Cognitive Robotics

02. Bayes Filters & Kalman Filters

AbdElMoniem Bayoumi, PhD

Fall 2022

Acknowledgment

- These slides have been created by Wolfram Burgard, Dieter Fox, Cyrill Stachniss and Maren Bennewitz

Previous Lecture

- Basic laws of probabilities

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

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

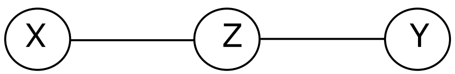
$$P(x, y) = P(x \mid y)\, P(y)$$

- Bayes rule

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

- Conditional independence

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



- Recursive Bayesian update to incorporate observations
  - Markov assumption: Measurements z_i independent when state x is known $P(x \mid z_1, \ldots, z_n) = \eta_{1\ldots n} \prod_{i=1\ldots n} P(z_i \mid x)\ P(x)$

Example again: Incorporating a Measurement

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

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

$$P(open \mid z) = \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{0.3}{0.3 + 0.15} = 0.67$$

- z increases the probability that the door is open.

Example again: Incorporating a Measurement

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

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

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

- z_2 lowers the probability that the door is open.

Actions

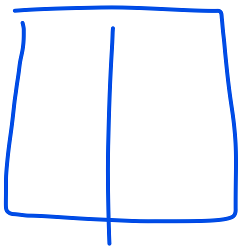
- Often the world is **dynamic** since
  - **actions carried out by the robot**,
  - **actions carried out by other agents**,
  - or just the **time** passing by

  change the world.

el tlata dol homa elly
by5lo el world dynamic

- 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**.

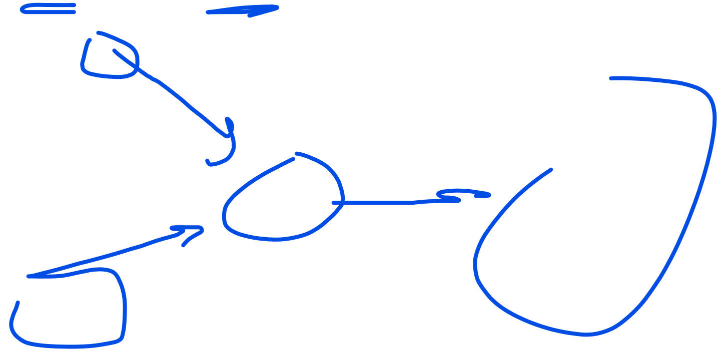
errors in sensors is usually less than the actoators.

Modeling Actions

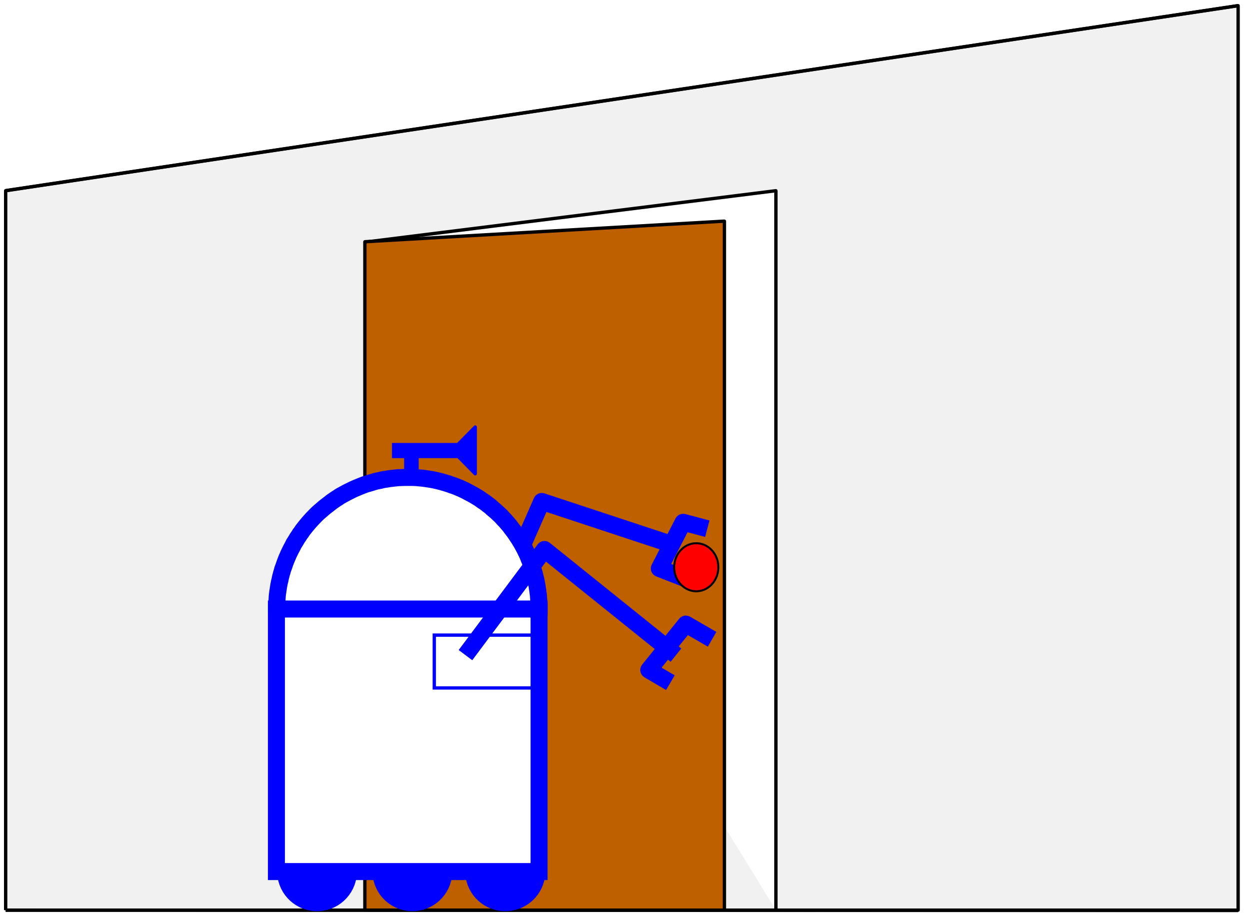
- To incorporate the outcome of an action *u* into the current "belief", we use the conditional probability density function

