HCC Part 4 Lecture 1
Uncertainty & Probabilistic Robotics

Chieh-Chih (Bob) Wang 王傑智

2025 Spring

With slides by Sebastian Thrum, Roland Seigwart, & Illah R. Nourbakhsh

Today’s Goals

- Uncertainty & Probabilistic Robotics
- Uncertainty Presentation & Propagation
- Introduction to Kalman Filter

Uncertainty

- Sensing is always related to uncertainties.
  - What are the sources of uncertainties?
  - How can uncertainty be represented or quantified?
  - **How do they propagate - uncertainty of a function of uncertain values?**
  - **How do uncertainties combine if different sensor reading are fused?**
- Some definitions:
  - Sensitivity: G=out/in
  - Resolution: Smallest change which can be detected
  - Dynamic Range: $value_{max}$/ resolution (10^4 -10^6)
  - Accuracy: $error_{max}$= (measured value) - (true value)
- Errors are usually unknown:

  deterministic non deterministic (random)

Uncertainty

- Environments, Action & Cognition are also always related to uncertainties.

- Probabilistic Robotics
  - Key idea:
    Explicit representation of uncertainty using the calculus of probability theory
  - Perception = state estimation
  - Action = utility optimization

Axioms of Probability Theory

Pr*(A)* denotes probability that proposition *A* is true.

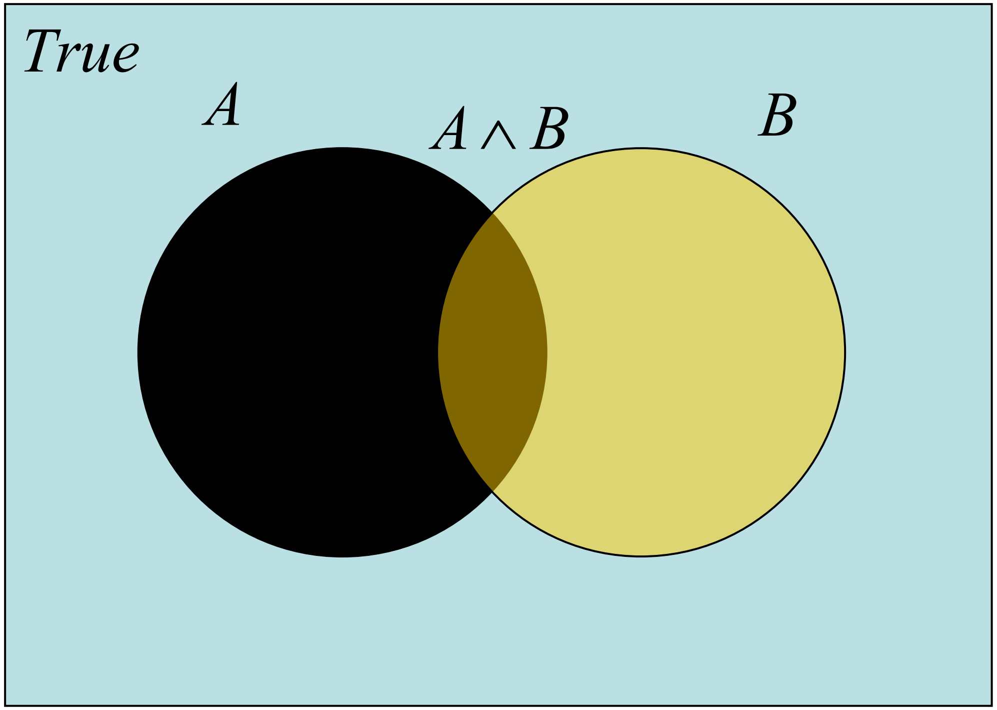
$$0 \leq \Pr(A) \leq 1$$

$$\Pr(True) = 1 \qquad \Pr(False) = 0$$

$$\Pr(A \vee B) = \Pr(A) + \Pr(B) - \Pr(A \wedge B)$$

A Closer Look at Axiom 3

$$\Pr(A \vee B) = \Pr(A) + \Pr(B) - \Pr(A \wedge B)$$



Using the Axioms

$$\begin{aligned}
\Pr(A \vee \neg A) &= \Pr(A) + \Pr(\neg A) - \Pr(A \wedge \neg A) \\
\Pr(True) &= \Pr(A) + \Pr(\neg A) - \Pr(False) \\
1 &= \Pr(A) + \Pr(\neg A) - 0 \\
\Pr(\neg A) &= 1 - \Pr(A)
\end{aligned}$$

Discrete Random Variables

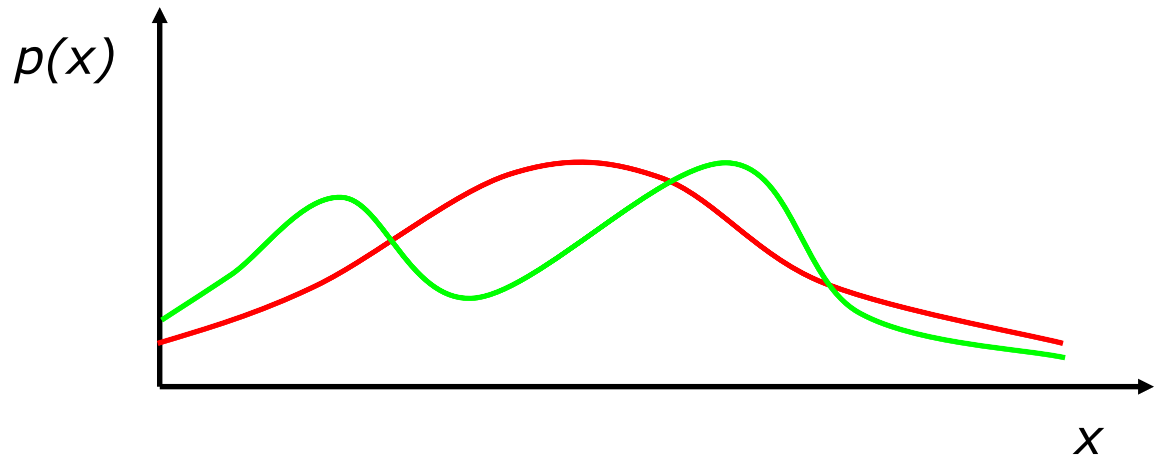
- X denotes a random variable.
- X can take on a countable number of values in $\{x_1, x_2, \ldots, x_n\}$.
- $P(X=x_i)$, or $P(x_i)$, is the probability that the random variable X takes on value x_i.
- $P(\ )$ is called probability mass function.
- E.g. $P(Room) = \langle 0.7, 0.2, 0.08, 0.02 \rangle$

Continuous Random Variables

- *X* takes on values in the continuum.
- *p(X=x)*, or *p(x)*, is a probability density function.

$$\Pr(x \in (a,b)) = \int_a^b p(x)\,dx$$

- E.g.



Joint and Conditional Probability

- $P(X{=}x \text{ and } Y{=}y) = P(x,y)$

- If X and Y are independent then

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

- $P(x \mid y)$ is the probability of x given y

$$P(x \mid y) = P(x,y) \,/\, P(y)$$

$$P(x,y) = P(x \mid y)\,P(y)$$

- If X and Y are independent then

$$P(x \mid y) = P(x)$$

Law of Total Probability, Marginals

Discrete case

$$\sum_x P(x) = 1$$

$$P(x) = \sum_y P(x, y)$$

$$P(x) = \sum_y P(x \mid y) P(y)$$

Continuous case

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

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

$$p(x) = \int p(x \mid y) p(y)\, dy$$

Bayes Formula

$$P(x,y) = P(x \mid y)P(y) = P(y \mid x)P(x)$$

$$\Rightarrow$$

$$\boxed{P(x \mid y) = \frac{P(y \mid x)\ P(x)}{P(y)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}}$$

Normalization

$$P(x \mid y) = \frac{P(y \mid x)\ P(x)}{P(y)} = \eta\ P(y \mid x)\, P(x)$$

$$\eta = P(y)^{-1} = \frac{1}{\sum_x P(y \mid x) P(x)}$$

Algorithm:

$$\forall x : \mathrm{aux}_{x|y} = P(y \mid x)\ P(x)$$

$$\eta = \frac{1}{\sum_x \mathrm{aux}_{x|y}}$$

$$\forall x : P(x \mid y) = \eta\ \mathrm{aux}_{x|y}$$

Conditioning

- Law of total probability:

