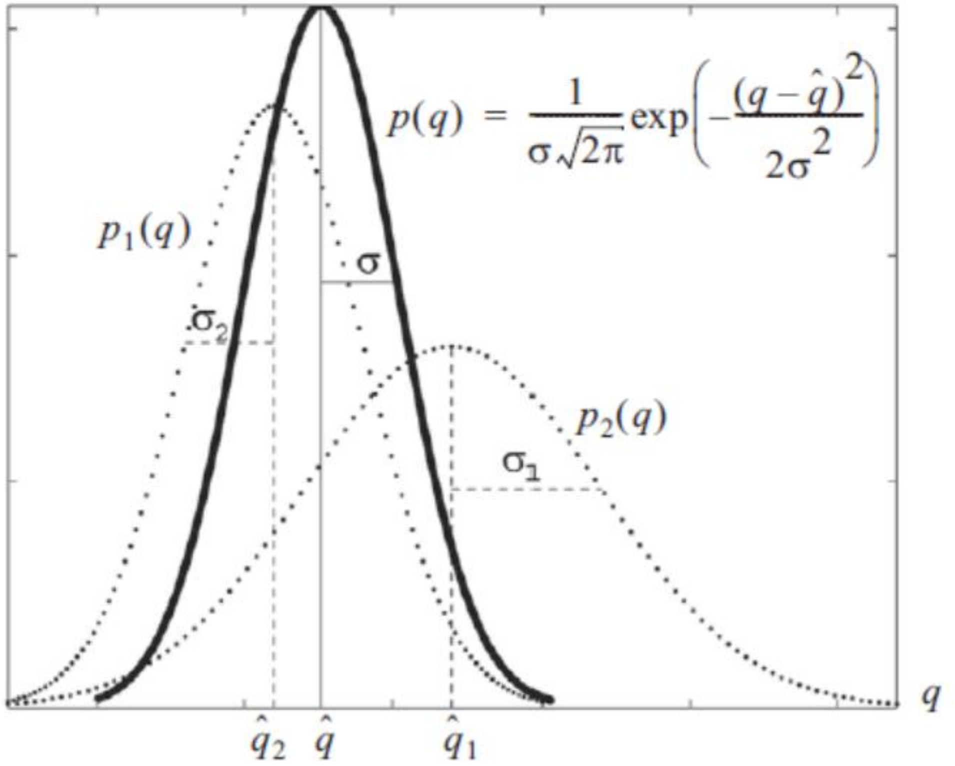


Special Case: Gaussian Distributions

- Bayes Rule is simply the multiplication of two probabilities normalized such that the result is another probability.
- In case:
  - The prior $p(x)$ (predicted state) is a Gaussian.
  - The likelihood $p(z|x)$ (measurement model) is also a Gaussian.
  - The product of two Gaussian functions is another Gaussian function.

Special Case: Gaussian Distributions

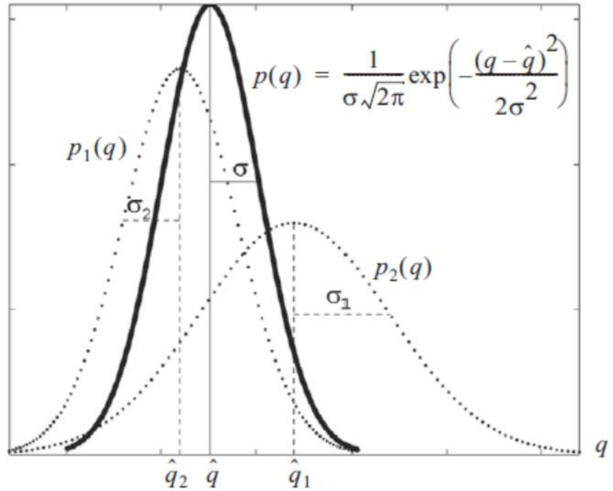
- Let $p_1(q) = N(\hat{q}_1, \sigma_1^2)$ and $p_2(q) = N(\hat{q}_2, \sigma_2^2)$ be two Gaussian distribution functions.
- Their product $p(q) \coloneqq p_1(q) * p_2(q) = N(\hat{q}, \sigma^2)$ is another Gaussian with mean and variance given by
  - $\hat{q} = \frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2}\hat{q}_1 + \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}\hat{q}_2$
  - $\sigma^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}$



Why is this important ?