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

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

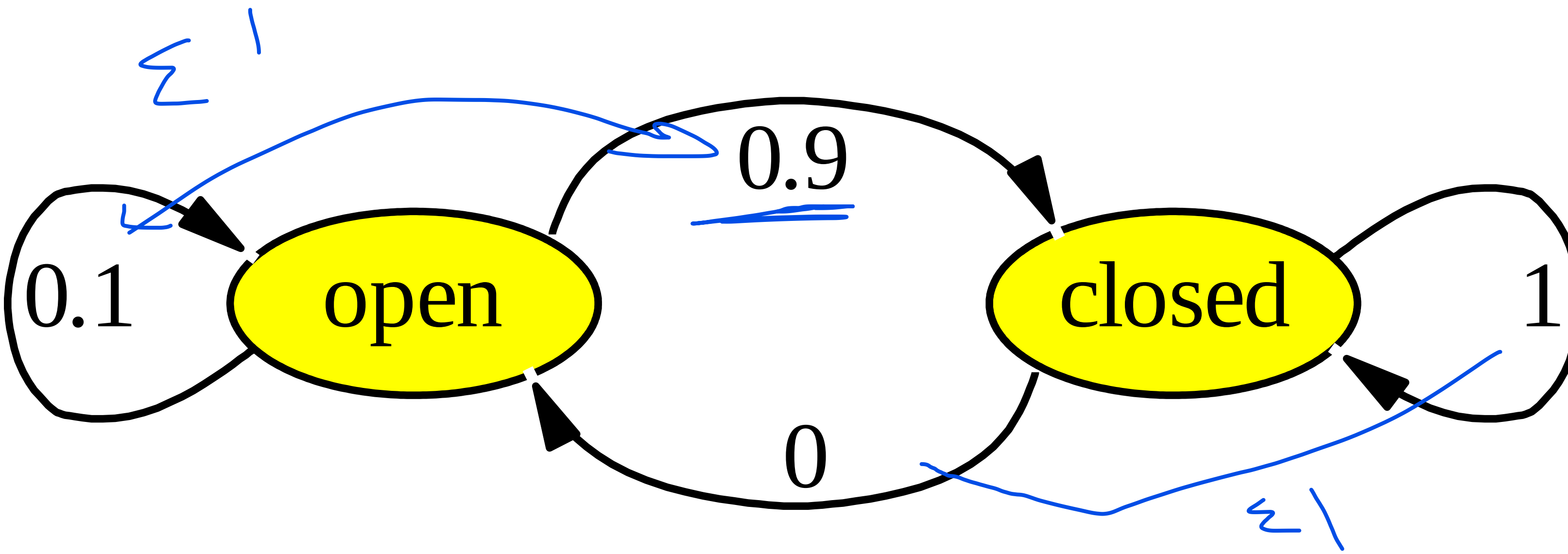


Example: Closing the door



State Transitions

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



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

Integrating the Outcome of Actions

Discrete case:

$$P(x \mid u) = \sum_{x'} P(x \mid u, x') P(x')$$

Continuous case:

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

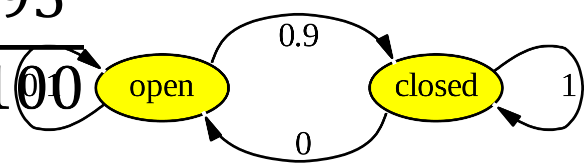
Example: The Resulting Belief (assuming no prior belief)

$$P(closed|u) = \sum_{x'} P(closed|u, x')P(x')$$

$$= P(closed|u, open)P(open)$$

$$+P(closed|u, closed)P(closed)$$

$$\rightleftarrows\rightleftarrows = \frac{9}{10} * \frac{1}{2} + \frac{1}{1} * \frac{1}{2} = \frac{95}{100}$$



$$P(open|u) = \sum_{x'} P(open|u, x')P(x')$$

$$= P(open|u, open)P(open)$$

$$+P(open|u, closed)P(closed)$$

$$\rightleftarrows\rightleftarrows = \frac{1}{10} * \frac{1}{2} + \frac{0}{1} * \frac{1}{2} = \frac{1}{20}$$

$$= 1 - P(closed|u)$$

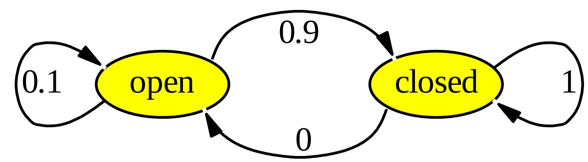
Example: The Resulting Belief (based on the belief after incorporating measurement)

$$P(closed|u, z_1, z_2) = \sum_{x'} P(closed|u, x')P(x'|z_1, z_2)$$

$$= P(closed|u, open)P(open|z_1, z_2)$$

$$+P(closed|u, closed)P(closed|z_1, z_2)$$

$$\rightleftarrows\rightleftarrows = \frac{9}{10} * \frac{5}{8} + \frac{1}{1} * \frac{3}{8} = \frac{15}{16}$$



$$P(open|u, z_1, z_2) = \sum_{x'} P(open|u, x')P(x'|z_1, z_2)$$

$$= P(open|u, open)P(open|z_1, z_2)$$

$$+P(open|u, closed)P(closed|z_1, z_2)$$

$$\rightleftarrows\rightleftarrows = \frac{1}{10} * \frac{5}{8} + \frac{0}{1} * \frac{3}{8} = \frac{1}{16}$$