$$P(x) = \int P(x, z) dz$$

$$P(x) = \int P(x \mid z) P(z) dz$$

$$P(x \mid y) = \int P(x \mid y, z) P(z \mid y) dz$$

Bayes Rule with Background Knowledge

$$P(x \mid y, z) = \frac{P(y \mid x, z)\; P(x \mid z)}{P(y \mid z)}$$

Conditional Independence

$$P(x, y \mid z) = P(x \mid z) P(y \mid z)$$

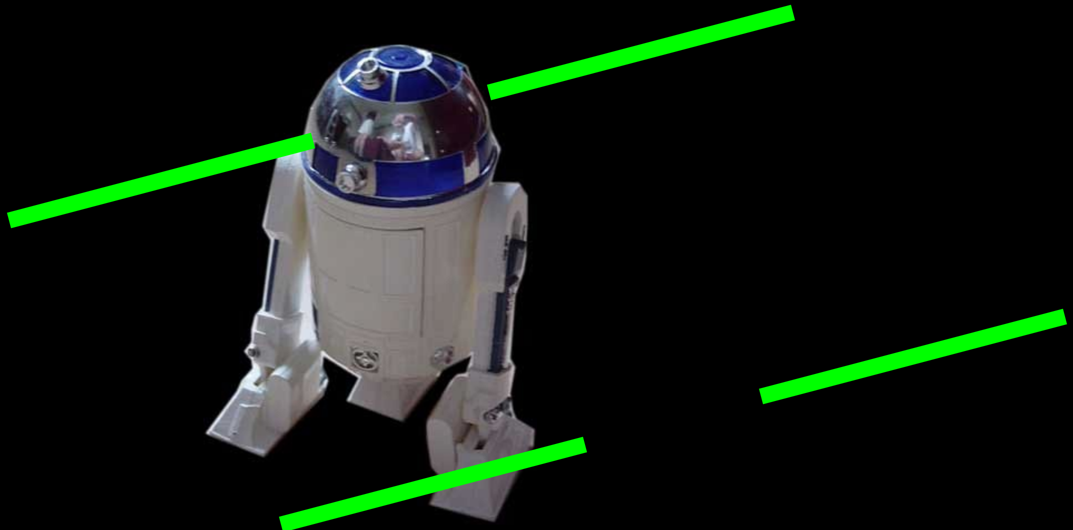
equivalent to

$$P(x \mid z) = P(x \mid z, y)$$

and

$$P(y \mid z) = P(y \mid z, x)$$

Robot Environment Interaction



$p(x_0)$

$p(x_0|z_0)$

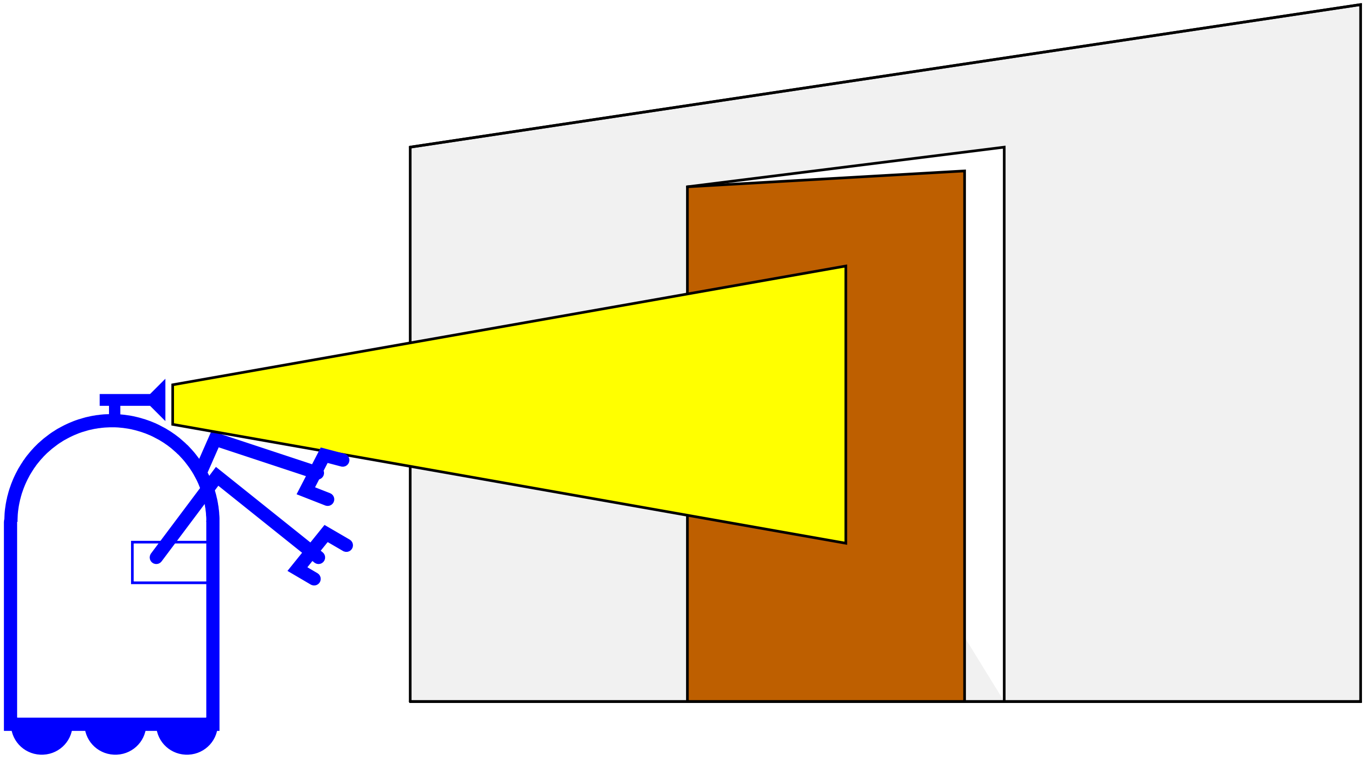
$p(x_1|z_0,u_1)$

$p(x_1|z_0,u_1,z_1)$

$p(x_2|z_0,u_1,z_1,u_2)$

Simple Example of State Estimation

- Suppose a robot obtains measurement z
- What is $P(open|z)$?



Causal vs. Diagnostic Reasoning

- *P(open|z)* is diagnostic.
- *P(z|open)* is causal.
- Often causal knowledge is easier to obtain.
- Bayes rule allows us **count frequencies!** knowledge:

$$P(open \mid z) = \frac{P(z \mid open) P(open)}{P(z)}$$

Example

- $P(z|open) = 0.6$ $\qquad$ $P(z|\neg open) = 0.3$
- $P(open) = P(\neg open) = 0.5$

$$P(open \mid z) = \frac{P(z \mid open)P(open)}{P(z \mid open)\,p(open) + P(z \mid \neg open)\,p(\neg open)}$$

$$P(open \mid z) = \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{2}{3} = 0.67$$

- z raises the probability that the door is open.

Combining Evidence

- Suppose our robot obtains another observation *z2*.
- How can we integrate this new information?
- More generally, how can we estimate $P(x|\ z1...zn\ )$?

Recursive Bayesian Updating

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x, z_1, \ldots, z_{n-1})\, P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

Markov assumption: z_n is independent of $z_1, \ldots, z_{n-1}$ if we know x.

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x)\, P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

$$= \eta\, P(z_n \mid x)\, P(x \mid z_1, \ldots, z_{n-1})$$

$$= \eta_{1\ldots n} \prod_{i=1\ldots n} P(z_i \mid x)\, P(x)$$

Example: Second Measurement

- $P(z_2|open) = 0.5$ $\qquad P(z_2|\neg open) = 0.6$
- $P(open|z_1)=2/3$

$$P(open \mid z_2, z_1) = \frac{P(z_2 \mid open)\, P(open \mid z_1)}{P(z_2 \mid open)\, P(open \mid z_1) + P(z_2 \mid \neg open)\, P(\neg open \mid z_1)}$$

$$= \frac{\frac{1}{2} \cdot \frac{2}{3}}{\frac{1}{2} \cdot \frac{2}{3} + \frac{3}{5} \cdot \frac{1}{3}} = \frac{5}{8} = 0.625$$

- z_2 lowers the probability that the door is open.

Actions

- Often the world is **dynamic** since
  - **actions carried out by the robot**,
  - **actions carried out by other agents**,
  - or just the **time** passing by changes the world.

- How can we **incorporate** such **actions**?

Typical Actions

- The robot **turns its wheels** to move
- The robot **uses its manipulator** to grasp an object
- Plants grow over **time**...

- Actions are **never carried out with absolute certainty**.
- In contrast to measurements, **actions generally increase the uncertainty**.

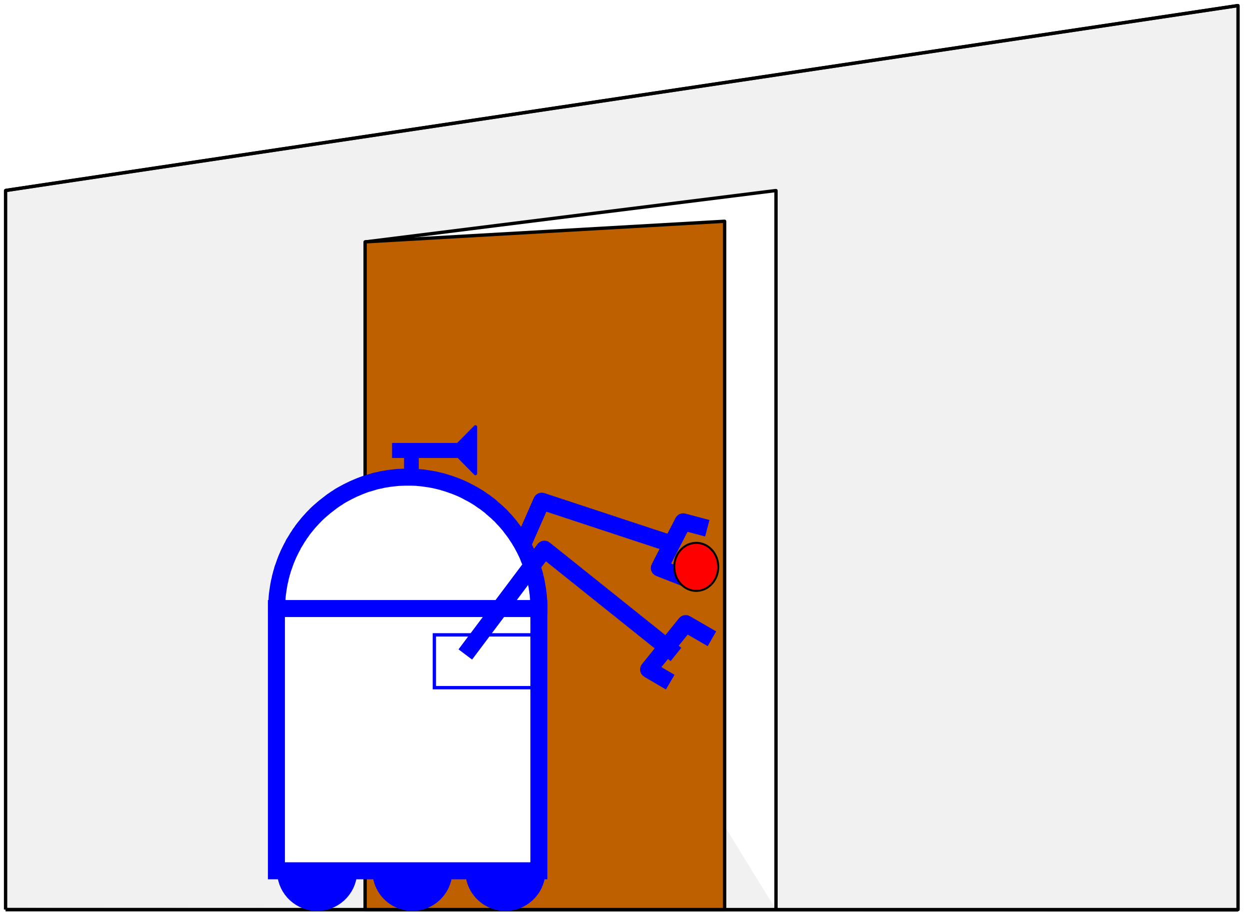
Modeling Actions

- To incorporate the outcome of an action *u* into the current "belief", we use the conditional pdf