- **Key Implication: Uncertainty is Reduced by Fusion**
  - The product of two Gaussian beliefs gives a more certain estimate because the resulting variance is smaller than either original variance.
  - Lower variance means lower uncertainty, so the fused estimate is more confident.
  - The fused mean lies between the two original means, balancing the two sources of information.

> This is the key idea behind **sensor fusion** and **Kalman filtering**: multiple uncertain measurements can produce a better estimate than either one alone.



Outline

- Recap last lecture
- Bayes Rule
- Bayesian Filtering

Bayesian Filtering

- Filtering is concerned with the sequential process of maintaining a probabilistic model for a state x_t which evolves over time and which is periodically observed by a sensor.
- The general filtering problem can be formulated in Bayesian form.

Bayesian Filtering

- Filtering is concerned with the sequential process of maintaining a probabilistic model for a state x_t which evolves over time and which is periodically observed by a sensor.
- The general filtering problem can be formulated in Bayesian form.
- For robot localization, the state we are interested in filtering is the robot's state x_t.
- For wheeled robots, $x_t = \left(\xi_x, \xi_y, \theta\right)$
- For aerial robots $x_t = (\xi, R)$

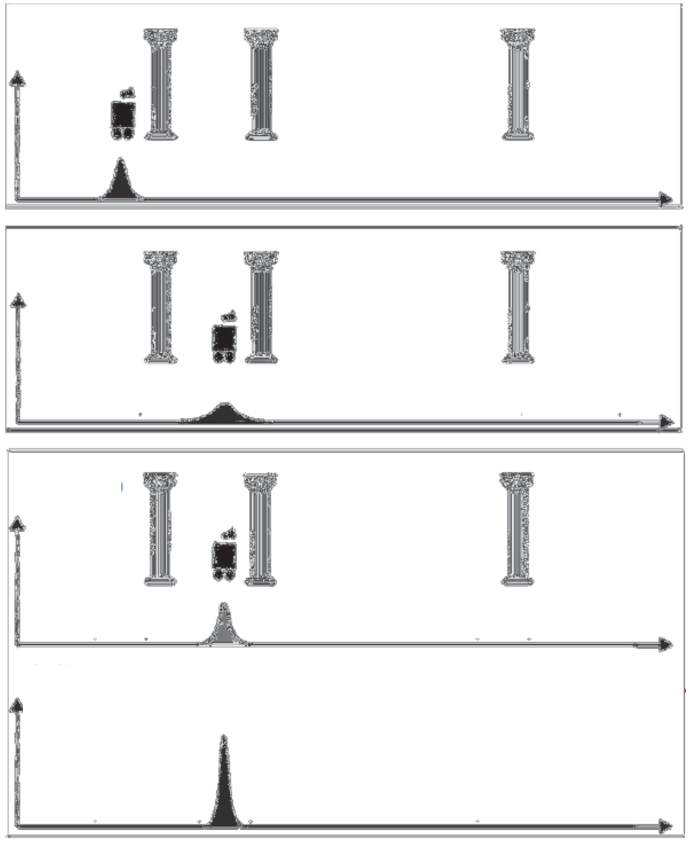
- From now on we consider time defined as discrete (asynchronous) times $t_k =: k$.

State, Input and Observation

- At a time instant k, we define the following:
  - x_k: The state vector to be estimated at time k.
  - u_k: The control vector applied at time $k-1$ to drive the state from x_{k-1} to x_k at time k.
  - z_k : An observation taken of the state x_k at time k.

State, Input and Observation

- At a time instant k, we define the following:
  - x_k: The state vector to be estimated at time k.
  - u_k: The control vector applied at time $k-1$ to drive the state from x_{k-1} to x_k at time k.
  - z_k : An observation taken of the state x_k at time k.