$$= 1 - P(closed|u)$$

Bayes Filters: Framework

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

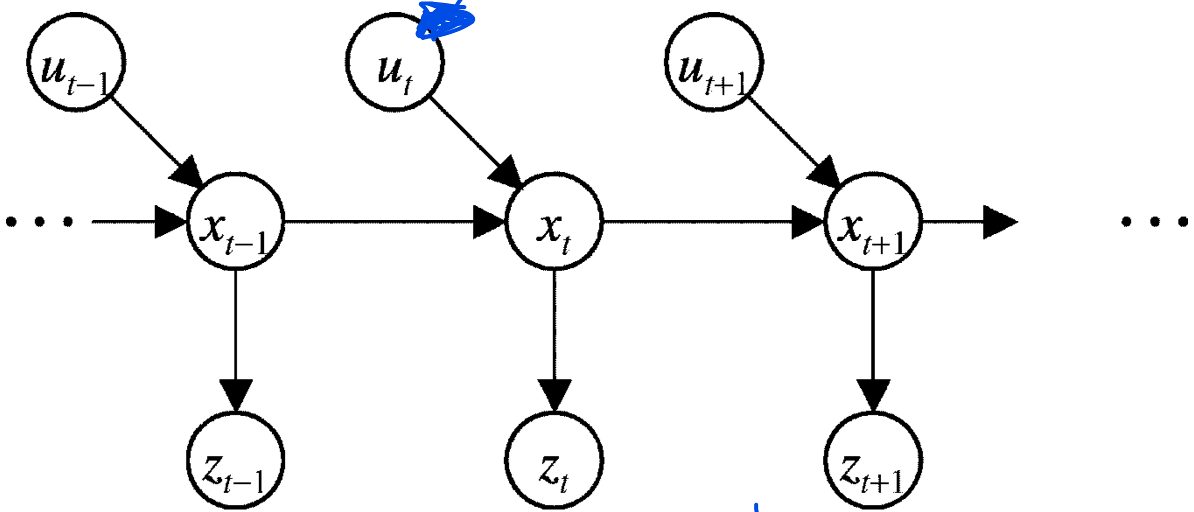
$$d_t = \{u_1, z_1 \ldots, u_t, z_t\}$$

  - **Sensor model** $P(z \mid x)$.
  - **Action model** $P(x \mid u, x')$.
  - **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

Ut hya el 5lt feh transition mn xt-1 to xt



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

each state depends only on the previous state.

Bayes Filters

z = observation
u = action
x = state

P(X|Y) = P(Y|X) P (X) / P(Y)

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

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

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

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

Xt-1 does not depend on ut, 34an ut da fl most2bl bta3 xt-1, fa 34an keda n2dr n2ol enha independent.

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

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

Bayes Filter Interpretation

- Prediction

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

- Correction

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

Bayes Filter Algorithm

1. Algorithm **Bayes_filter**(*Bel(x), d*):
2. $\eta=0$
3. If *d* is a **perceptual** data item *z* then
4. For all *x* do
5. $Bel'(x) = P(z \mid x) Bel(x)$
6. $\eta = \eta + Bel'(x)$
7. For all *x* do
8. $Bel'(x) = \eta^{-1} Bel'(x)$
9. Else if *d* is an **action** data item *u* then
10. For all *x* do
11. $Bel'(x) = \int P(x \mid u, x') \, Bel(x') \, dx'$
12. Return *Bel'(x)*

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

Bayes Filters come in many forms

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

- Kalman Filters
- Particle Filters
- Hidden Markov Models
- Dynamic Bayesian Networks
- Partially Observable Markov Decision Processes (POMDPs)

Bayes filters are a versatile tool for estimating the state of dynamic systems.

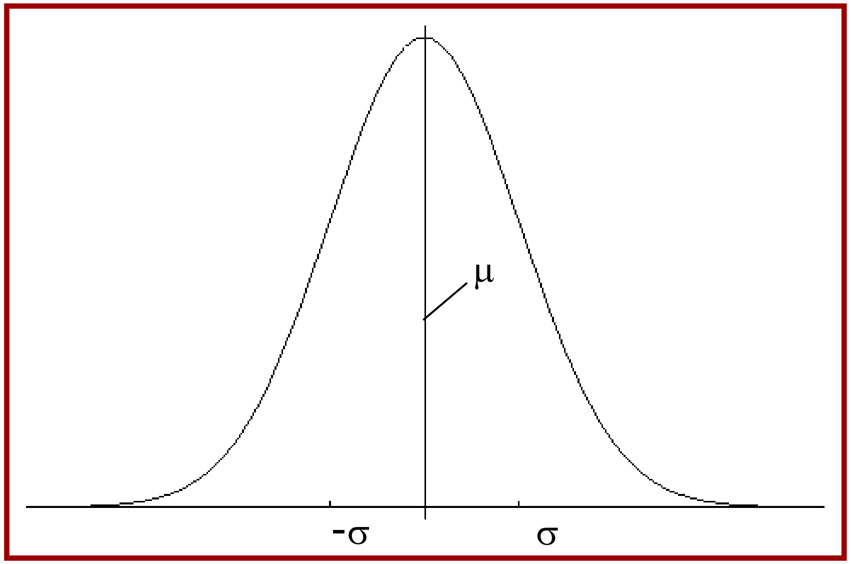
Kalman Filter

Gaussians

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

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

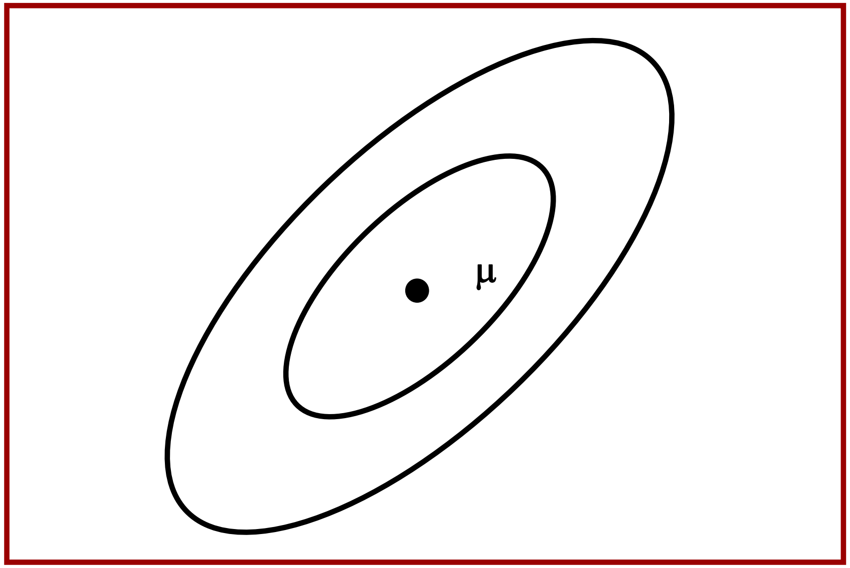
Univariate



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

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

Multivariate



Properties of Gaussians

Linear transformation:

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

Multiplication:

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

Multivariate Gaussians

Linear transformation:

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

Multiplication:

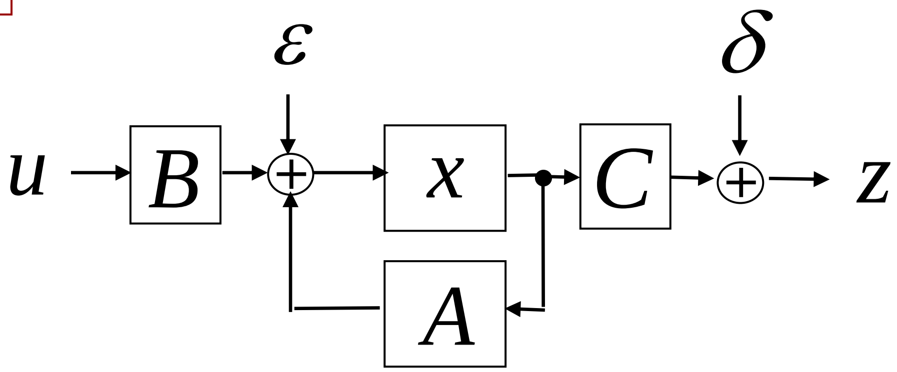
$$\left.\begin{array}{l} X_1 \sim N(\mu_1, \Sigma_1) \\ X_2 \sim N(\mu_2, \Sigma_2) \end{array}\right\} \Rightarrow p(X_1) \cdot p(X_2) \sim N\left(\frac{\Sigma_2}{\Sigma_1 + \Sigma_2} \mu_1 + \frac{\Sigma_1}{\Sigma_1 + \Sigma_2} \mu_2, \quad \frac{1}{\Sigma_1^{-1} + \Sigma_2^{-1}} \right)$$

- 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

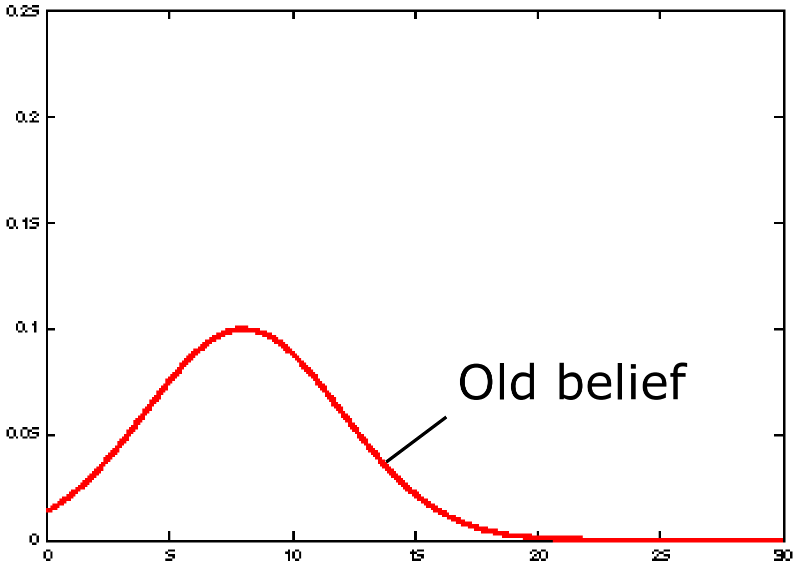
A_t Matrix (nxn) that describes how the state evolves from t-1 to t without controls or noise.

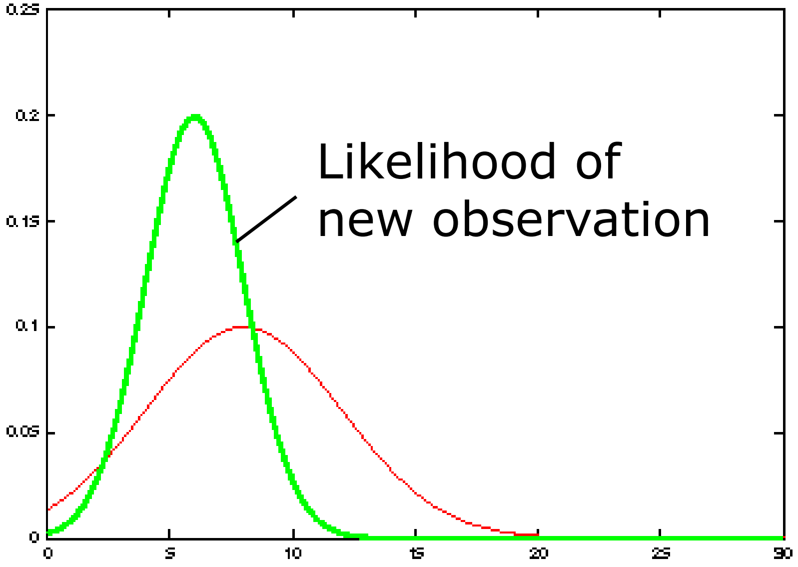
B_t Matrix (nxl) that describes how the control u_t changes the state from t-1 to t.

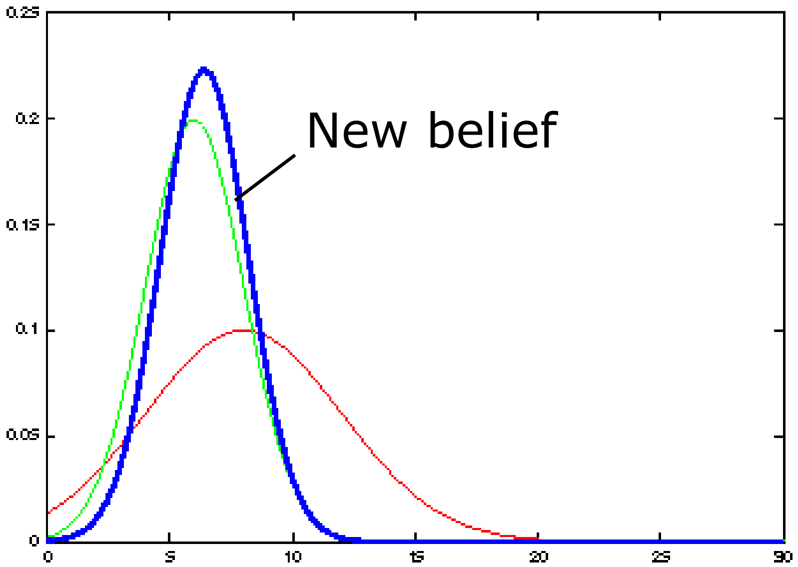
C_t Matrix (kxn) that describes how to map the state x_t to an observation z_t.