$$P(x|u,x')$$

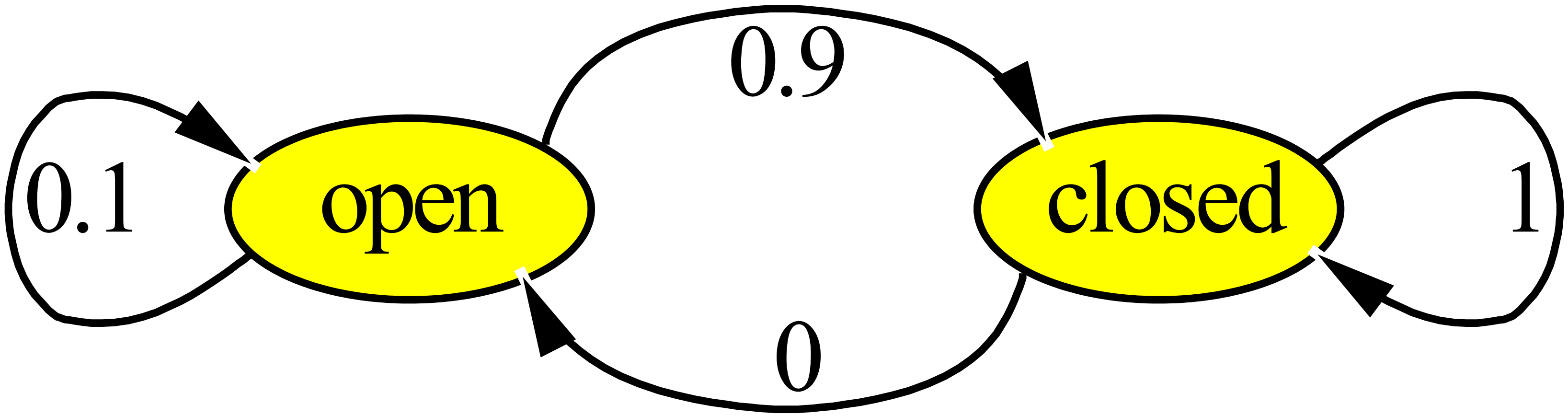
- This term specifies the pdf that **executing *u* changes the state from *x'* to *x***.

Example: Closing the door



State Transitions

$P(x|u,x')$ for u = "close door":



If the door is open, the action "close door" succeeds in 90% of all cases.

Integrating the Outcome of Actions

Continuous case:

$$P(x \mid u) = \int P(x \mid u, x')P(x')dx'$$

Discrete case:

$$P(x \mid u) = \sum P(x \mid u, x')P(x')$$

Example: The Resulting Belief

$$
\begin{aligned}
P(closed \mid u) &= \sum P(closed \mid u, x')P(x') \\
&= P(closed \mid u, open)P(open) \\
&\quad + P(closed \mid u, closed)P(closed) \\
&= \frac{9}{10} * \frac{5}{8} + \frac{1}{1} * \frac{3}{8} = \frac{15}{16}
\end{aligned}
$$

$$
\begin{aligned}
P(open \mid u) &= \sum P(open \mid u, x')P(x') \\
&= P(open \mid u, open)P(open) \\
&\quad + P(open \mid u, closed)P(closed) \\
&= \frac{1}{10} * \frac{5}{8} + \frac{0}{1} * \frac{3}{8} = \frac{1}{16} \\
&= 1 - P(closed \mid u)
\end{aligned}
$$

Bayes Filters: Framework

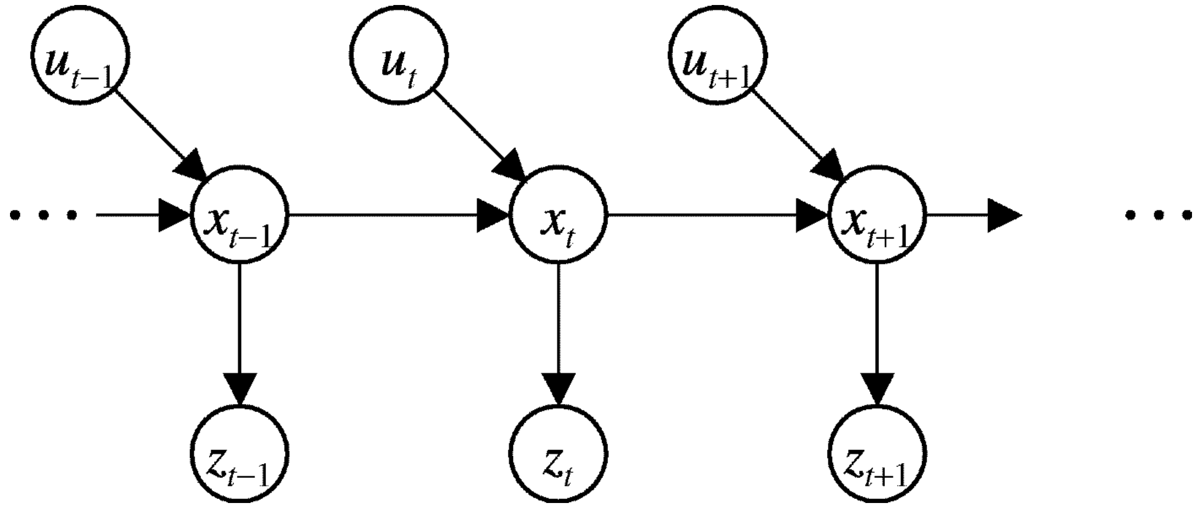
- **Given:**
  - Stream of observations *z* and action data *u:*

$$d_t = \{u_1, z_1 \ldots, u_t, z_t\}$$

  - Sensor model *P(z|x).*
  - Action model *P(x|u,x')*.
  - Prior probability of the system state *P(x).*
- **Wanted:**
  - Estimate of the state *X* of a dynamical system.
  - The posterior of the state is also called **Belief**:

$$Bel(x_t) = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

Markov Assumption



$$p(z_t \mid x_{0:t}, z_{1:t}, u_{1:t}) = p(z_t \mid x_t)$$

$$p(x_t \mid x_{1:t-1}, z_{1:t}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$$

Underlying Assumptions

- Static world
- Independent noise
- Perfect model, no approximation errors

Bayes Filters

z = observation
u = action
x = state

$$\boxed{Bel(x_t)} = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

Bayes $$= \eta \; P(z_t \mid x_t, u_1, z_1, \ldots, u_t) \; P(x_t \mid u_1, z_1, \ldots, u_t)$$

Markov $$= \eta \; P(z_t \mid x_t) \; P(x_t \mid u_1, z_1, \ldots, u_t)$$

Total prob. $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_1, z_1, \ldots, u_t, x_{t-1}) \; P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \; dx_{t-1}$$

Markov $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \; dx_{t-1}$$

Markov $$= \eta P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \, P(x_{t-1} \mid u_1, z_1, \ldots, z_{t-1}) \; dx_{t-1}$$

$$\boxed{= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}}$$

Bayes Filters are Familiar!

$$Bel(x_t) = \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}$$

- Kalman filters
- Particle filters
- Hidden Markov models
- Dynamic Bayesian networks
- Partially Observable Markov Decision Processes (POMDPs)

Bayes Filter Summary

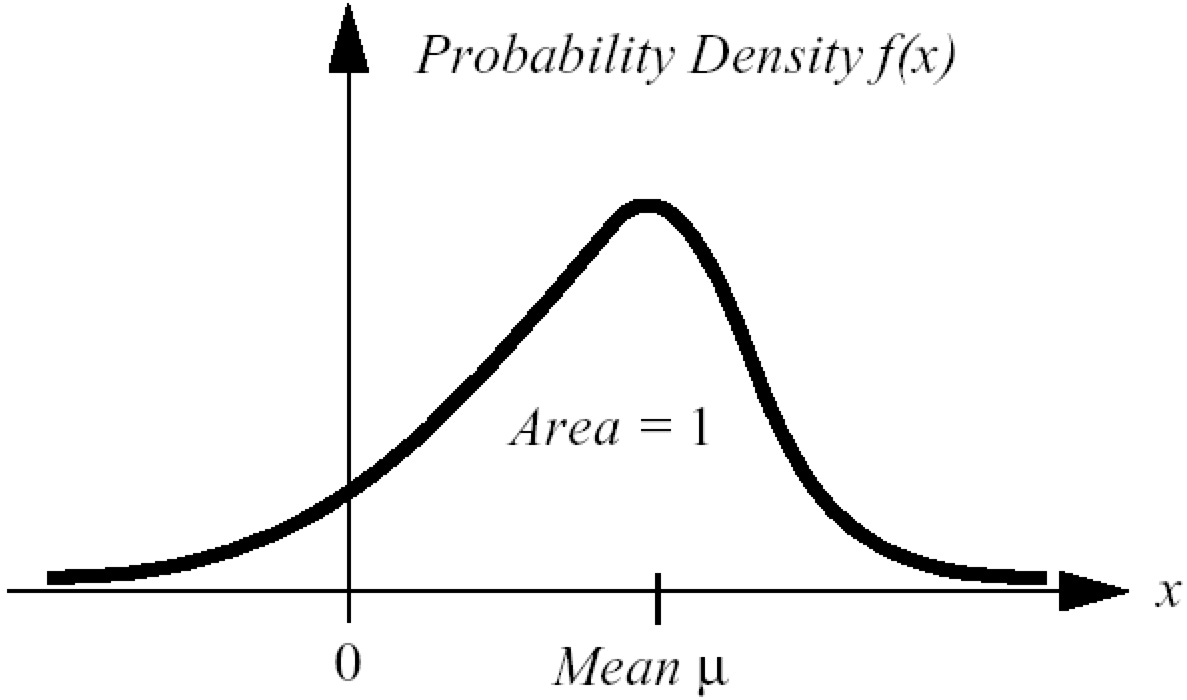
- Bayes rule allows us to compute probabilities that are hard to assess otherwise.
- Under the Markov assumption, recursive Bayesian updating can be used to efficiently combine evidence.
- Bayes filters are a probabilistic tool for estimating the state of dynamic systems.