$\boldsymbol{p(x_0)}$

Robot belief at $k=0$
$bel(x_0)$

$\boldsymbol{p(x_1|u_1)}$

$\boldsymbol{p(z_1|x_1)}$

$\boldsymbol{p(x_1|z_1,u_1)}$

Robot belief at $k=1$
$bel(x_1)$

State, Input and Observation

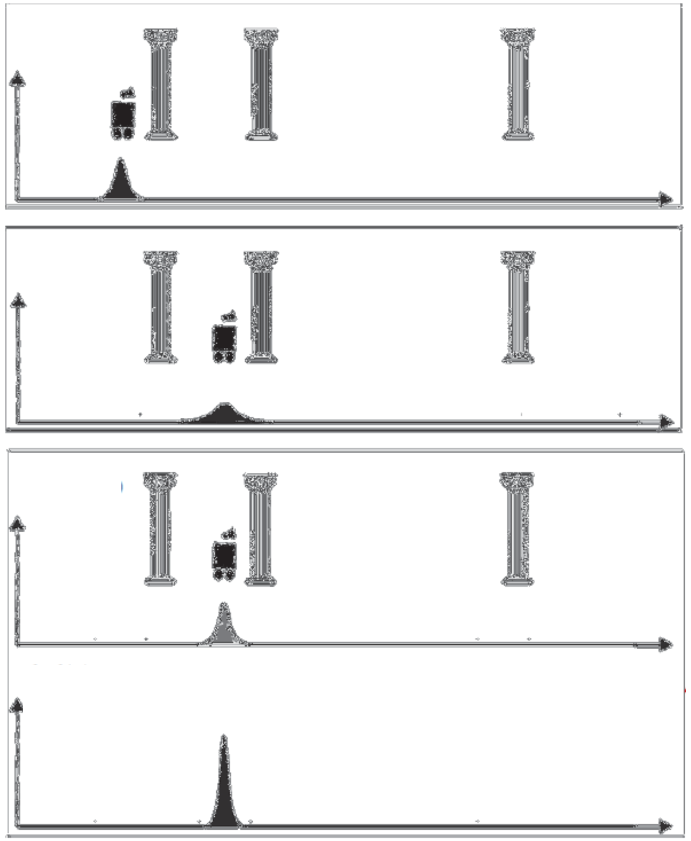
- At a time instant k, we define the following:
  - x_k: The state vector to be estimated at time k.
  - u_k: The control vector applied at time $k-1$ to drive the state from x_{k-1} to x_k at time k.
  - z_k : An observation taken of the state x_k at time k.

$\boldsymbol{p(x_0)}$

$\boldsymbol{p(x_1|u_1)}$

$\boldsymbol{p(z_1|x_1)}$

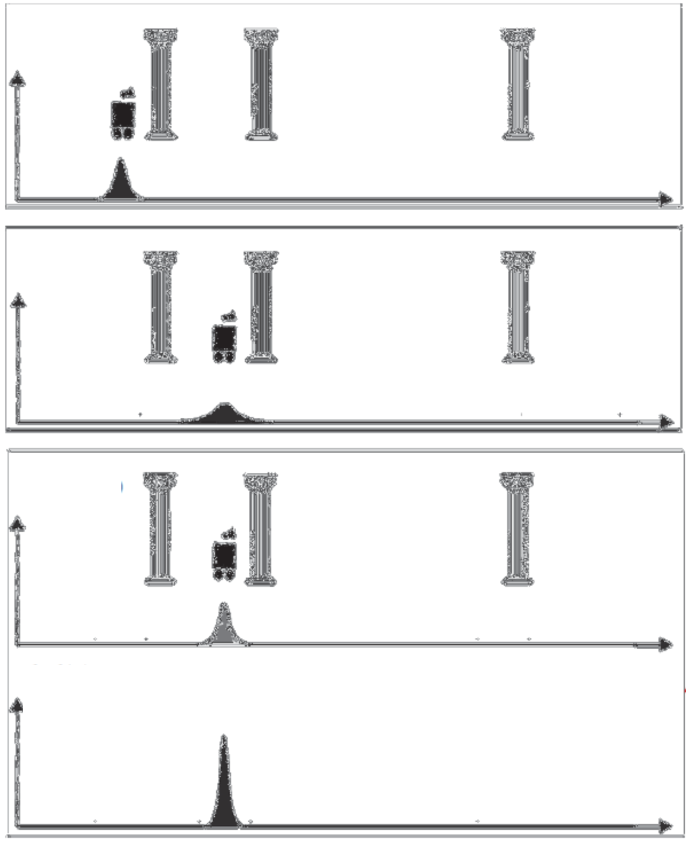
$\boldsymbol{p(x_1|z_1, u_1)}$



In practice, at $k = 0$, we either assume $p(x_0)$ is:
- A **uniform** distribution, i.e., the robot's initial state is unknown.
- A **Dirac delta** distribution, i.e., the robot's initial state is perfectly known

State, Input and Observation

- At a time instant k, we define the following:
  - x_k: The state vector to be estimated at time k.
  - u_k: The control vector applied at time $k-1$ to drive the state from x_{k-1} to x_k at time k.
  - z_k : An observation taken of the state x_k at time k.