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: Correction Update







Kalman Filter: Correction Updates

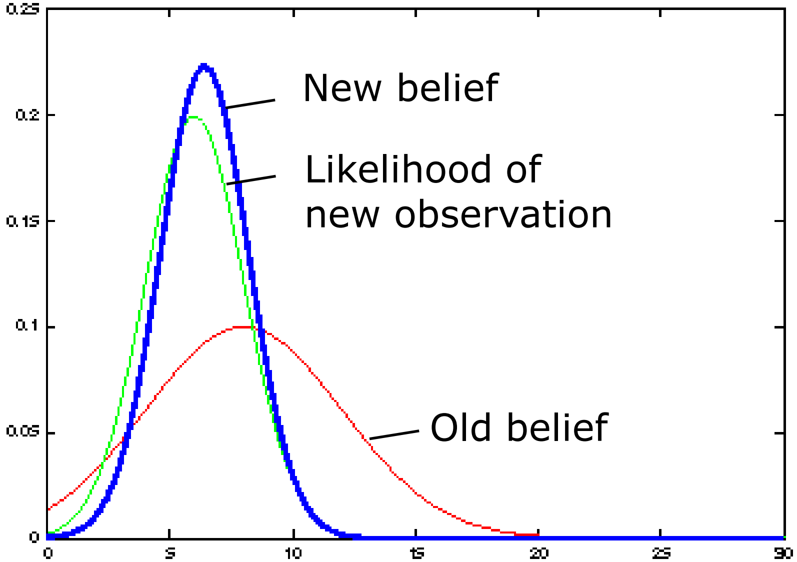
Direct measurement of state:

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

Indirect measurement of state through *C*:

fe 7aga na2sa hena.

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



- Effect of **measurement *z***

Kalman Filter: Prediction Updates

- Effect of **action *u***

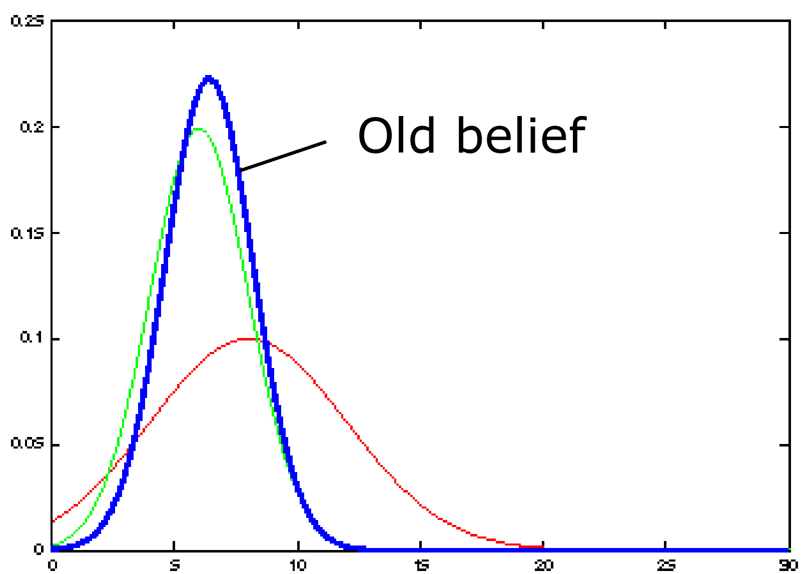
lw galy action b3ml prediction, w lw galy measurement b3ml correction.