Today's Goals

- Uncertainty & Probabilistic Robotics
- Uncertainty Presentation & Propagation
- Introduction to Kalman Filter

Uncertainty Representation

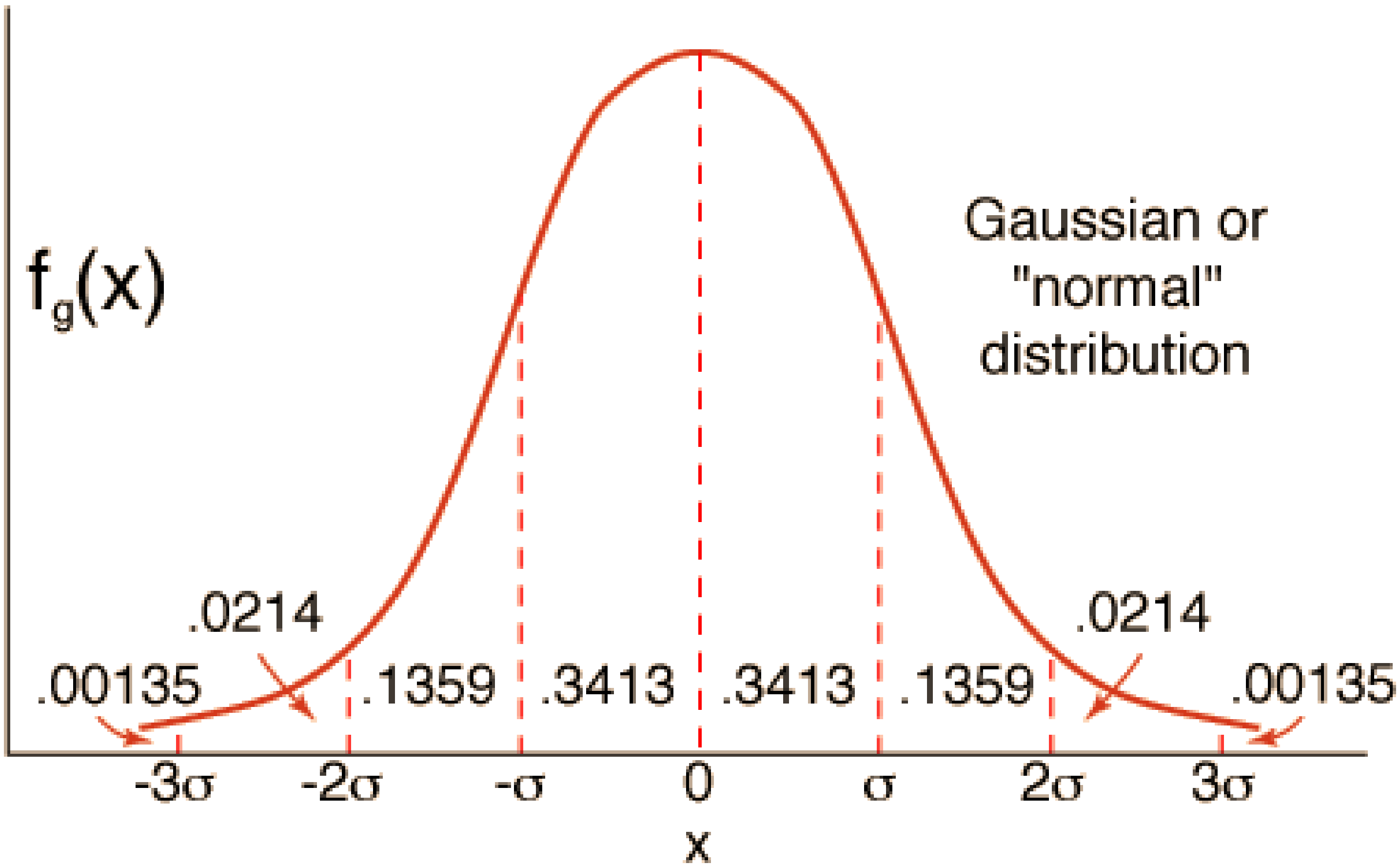
- Statistical representation
- Independence of random variables



Gaussian Distribution

$\mu = 0$ and $\sigma = 1$

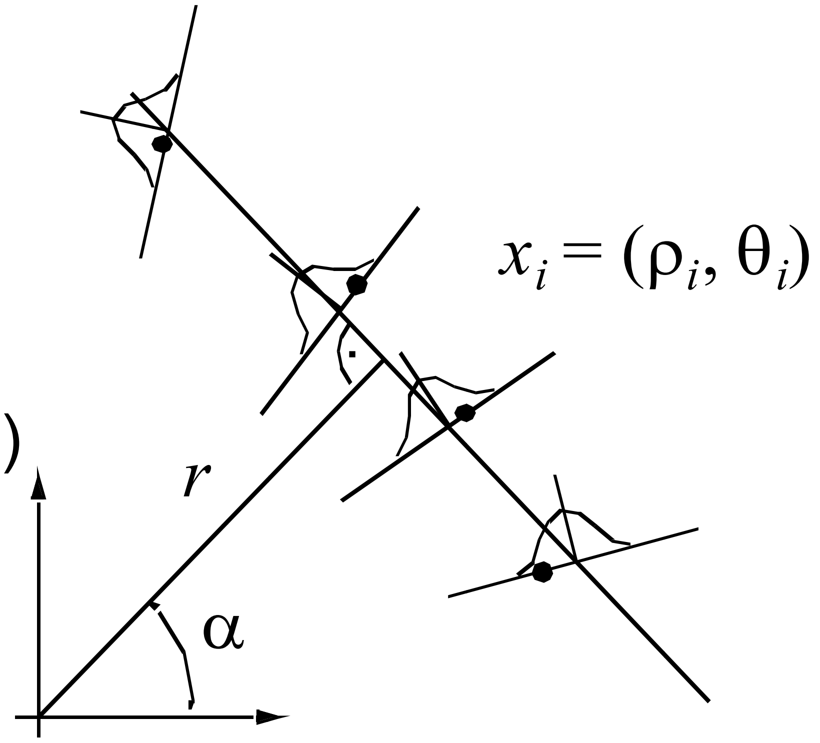
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



The Error Propagation Law: Motivation

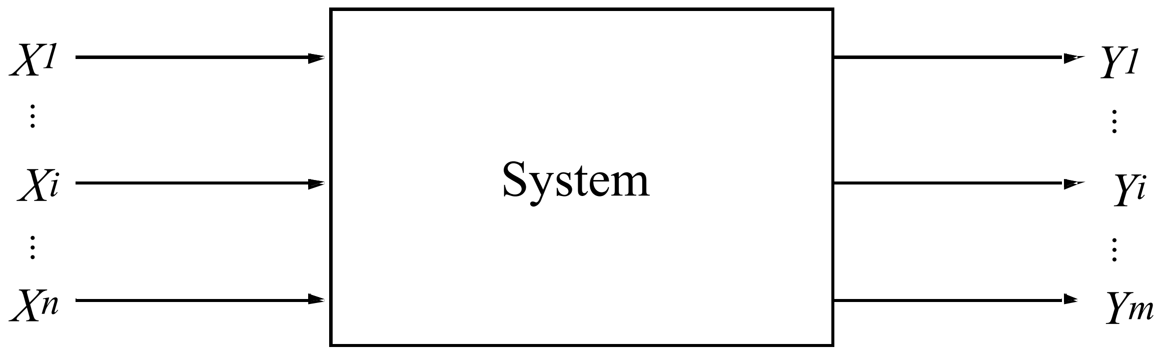
- Imagine extracting a line based on point measurements with uncertainties.

- The model parameters ρ_i (length of the perpendicular) and θ_i (its angle to the abscissa) describe a line uniquely.



- The question:
  - What is the uncertainty of the extracted line knowing the uncertainties of the measurement points that contribute to it ?