A common approach for sensor fusion is to use proprioceptive sensors as u_k and exteroceptive sensors as z_k

History

- The robot's path / history of states is given by
$$X_k := \{x_0, x_1, \cdots, x_{k-1}, x_k\} = \{X_{k-1}, x_k\}$$
- The history of control inputs (or proprioceptive sensor readings)
$$U_k := \{u_1, \cdots, u_{k-1}, u_k\} = \{U_{k-1}, u_k\}$$
- The history of observations (or exteroceptive sensor readings)
$$Z_k := \{z_1, \cdots, z_{k-1}, z_k\} = \{Z_{k-1}, z_k\}$$

Recursive Bayesian Filtering

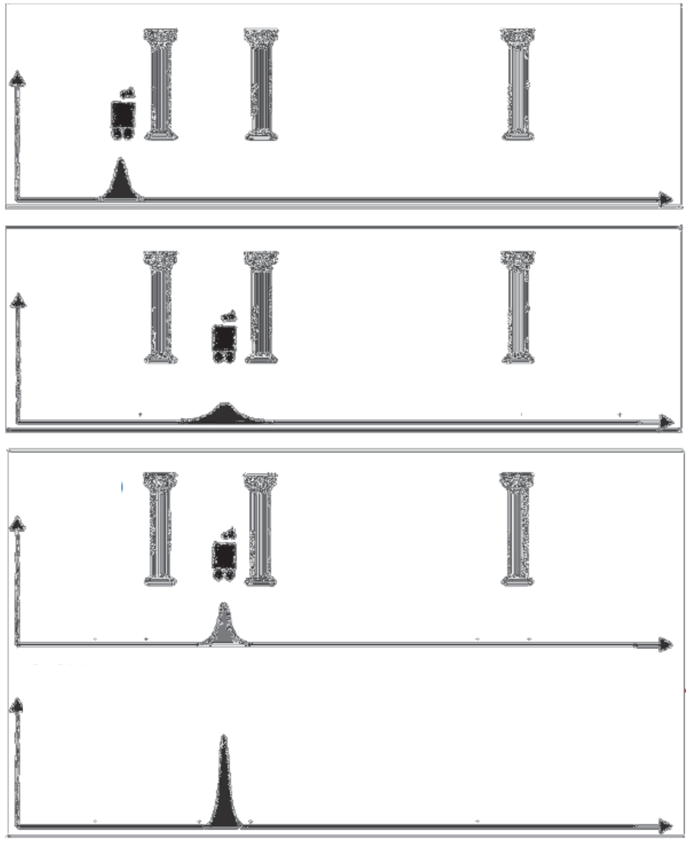
- In probabilistic form, the general Bayesian filtering problem is to find at time k the posterior function $\boldsymbol{p(x_k|Z_k, U_k)}$ given all recorded observations and control inputs.

$$p(x_{k-1}|Z_{k-1}, U_{k-1})$$

$$p(x_k|Z_{k-1}, U_k)$$

$$p(z_k|x_k)$$

$$p(x_k|Z_k, U_k)$$



Recursive Bayesian Filtering

- In probabilistic form, the general Bayesian filtering problem is to find at time k the posterior function $p(x_k|Z_k, U_k)$ given all recorded observations and control inputs.