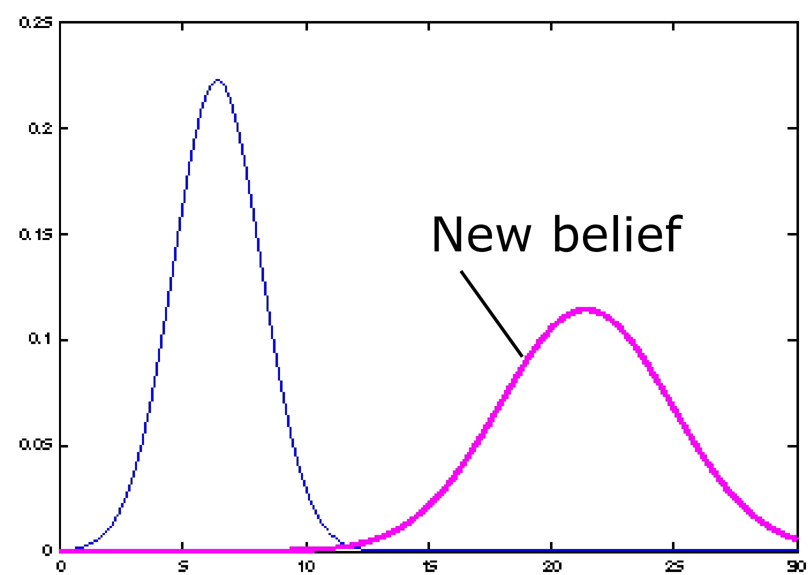
1D

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

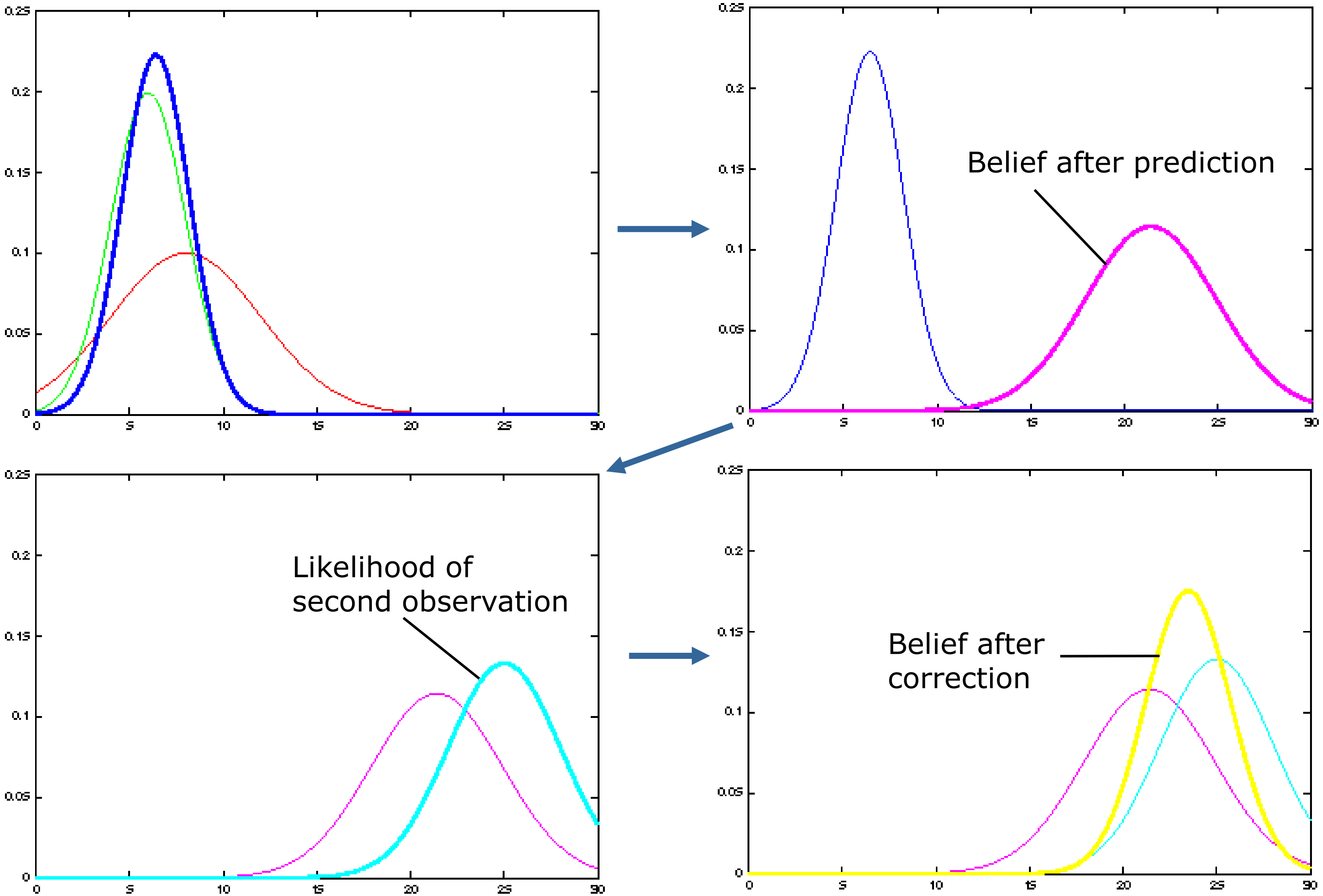
multi-D

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





Kalman Filter: Combined 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:

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

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

$$\boxed{\begin{array}{ll} \overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) & bel(x_{t-1})\, dx_{t-1} \\ \Downarrow & \Downarrow \\ \sim N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right) & \sim N\left(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}\right) \end{array}}$$

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:

$$\boxed{z_t = C_t x_t + \delta_t}$$

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

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

Linear Gaussian Systems: Observations

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

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

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

$$\Downarrow$$

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

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

lazm n7fz el slide de, 34an hanst5dmha odam.

Kalman Filter Algorithm

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

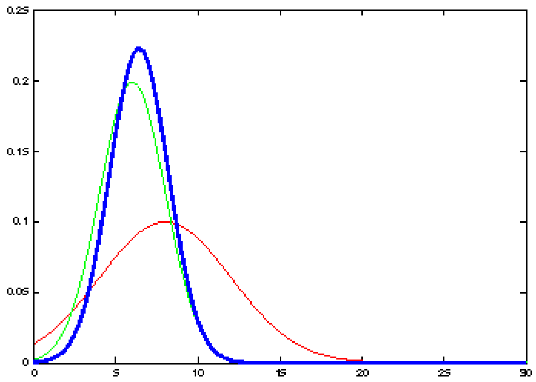
2. Prediction:
3. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$ // apply motion model
4. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

5. Correction:
6. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$ // compute Kalman gain
7. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$ // compare expected with observed measurement
8. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

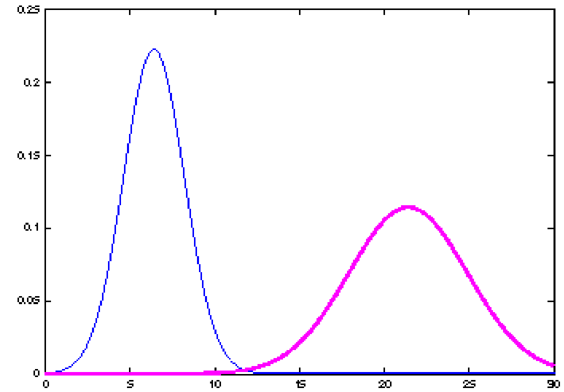
9. Return μ_t, Σ_t

The Prediction-Correction-Cycle

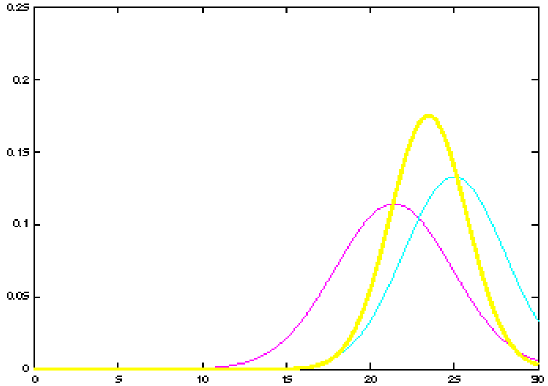
Prediction



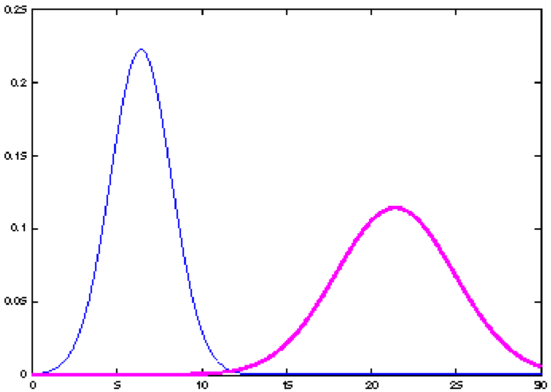
$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$