The Error Propagation Law

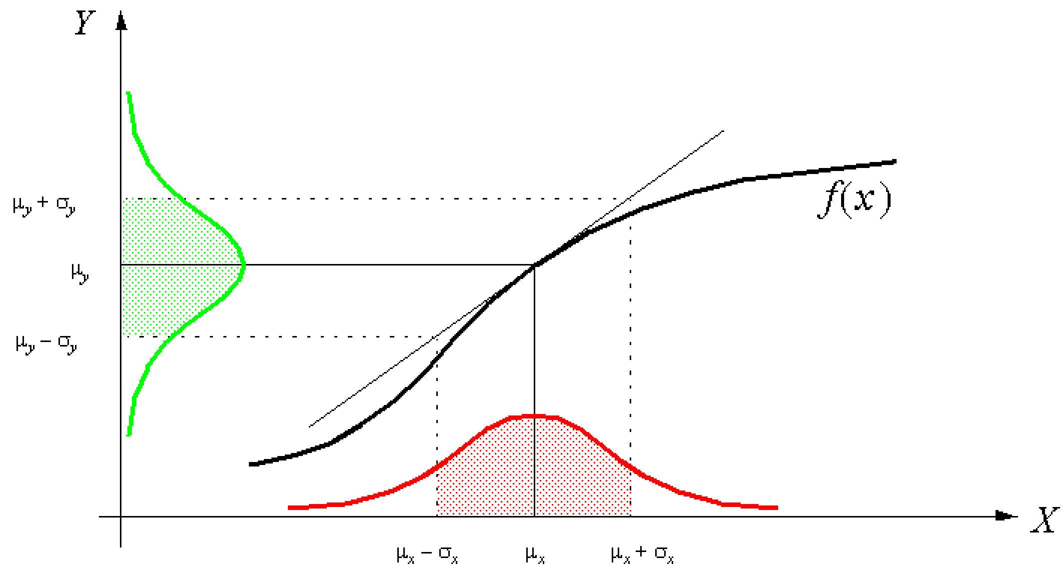


- Error propagation in a multiple-input multi-output system with *n* inputs and *m* outputs.

$$Y_j = f_j\,(X_1 \ldots X_n)$$

The Error Propagation Law

- One-dimensional case of nonlinear error propagatic problem
- It can be shown, that the output covariance matrix C_Y is given by the error propagation law:



$$C_Y = F_X C_X F_X^T$$

- where
  - C_X: covariance matrix representing the input uncertainties
  - C_Y: covariance matrix representing the propagated uncertainties for the outputs.
  - F_X: is the **Jacobian** matrix defined as:

$$F_X = \nabla f = \left[\nabla_X \cdot f(X)^T\right]^T = \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial X_1} & \cdots & \frac{\partial}{\partial X_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial X_1} & \cdots & \frac{\partial f_1}{\partial X_n} \\ \vdots & \cdots & \vdots \\ \frac{\partial f_m}{\partial X_1} & \cdots & \frac{\partial f_m}{\partial X_n} \end{bmatrix}$$

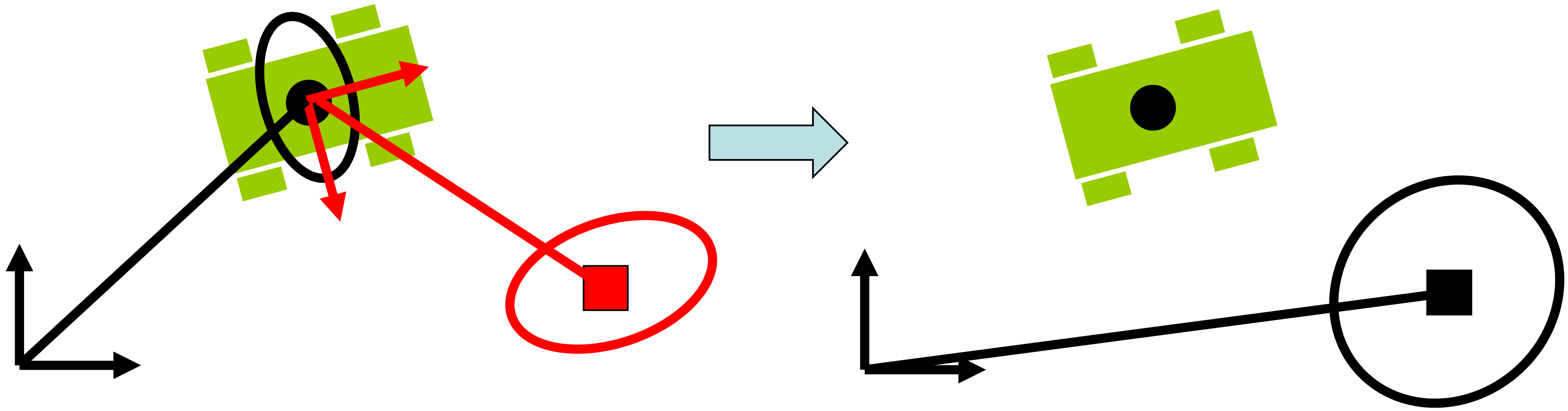
  - which is the transposed of the gradient of f(X).

Uncertain Spatial Relationships

- In robotics, manipulating uncertain spatial relationships is fundamental.
  - Compounding relationship
  - Inverse relationship
  - Tail-to-tail relationship

Chapter 2.1 of Wang's PhD thesis

Compounding (head-to-tail)



$$\mathbf{x}_{ij} = \begin{bmatrix} x_{ij} \\ y_{ij} \\ \theta_{ij} \end{bmatrix} \quad , \quad \mathbf{x}_{jk} = \begin{bmatrix} x_{jk} \\ y_{jk} \\ \theta_{jk} \end{bmatrix}$$

$$\mathbf{x}_{ik} \triangleq \oplus(\mathbf{x}_{ij}, \mathbf{x}_{jk}) = \begin{bmatrix} x_{jk}\cos\theta_{ij} - y_{jk}\sin\theta_{ij} + x_{ij} \\ x_{jk}\sin\theta_{ij} + y_{jk}\cos\theta_{ij} + y_{ij} \\ \theta_{ij} + \theta_{jk} \end{bmatrix}$$

Compounding

- The first-order estimate of the mean

$$\mu_{\mathbf{x}_{ik}} \approx \oplus(\mu_{\mathbf{x}_{ij}}, \mu_{\mathbf{x}_{jk}})$$

- The first-order estimate of the covariance

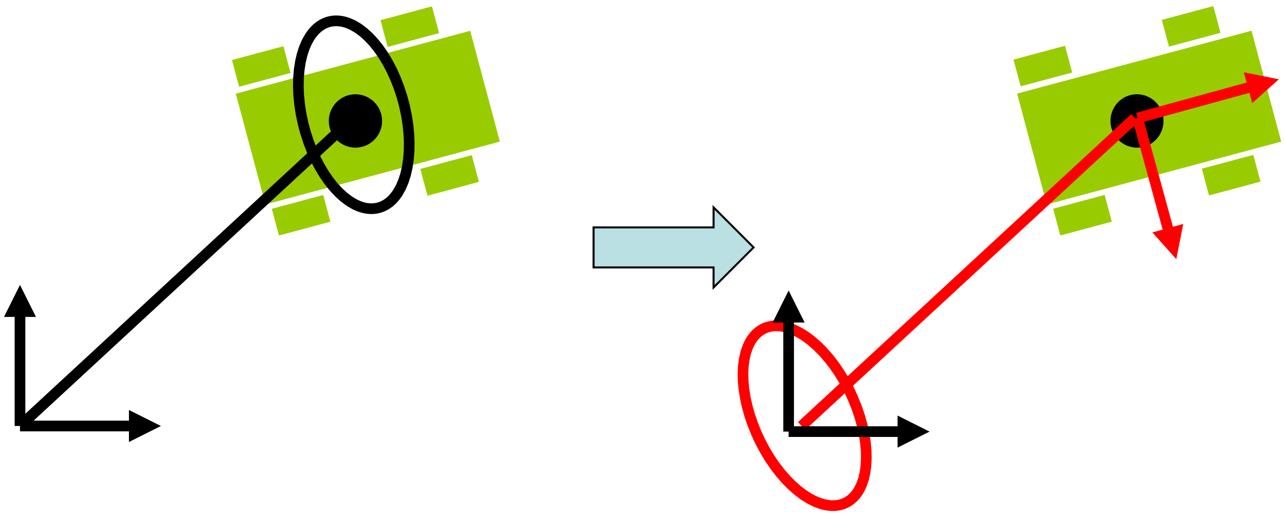
$$\Sigma_{\mathbf{x}_{ik}} \approx \nabla_{\oplus} \begin{bmatrix} \Sigma_{\mathbf{x}_{ij}} & \Sigma_{\mathbf{x}_{ij}\mathbf{x}_{jk}} \\ \Sigma_{\mathbf{x}_{jk}\mathbf{x}_{ij}} & \Sigma_{\mathbf{x}_{jk}} \end{bmatrix} \nabla_{\oplus}^{T}$$

where the Jacobian of the compounding operator

$$\nabla_{\oplus} \triangleq \frac{\partial \oplus (\mathbf{x}_{ij}, \mathbf{x}_{jk})}{\partial(\mathbf{x}_{ij}, \mathbf{x}_{jk})} = \begin{bmatrix} 1 & 0 & -(y_{ik} - y_{ij}) & \cos\theta_{ij} & -\sin\theta_{ij} & 0 \\ 0 & 1 & (x_{ik} - x_{ij}) & \sin\theta_{ij} & \cos\theta_{ij} & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$\Sigma_{\mathbf{x}_{ik}} \approx \nabla_{1\oplus}\Sigma_{\mathbf{x}_{ik}}\nabla_{1\oplus}^{T} + \nabla_{2\oplus}\Sigma_{\mathbf{x}_{jk}}\nabla_{2\oplus}^{T}$$

