Principles of Robot Autonomy I

Bayes Filter, Kalman Filter

Stanford University

MSL

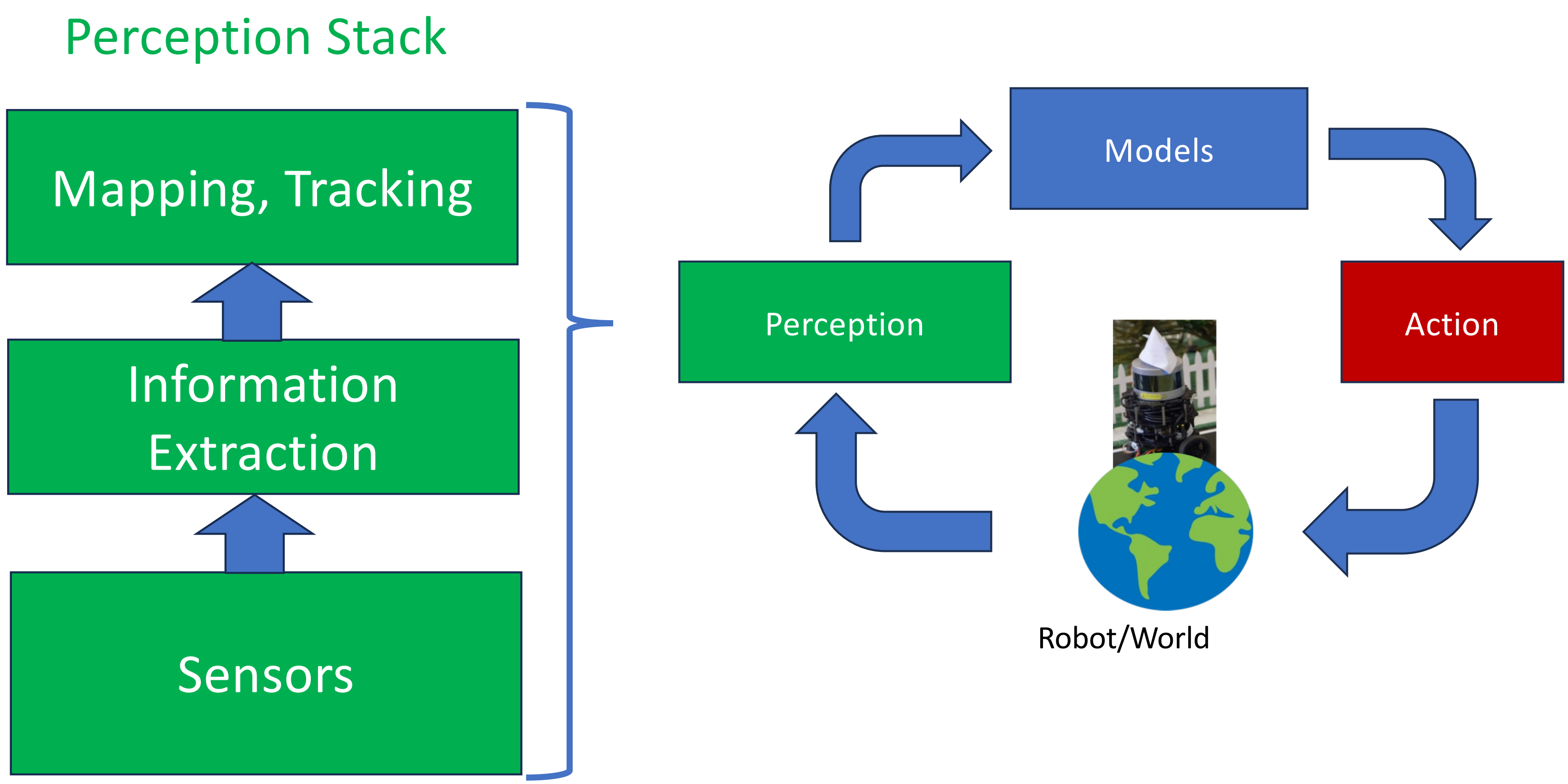
IPRL

ASL Autonomous Systems Lab

Logistics

- Homework 4: due Thurs 11/13
- Homework 5: out Thurs 11/13, Due 12/2 (last one!)
- Guest Lecture: Dr. Vincent Vincent Vanhoucke, Distinguished Engineer, Waymo, Thurs 11/20
- Midterm grades out soon
- Lecture 14:
  - Occ grid mapping
  - Frontier exploration, information gathering
  - General Bayesian filter