$p(x_{k-1}|Z_{k-1}, U_{k-1})$

1 **Prediction update using u_k**

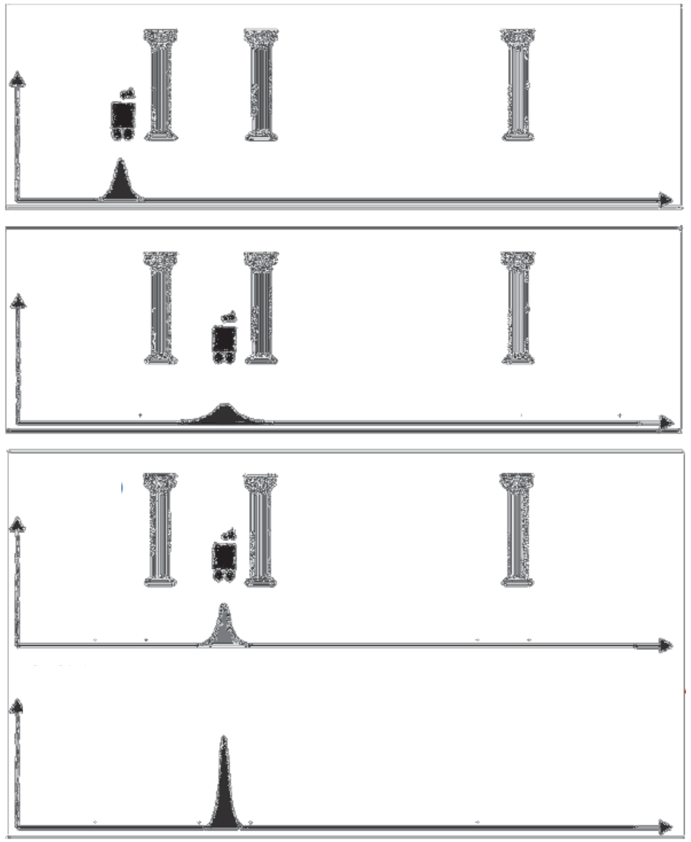
$p(x_k|Z_{k-1}, U_k)$

+

$p(z_k|x_k)$

2 **Correction update using z_k**

$p(x_k|Z_k, U_k)$



Recursive Bayesian Filtering

- In probabilistic form, the general Bayesian filtering problem is to find at time k the posterior function $p(x_k|Z_k, U_k)$ given all recorded observations and control inputs.