The Prediction-Correction-Cycle



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



Correction

The Prediction-Correction-Cycle

Prediction

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

1D

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

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

multi-D

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

Correction

Kalman Filter Example: Falling Mass

- Mass accelerated by gravity

$$\ddot{y}(t) = -g$$

$$\Rightarrow \dot{y}(t) = \dot{y}(t_0) - g(t - t_0)$$

$$\Rightarrow y(t) = y(t_0) + \dot{y}(t_0)(t - t_0) - \frac{g}{2}(t - t_0)^2$$

- State consists of height and vertical speed

$$\mathbf{x}(k) \equiv [y(k) \quad \dot{y}(k)]$$

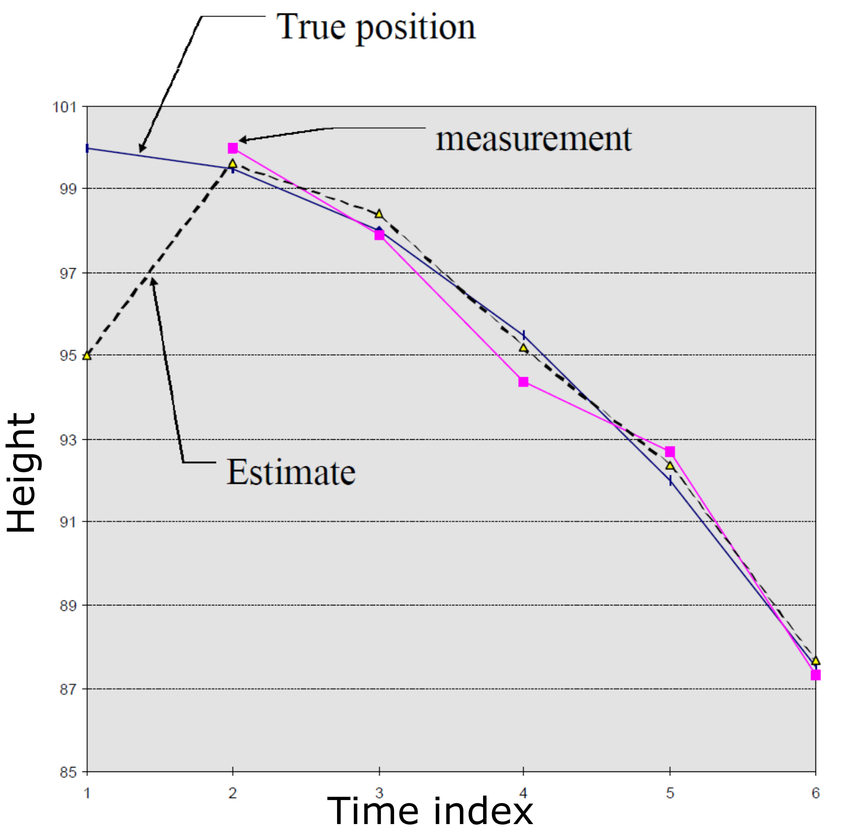
- Time increment of 1s

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} (-g)$$

- Measurement of height
- Initial state

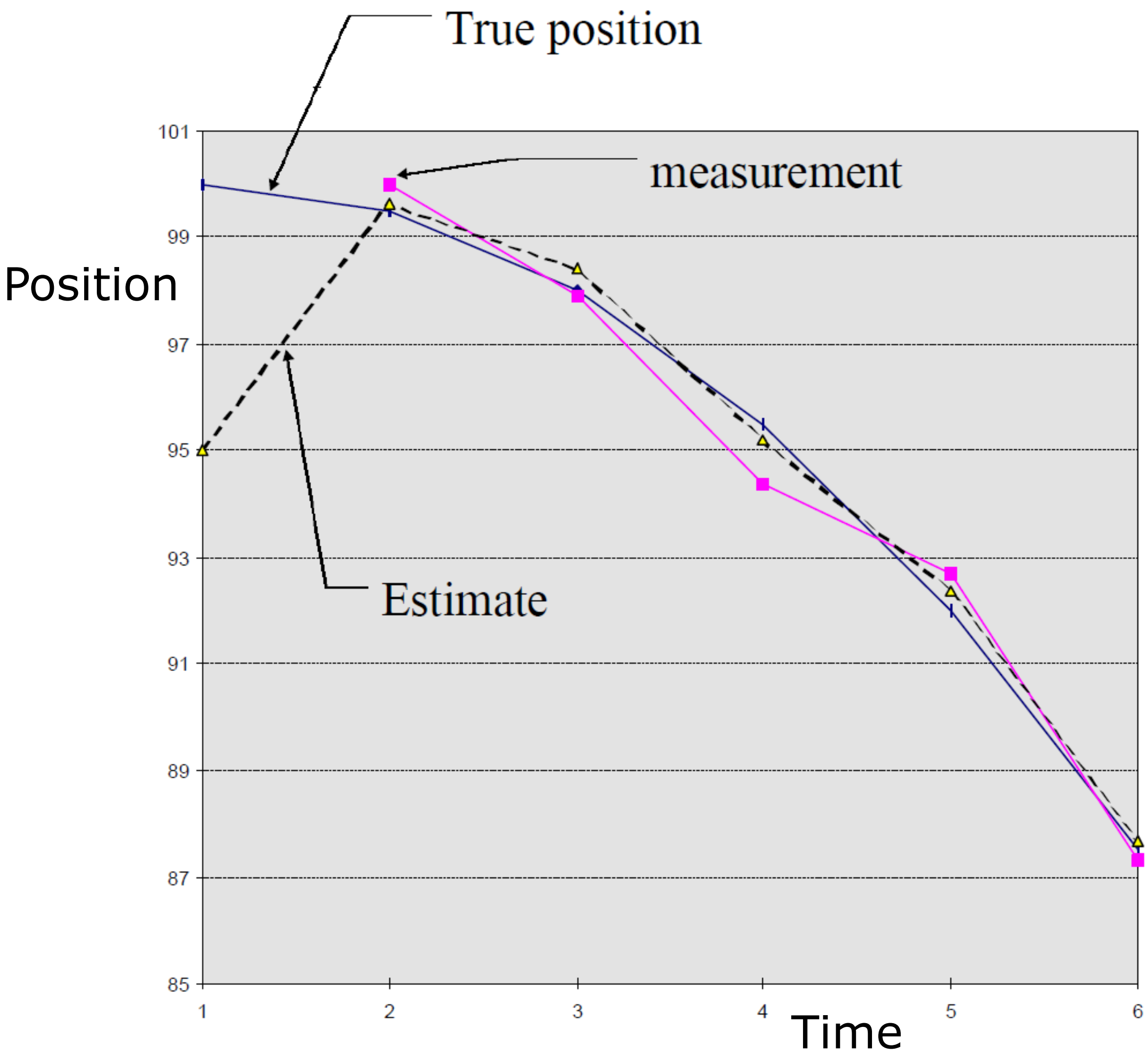
X(0) = [95,1]

[Kleeman]