Inverse



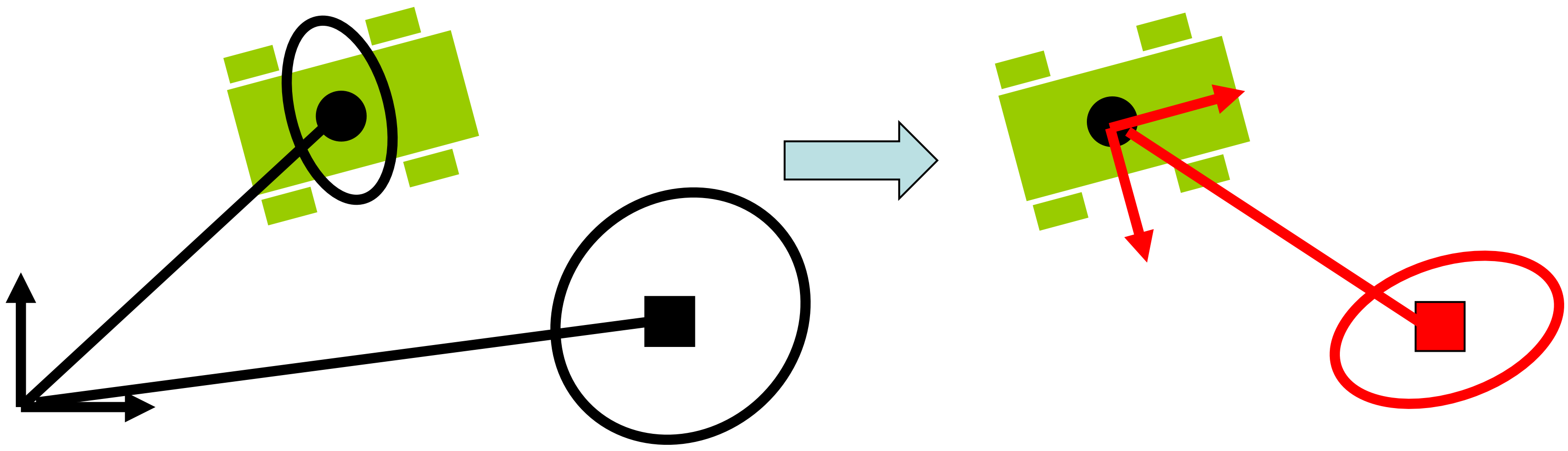
$$\mathbf{x}_{ji} \triangleq \ominus(\mathbf{x}_{ij}) = \begin{bmatrix} -x_{ij}\cos\theta_{ij} - y_{ij}\sin\theta_{ij} \\ x_{ij}\sin\theta_{ij} - y_{ij}\cos\theta_{ij} \\ -\theta_{ij} \end{bmatrix}$$

$$\mu_{\mathbf{x}_{ji}} \approx \ominus(\mu_{\mathbf{x}_{ij}})$$

$$\Sigma_{\mathbf{x}_{ji}} \approx \nabla_{\ominus}\Sigma_{\mathbf{x}_{ij}}\nabla_{\ominus}^{T}$$

$$\nabla_{\ominus} \triangleq \frac{\partial \mathbf{x}_{ji}}{\partial \mathbf{x}_{ij}} = \begin{bmatrix} -\cos\theta_{ij} & -\sin\theta_{ij} & y_{ji} \\ \sin\theta_{ij} & -\cos\theta_{ij} & -x_{ji} \\ 0 & 0 & -1 \end{bmatrix}$$

Tail-to-tail



$$\mathbf{x}_{jk} \triangleq \oplus(\ominus(\mathbf{x}_{ij}), \mathbf{x}_{ik}) = \oplus(\mathbf{x}_{ji}, \mathbf{x}_{ik})$$

$$\mu_{\mathbf{x}_{jk}} \approx \oplus(\ominus(\mu_{\mathbf{x}_{ij}}), \mu_{\mathbf{x}_{ik}})$$

$$\Sigma_{\mathbf{x}_{jk}} \approx \nabla_{\oplus} \begin{bmatrix} \Sigma_{\mathbf{x}_{ji}} & \Sigma_{\mathbf{x}_{ji}\mathbf{x}_{jk}} \\ \Sigma_{\mathbf{x}_{jk}\mathbf{x}_{ji}} & \Sigma_{\mathbf{x}_{jk}} \end{bmatrix} \nabla_{\oplus}^{T} \approx \nabla_{\oplus} \begin{bmatrix} \nabla_{\ominus}\Sigma_{\mathbf{x}_{ij}}\nabla_{\ominus}^{T} & \Sigma_{\mathbf{x}_{ij}\mathbf{x}_{jk}}\nabla_{\ominus}^{T} \\ \nabla_{\ominus}\Sigma_{\mathbf{x}_{jk}\mathbf{x}_{ij}} & \Sigma_{\mathbf{x}_{jk}} \end{bmatrix} \nabla_{\oplus}^{T}$$

Unknown or Nonlinear System Model

- Sampling

- Unscented Transform

Today's Goals

- Uncertainty & Probabilistic Robotics
- Uncertainty Presentation & Propagation
- Introduction to Kalman Filter

Reminder: Bayes Filter

- Prediction

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\, bel(x_{t-1})\, dx_{t-1}$$

- Correction

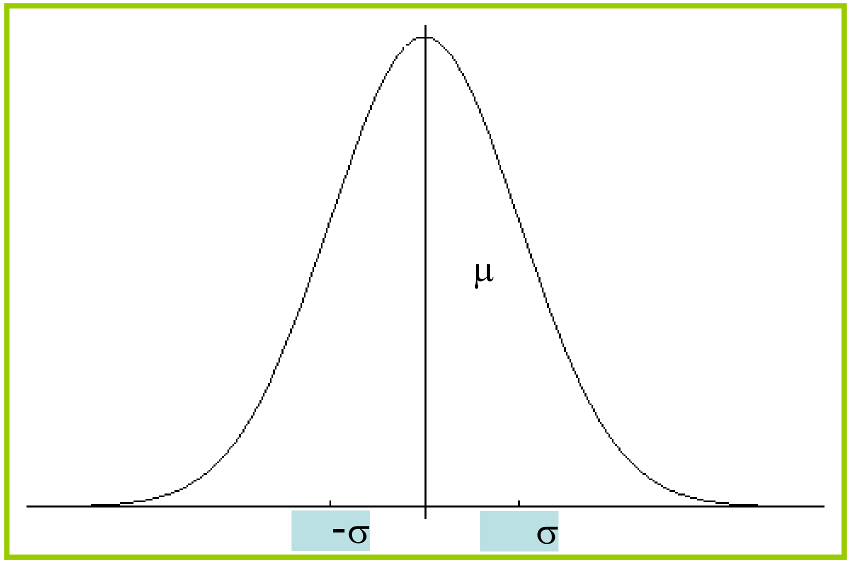
$$bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t)$$

Gaussians

$$p(x) \sim N(\mu, \sigma^2):$$

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

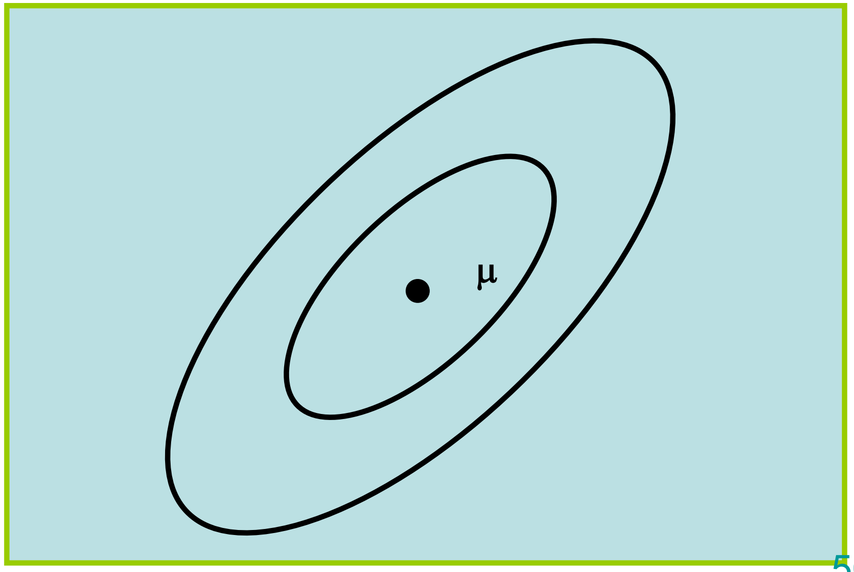
Univariate



$$p(\mathbf{x}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}):$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

Multivariate



Properties of Gaussians

$$\left.\begin{array}{l} X \sim N(\mu, \sigma^2) \\ Y = aX + b \end{array}\right\} \Rightarrow \quad Y \sim N(a\mu + b, a^2\sigma^2)$$

$$\left.\begin{array}{l} X_1 \sim N(\mu_1, \sigma_1^{\ 2}) \\ X_2 \sim N(\mu_2, \sigma_2^{\ 2}) \end{array}\right\} \Rightarrow p(X_1) \cdot p(X_2) \sim N\left(\frac{\sigma_2^{\ 2}}{\sigma_1^{\ 2} + \sigma_2^{\ 2}} \mu_1 + \frac{\sigma_1^{\ 2}}{\sigma_1^{\ 2} + \sigma_2^{\ 2}} \mu_2, \quad \frac{1}{\sigma_1^{\ -2} + \sigma_2^{\ -2}} \right)$$

Multivariate Gaussians

$$\left.\begin{array}{l} X \sim N(\mu, \Sigma) \\ Y = AX + B \end{array}\right\} \Rightarrow \quad Y \sim N(A\mu + B, A\Sigma A^T)$$

$$\left.\begin{array}{l} X_1 \sim N(\mu_1, \Sigma_1) \\ X_2 \sim N(\mu_2, \Sigma_2) \end{array}\right\} \Rightarrow p(X_1) \cdot p(X_2) \sim N\left(\frac{\Sigma_2}{\Sigma_1 + \Sigma_2} \mu_1 + \frac{\Sigma_1}{\Sigma_1 + \Sigma_2} \mu_2, \quad \frac{1}{\Sigma_1^{-1} + \Sigma_2^{-1}} \right)$$

- We stay in the "Gaussian world" as long as we start with Gaussians and perform only linear transformations.

Discrete Kalman Filter

Estimates the state x of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

with a measurement

$$z_t = C_t x_t + \delta_t$$

Components of a Kalman Filter

A_t Matrix (nxn) that describes how the state evolves from t to $t\text{-}1$ without controls or noise.