Robot Perception

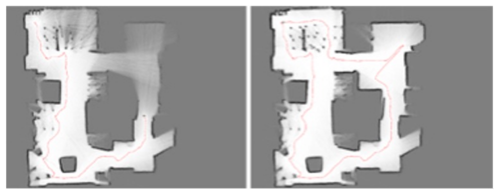


Perception Stack: Kalman Filtering

Use sensor data to update the models

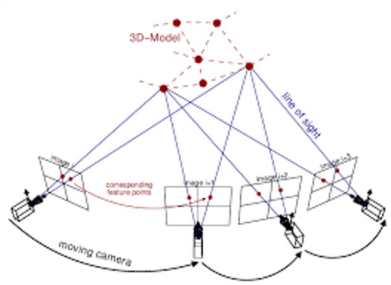
Localization, Mapping, Tracking:

- EKF/Monte Carlo localization
- Occupancy grid mapping
- Factor graphs/SLAM
- Tracking (EKF and Particle Filter)
- **AA273: Filtering (Schwager)**
- **AA275: Navigation (Gao)**



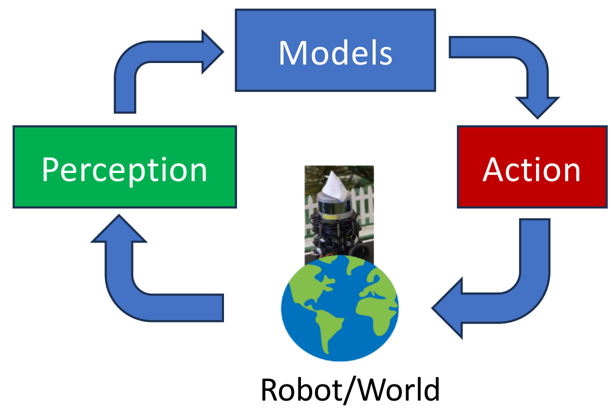
Information extraction

- Computer vision: features, correspondences, Structure from Motion (SfM), depth
- Lidar scan matching, ICP
- **CS231A: Comp Vision**

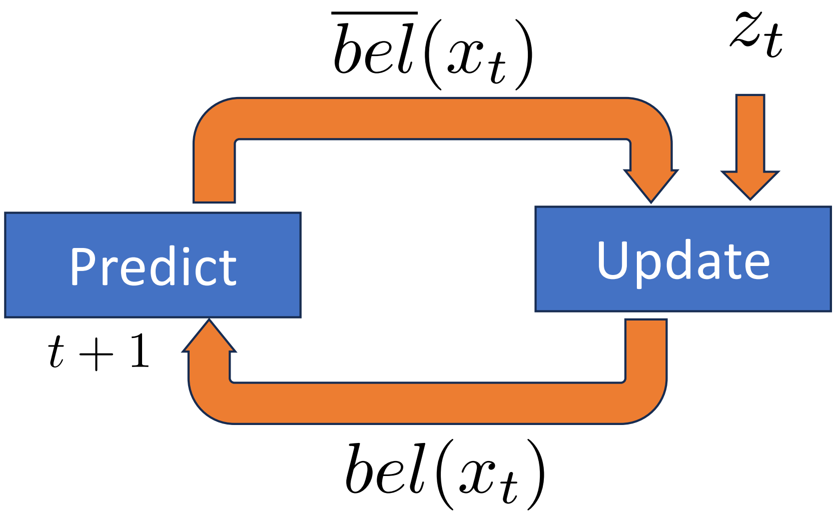


Sensors:

- RGB Cameras, RGB-D/stereo cameras, Lidar
- IMU, GPS, wheel encoders



Bayes filter algorithm



- Algorithm initialized with $bel(x_0)$ (e.g., uniform or points mass)

Data: $bel(x_{t-1}), u_t, z_t$
Result: $bel(x_t)$
foreach x_t **do**
  $\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\, bel(x_{t-1})\, dx_{t-1};$
  $bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t);$
end
Return $bel(x_t)$

Model for robot-environment interaction

- Robot state (unknown): x_t

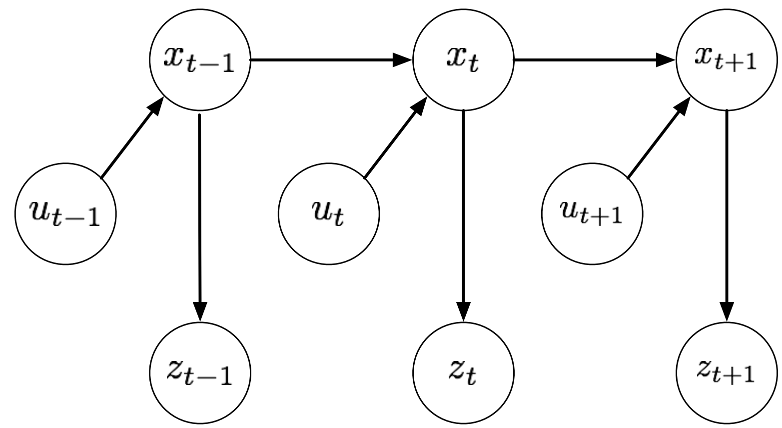
$$x_{0:t} = x_0, x_2, \ldots, x_t$$

- Measurement (known): z_t

$$z_{1:t} = z_1, z_2, \ldots, z_t$$

- Control input (known):

$$u_{1:t} = u_1, u_2, \ldots, u_t$$



State equation

- General probabilistic generative model

Convention: first take control action and then take measurement

$$p(x_t \,|\, x_{0:t-1},\, z_{1:t-1},\, u_{1:t})$$

- Key assumption: Markovianity

State transition probability

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

- In other words, we assume *conditional independence*, conditioned on x_{t-1}
- Special case (typical dynamics model):

$$x_t = f(x_{t-1}, u_t) + w_{t-1}, \quad w_{t-1} \sim \mathcal{N}(0, Q_{t-1})$$

Measurement equation and overall stochastic model

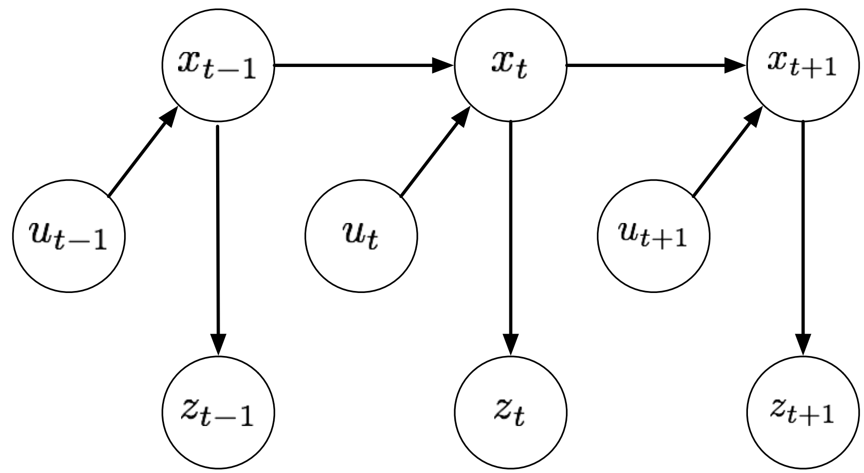
- Conditionally independent measurements