$bel(x_{k-1})$

1 **Prediction update using u_k**

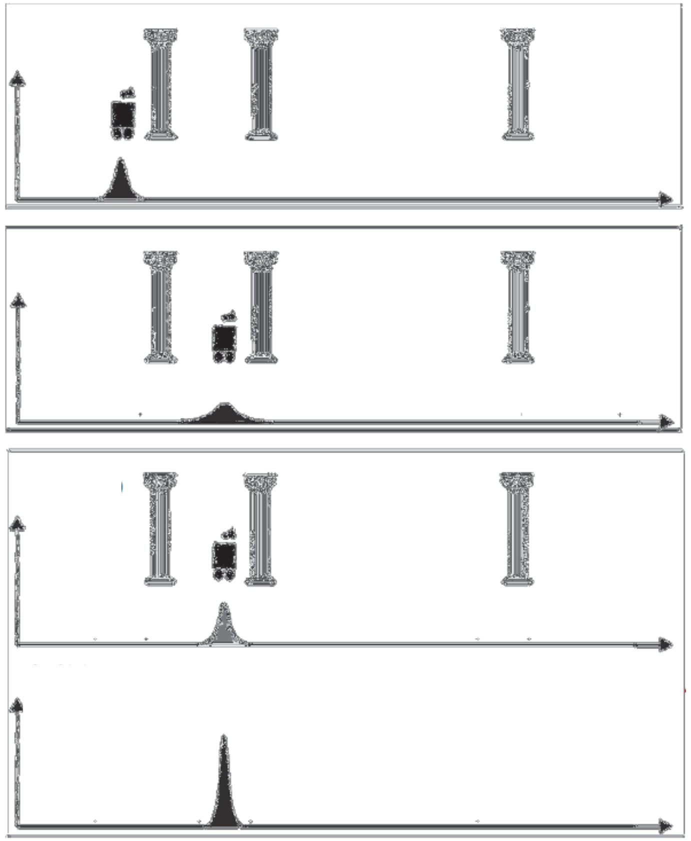
$p(x_k|Z_{k-1}, U_k)$

+

$p(z_k|x_k)$

2 **Correction update using z_k**

$bel(x_k)$

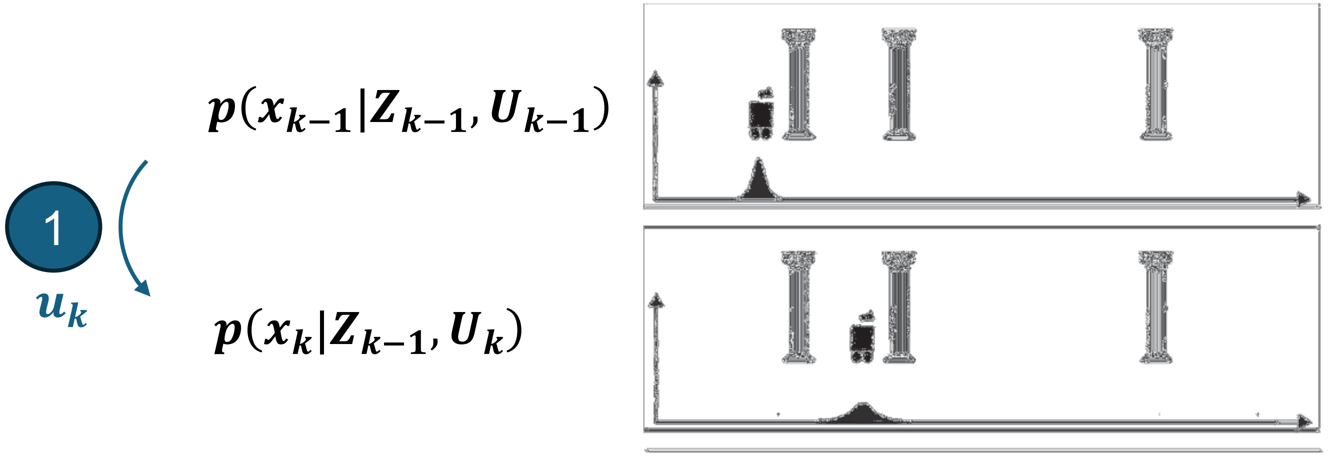


Prediction Update

- We use in this step a **state transition model**

$$p(x_k|x_{k-1},u_k)$$

- The state transition model describes how the system state evolves from one time step to the next.



Prediction Update

- We use in this step a **state transition model**

$$p(x_k | x_{k-1}, u_k)$$

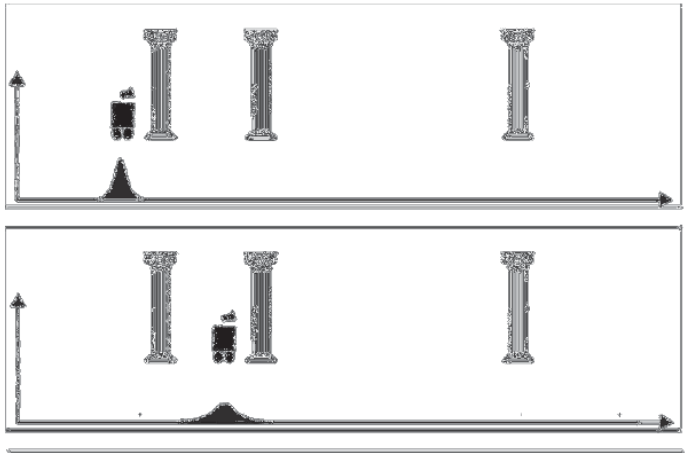
- Key idea: If I know the state x_{k-1} at previous time step and current input u_k, how do I predict the next value of the state x_k

1

u_k

$p(x_{k-1}|Z_{k-1}, U_{k-1})$

$p(x_k|Z_{k-1}, U_k)$

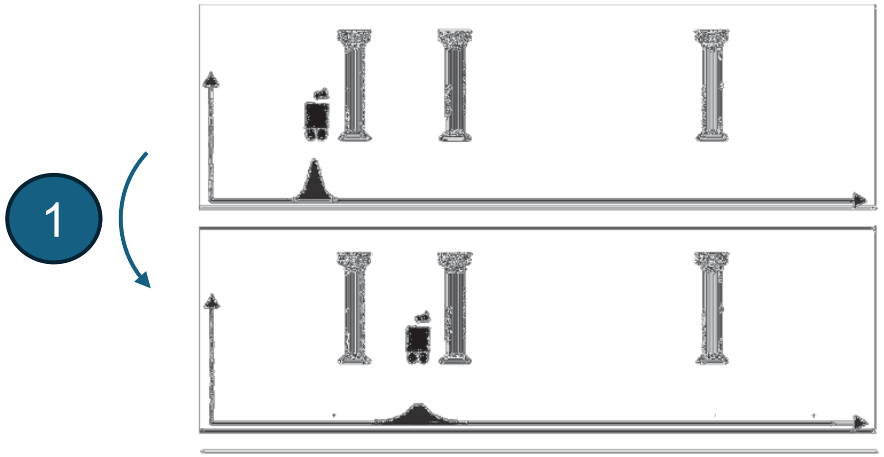


Prediction Update

- We use in this step a **state transition model**

$$p(x_k | x_{k-1}, u_k)$$

- Key idea: If I know the state x_{k-1} at previous time step and current input u_k, how do I predict the next value of the state x_k