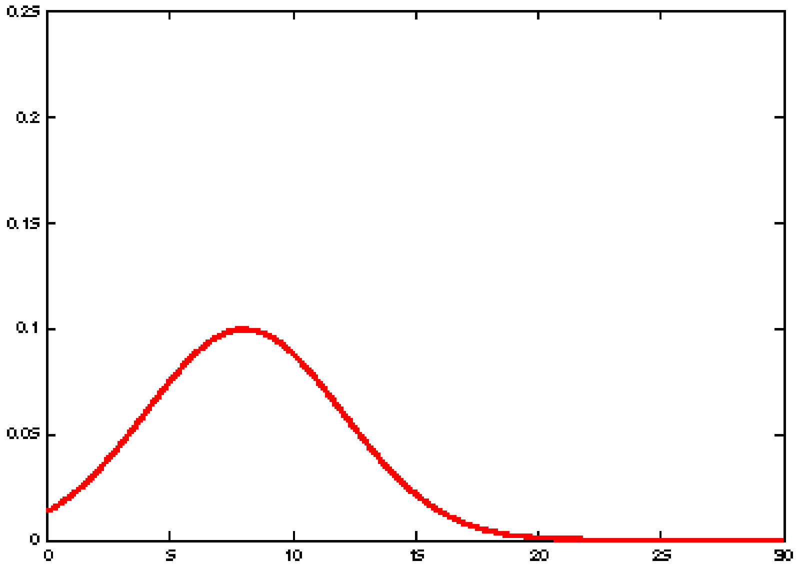
B_t Matrix (nxl) that describes how the control u_t changes the state from t to $t\text{-}1$.

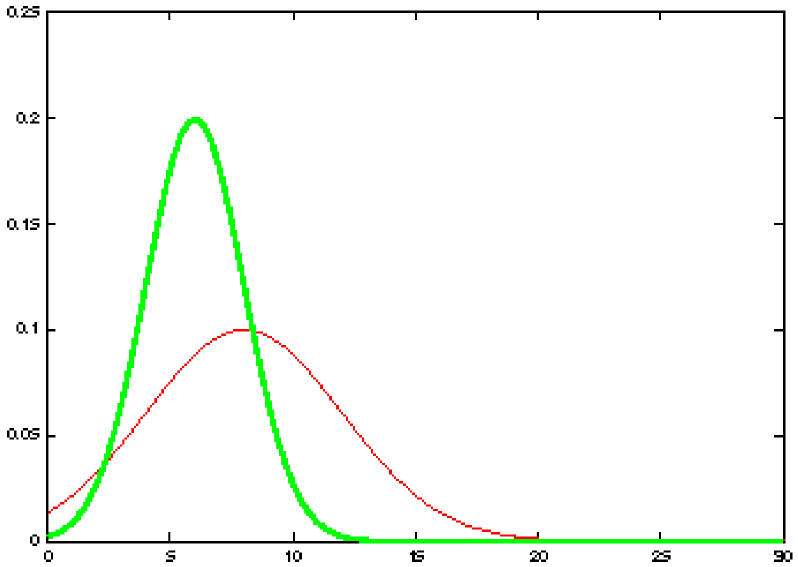
C_t Matrix (kxn) that describes how to map the state x_t to an observation z_t.

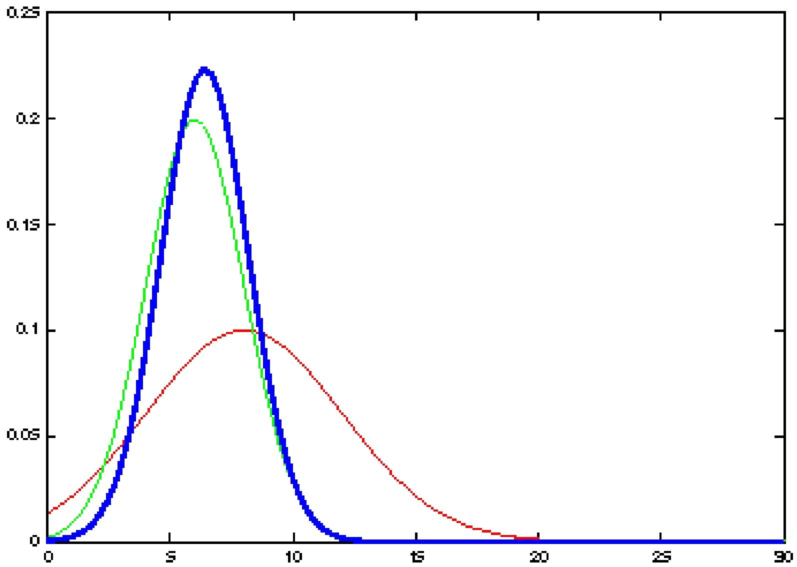
ε_t

δ_t Random variables representing the process and measurement noise that are assumed to be independent and normally distributed with covariance R_t and Q_t respectively.

Kalman Filter Updates in 1D



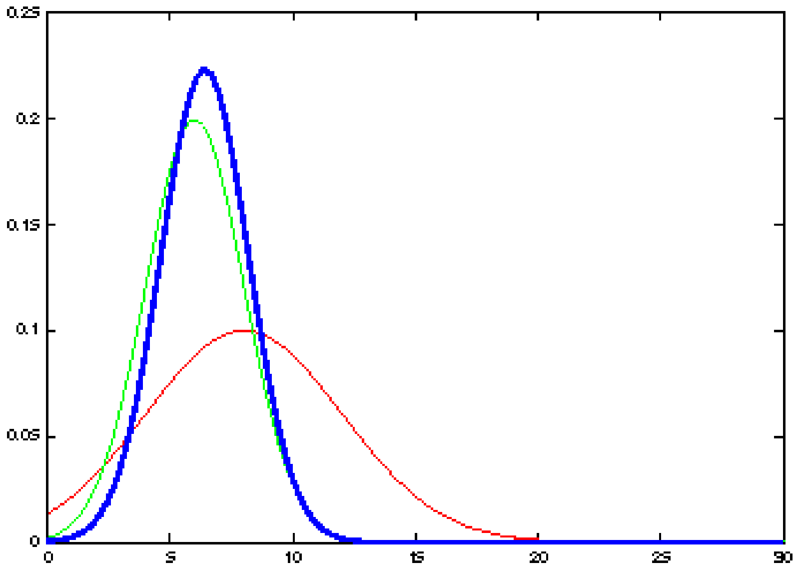




Kalman Filter Updates in 1D

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\bar{\sigma}_t^2 \end{cases} \quad \text{with} \quad K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \bar{\sigma}_{obs,t}^2}$$

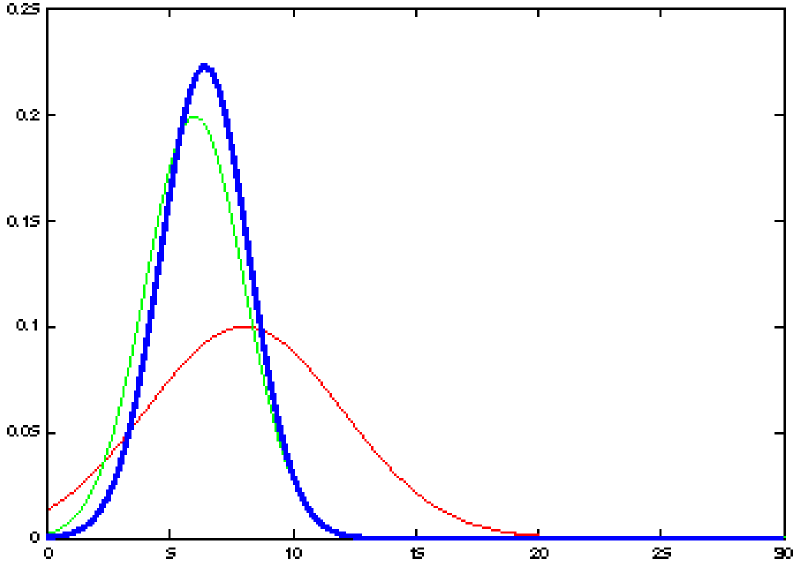
$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\bar{\Sigma}_t \end{cases} \quad \text{with} \quad K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