Kalman Filter Example: Falling Mass

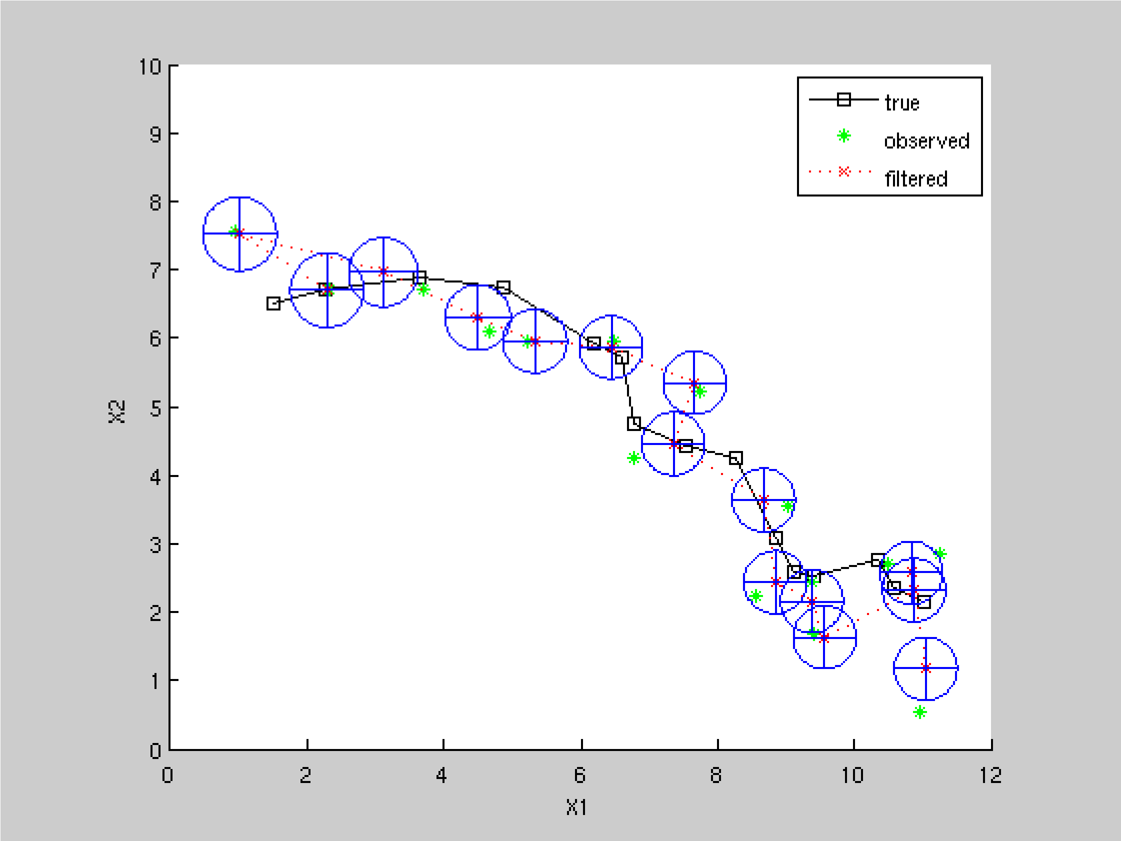
Q=1, R=0



	Estimates	
	Position	velocity
t=kT	$\hat{x}_1(k)$	$\hat{x}_2(k)$
0	95.0	1.0
1	99.63	0.38
2	98.43	-1.16
3	95.21	-2.91
4	92.35	-3.70
5	87.68	-4.84

Example Kalman Filter

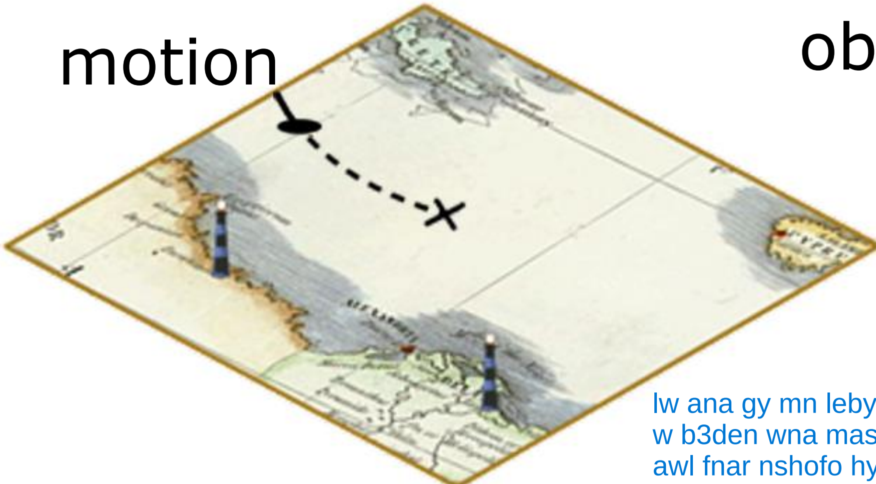
- Point moving on a plane with constant velocity + noise
- State: position, speed
- Observation: position only



De moshkela hn7lha kman 7 mo7adraat :D

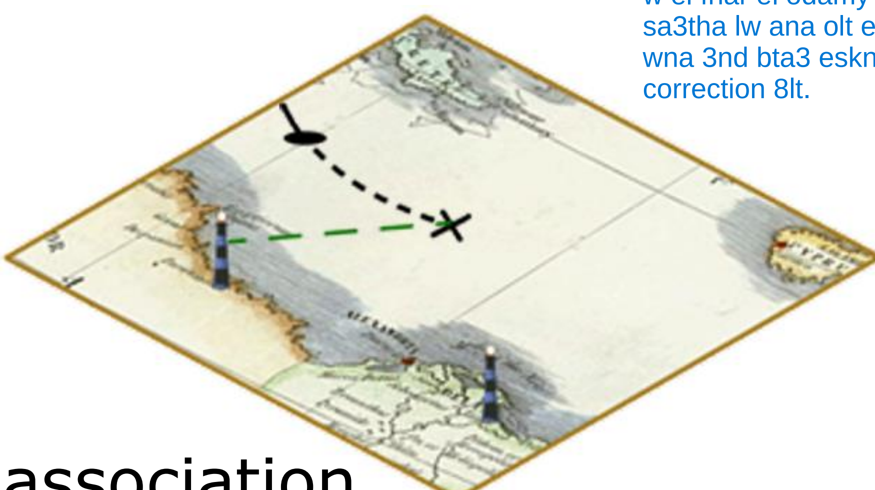
Data Association Problem

motion



observation

lw ana gy mn lebya, w mashy b safena,
w b3den wna mashy shoft fanar, el mfrod
awl fnar nshofo hykon bta3 marsa matro7
lakn ana 34an el denya mghyma kont 3det bta3 marsa matro7
w el fnar el odamy da bta3 eskndry bs ana msh 3aref,
sa3tha lw ana olt en da bta3 marsa matro7
wna 3nd bta3 eskndrya, fa kedda h3ml
correction 8lt.



association

correction

What happens when this lighthouse is assigned?

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!

Acknowledgment

- These slides have been created by Wolfram Burgard, Dieter Fox, Cyrill Stachniss and Maren Bennewitz