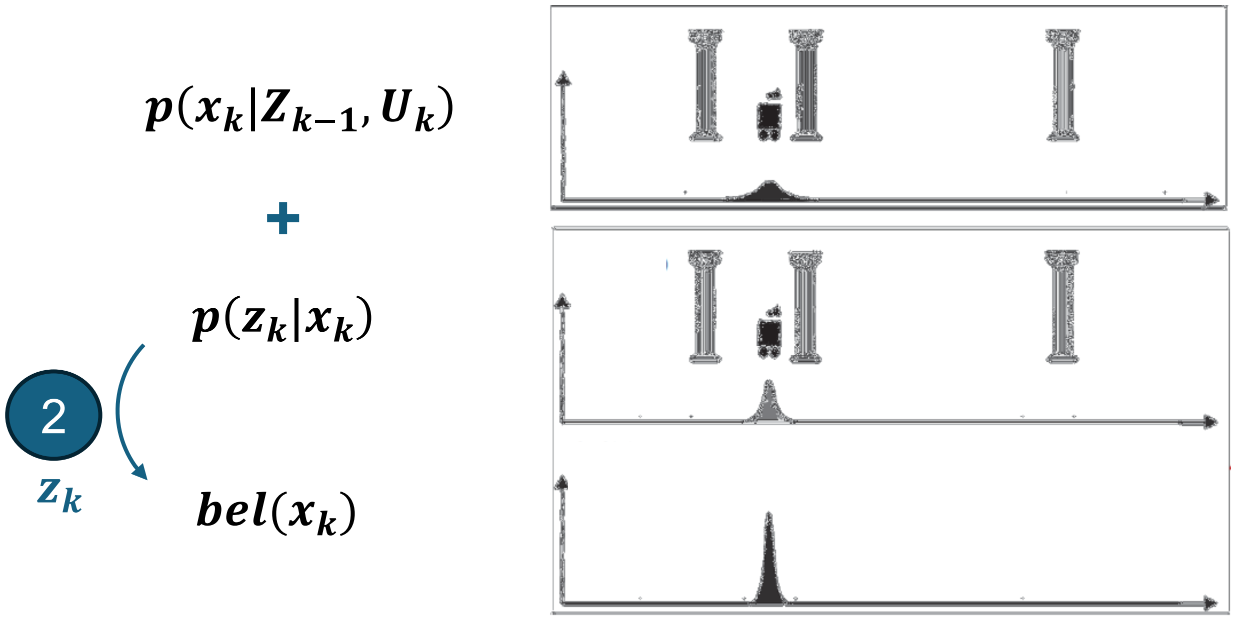


Example

If a robot is at one location and moves forward for one second, the transition model predicts where it should be next. Because real systems are not perfectly predictable, the model also includes uncertainty.