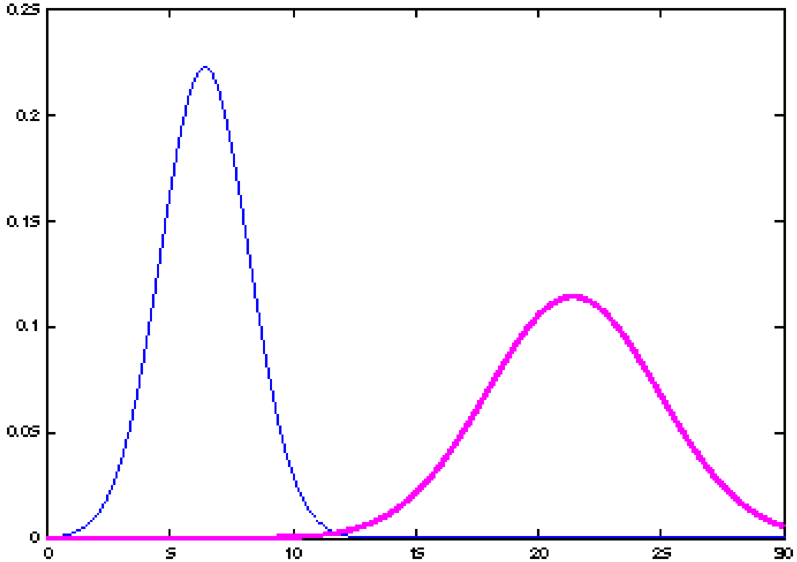


Kalman Filter Updates in 1D

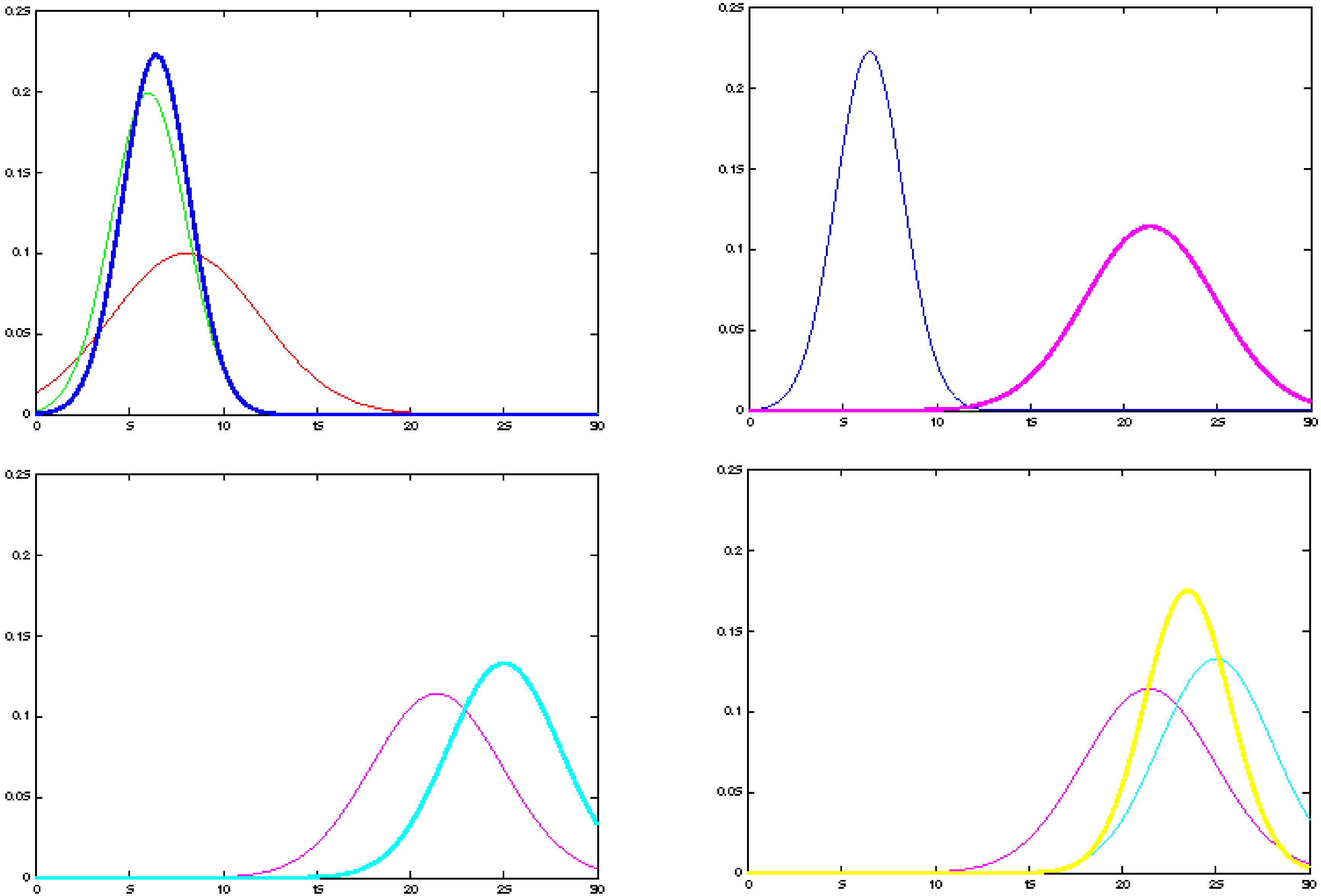
$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$





Kalman Filter Updates



Linear Gaussian Systems: Initialization

- Initial belief is normally distributed:

$$\boxed{bel(x_0) = N\left(x_0; \mu_0, \Sigma_0\right)}$$

Linear Gaussian Systems: Dynamics

- Dynamics are linear function of state and control plus additive noise:

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

$$p(x_t \mid u_t, x_{t-1}) = N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right)$$

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) \qquad bel(x_{t-1})\, dx_{t-1}$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$\sim N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right) \quad \sim N\left(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}\right)$$

Linear Gaussian Systems: Dynamics

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) \qquad bel(x_{t-1})\, dx_{t-1}$$

$$\Downarrow \qquad\qquad\qquad \Downarrow$$

$$\sim N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right) \quad \sim N\left(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}\right)$$

$$\Downarrow$$

$$\overline{bel}(x_t) = \eta \int \exp\left\{-\frac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1} (x_t - A_t x_{t-1} - B_t u_t)\right\}$$

$$\exp\left\{-\frac{1}{2}(x_{t-1} - \mu_{t-1})^T \Sigma_{t-1}^{-1} (x_{t-1} - \mu_{t-1})\right\} dx_{t-1}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$

Linear Gaussian Systems: Observations

- Observations are linear function of state plus additive noise:

$$z_t = C_t x_t + \delta_t$$

$$p(z_t \mid x_t) = N\left(z_t; C_t x_t, Q_t\right)$$

$$
\begin{array}{cccc}
bel(x_t) = & \eta & p(z_t \mid x_t) & \overline{bel}(x_t) \\
 & & \Downarrow & \Downarrow \\
 & & \sim N\left(z_t; C_t x_t, Q_t\right) & \sim N\left(x_t; \overline{\mu}_t, \overline{\Sigma}_t\right)
\end{array}
$$

Linear Gaussian Systems: Observations

$$bel(x_t) = \quad \eta \quad p(z_t \mid x_t) \qquad\qquad \overline{bel}(x_t)$$

$$\Downarrow \qquad\qquad\qquad \Downarrow$$

$$\sim N\left(z_t; C_t x_t, Q_t\right) \qquad \sim N\left(x_t; \bar{\mu}_t, \bar{\Sigma}_t\right)$$

$$\Downarrow$$

$$bel(x_t) = \eta \exp\left\{-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1}(z_t - C_t x_t)\right\} \exp\left\{-\frac{1}{2}(x_t - \bar{\mu}_t)^T \bar{\Sigma}_t^{-1}(x_t - \bar{\mu}_t)\right\}$$

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\bar{\Sigma}_t \end{cases} \qquad \text{with} \quad K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

Kalman Filter Algorithm

1. Algorithm **Kalman_filter**(μ_{t-1}, Σ_{t-1}, u_t, z_t):

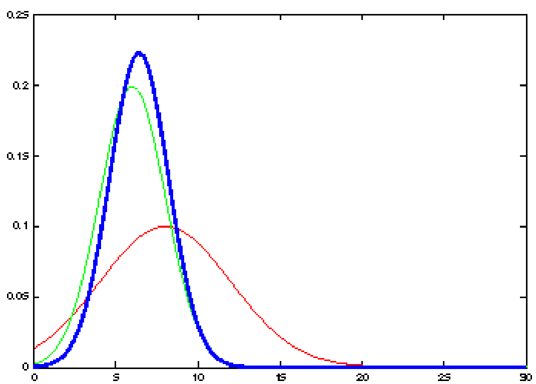
2. Prediction:
3. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
4. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

5. Correction:
6. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
7. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
8. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

9. Return μ_t, Σ_t

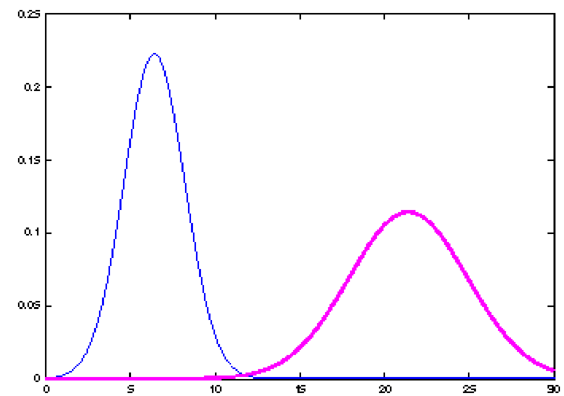
The Prediction-Correction-Cycle

Prediction

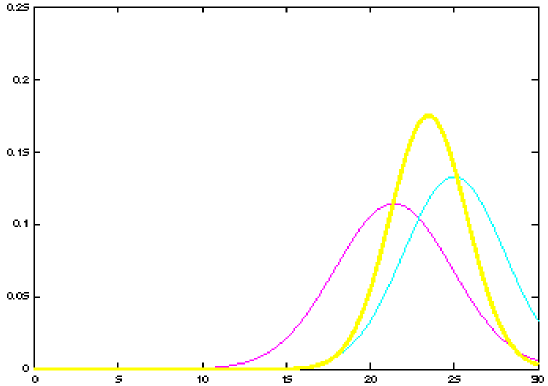


$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$

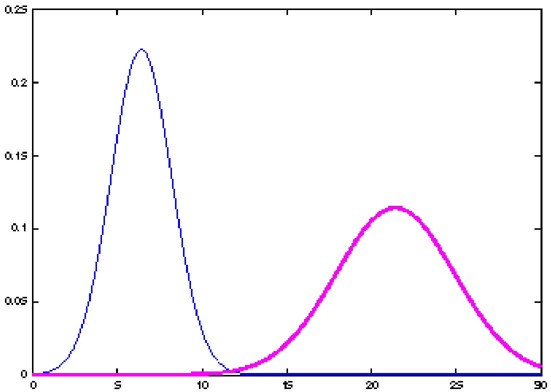


The Prediction-Correction-Cycle



$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\bar{\sigma}_t^2 \end{cases}, K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \bar{\sigma}_{obs,t}^2}$$

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\bar{\Sigma}_t \end{cases}, K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$



Correction

The Prediction-Correction-Cycle

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\bar{\sigma}_t^2 \end{cases}, K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \bar{\sigma}_{obs,t}^2}$$

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\bar{\Sigma}_t \end{cases}, K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$

Kalman Filter Summary

- Highly efficient: Polynomial in measurement dimensionality k and state dimensionality n:
  $O(k^{2.376} + n^2)$
- Optimal for linear Gaussian systems!
- Most robotics systems are nonlinear.

More about Filtering

- Gaussian Filters
  - The Extended Kalman Filter
  - The Unscented Kalman Filter
  - The Information Filter

- Nonparametric Filters
  - The Histogram Filter
  - Binary Bayes Filter with Static State
  - The Particle Filter

Questions?