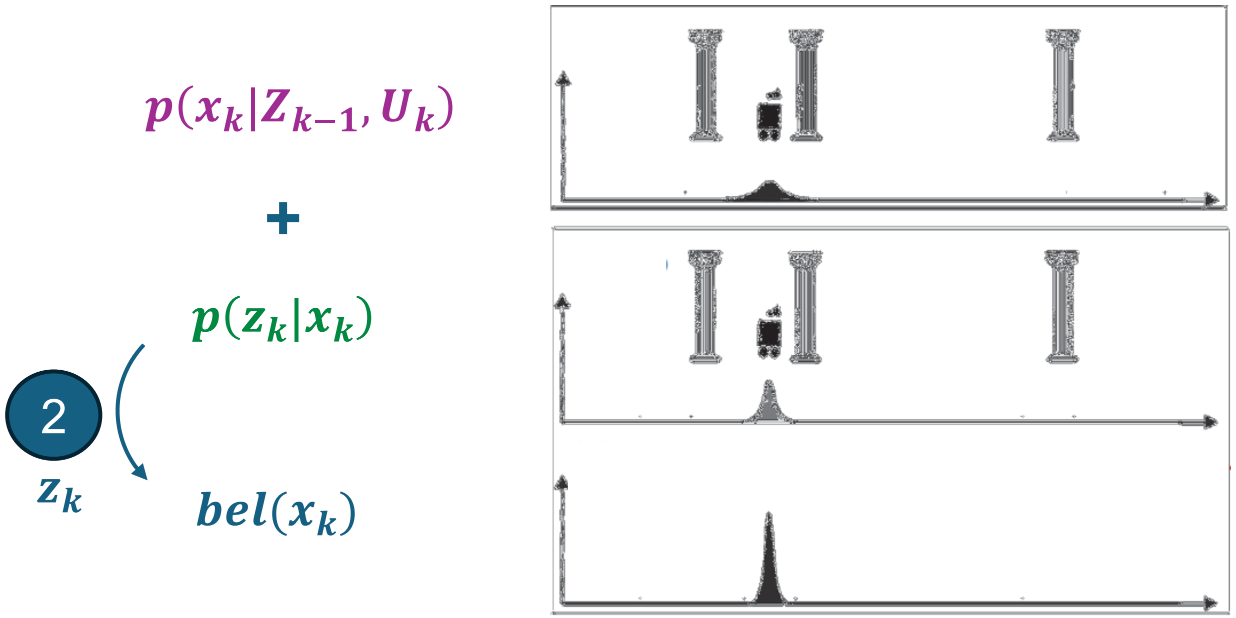
Correction Update

- In this step, we use the new measurement to refine the predicted state.
- The correction update combines the prior belief with the information contained in the new measurement.
- This is done using Bayes Rule.



Correction Update

- Prior Belief $p(x_k|Z_{k-1}, U_k)$:
  - Where the state transition model thinks the robot should be ?
- Likelihood $p(z_k|x_k)$:
  - How much we should trust the sensor’s reading about where the robot is ?
- Posterior $bel(x_k)$:
  - Combination of both



Correction Update

- Prior Belief $p(x_k|Z_{k-1}, U_k)$:
  - Where the state transition model thinks the robot should be ?
- Likelihood $p(z_k|x_k)$:
  - How much we should trust the sensor's reading about where the robot is ?
- Posterior $bel(x_k)$:
  - Combination of both

- If the sensor is very reliable, the estimate moves strongly toward the measurement.
- If the sensor is noisy, the estimate stays closer to the prediction.

$\boldsymbol{p(x_k|Z_{k-1}, U_k)}$

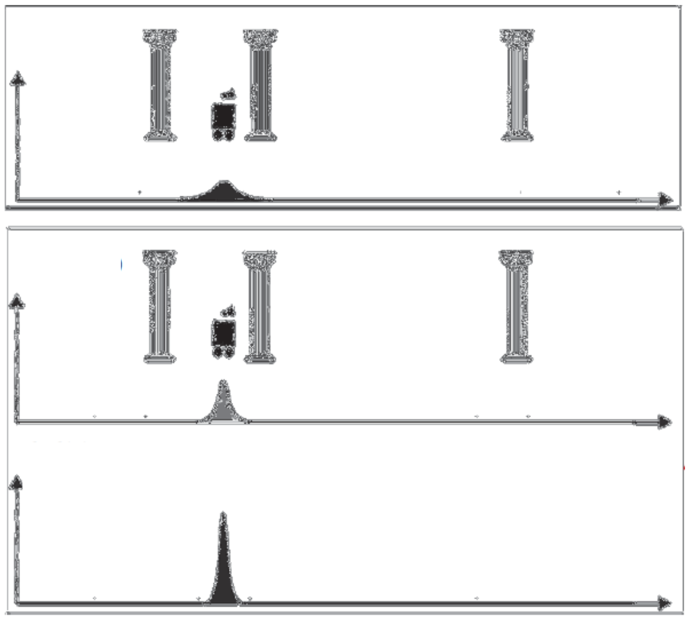
$\boldsymbol{+}$

$\boldsymbol{p(z_k|x_k)}$

2

$\boldsymbol{z_k}$

$\boldsymbol{bel(x_k)}$



Summary of Bayesian Filtering

u_k

State Transition Model

$p(x_k | x_{k-1}, Z_{k-1}, U_k)$

1 **Prediction update**

$\boldsymbol{bel}\,(\boldsymbol{x_{k-1}}) = p(x_{k-1} | Z_{k-1}, U_{k-1})$

$\overline{\boldsymbol{bel}}\,(\boldsymbol{x_k}) = p(x_k | Z_{k-1}, U_k)$

$\boldsymbol{bel}\,(\boldsymbol{x_k}) = p(x_k | Z_k, U_k)$

Correction update 2

Sensor Model

$p(z_k | x_k)$

z_k