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

Measurement probability

- Overall dynamic Bayes network model (also referred to as hidden Markov model)
- Special case (typical measurement model):

$$z_t = g(x_t, u_t) + v_t, \quad v_t \sim \mathcal{N}(0, R_t)$$



Belief distribution

- Belief distribution: $bel(x_t) := p(x_t \mid z_{1:t},\, u_{1:t})$
- Also call the Bayesian posterior
- Probabilities over state variables conditioned on the available data

- Prediction distribution: $\overline{bel}(x_{t+1}) := p(x_{t+1} \mid z_{1:t}, u_{1:t+1})$
- Probabilities over state projected one step into the future

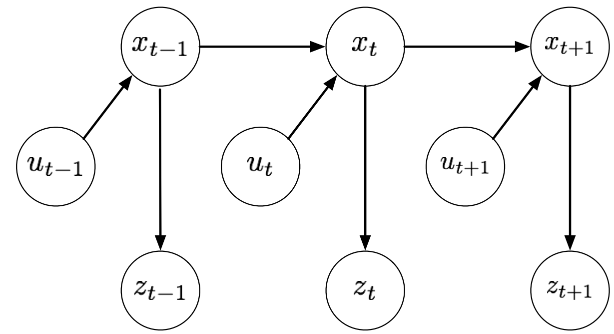
Bayes filter algorithm

- Bayes' filter algorithm: most general algorithm for calculating beliefs
- Key assumption: state is complete

- Recursive algorithm
  - Step 1 (prediction): compute $\overline{bel}\,(x_t)$
  - Step 2 (measurement update): compute $bel(x_t)$
- Algorithm initialized with $bel(x_0)$ (e.g., uniform or points mass)

Data: $bel(x_{t-1}), u_t, z_t$
Result: $bel(x_t)$
foreach x_t **do**
  $\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\, bel(x_{t-1})\, dx_{t-1};$
  $bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t);$
end
Return $bel(x_t)$

Derivation: measurement update



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

Recall by definition:

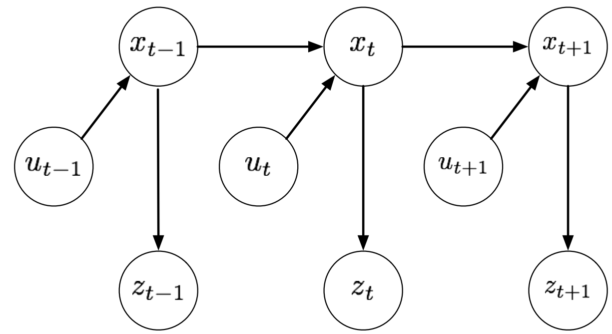
$$p(x \mid y) = \frac{p(x, y)}{p(y)}$$

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

$$= \frac{p(z_t \mid x_t, z_{1:t-1},\, u_{1:t})\, p(x_t \mid z_{1:t-1},\, u_{1:t})}{\underbrace{p(z_t \mid z_{1:t-1},\, u_{1:t})}_{:=\eta^{-1}}}$$ Bayes rule

$$= \eta\, p(z_t \mid x_t)\, \underbrace{p(x_t \mid z_{1:t-1},\, u_{1:t})}_{=\overline{bel}(x_t)}$$ Markov property

Derivation: prediction step



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

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

Total probability

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

Recall by definition:
$p(x, y) = p(x \mid y) p(y)$

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

Markov

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

Discrete Bayes' filter

- Discrete Bayes' filter algorithm: applies to problems with *finite* state spaces

- Belief $bel(x_t)$ represented as pmf $\{p_{k,t}\}$

Data: $\{p_{k,t-1}\}, u_t, z_t$
Result: $\{p_{k,t}\}$
foreach k **do**
$\quad \bar{p}_{k,t} = \sum_i p(X_t = x_k \,|\, u_t, X_{t-1} = x_i)\, p_{i,t-1};$
$\quad p_{k,t} = \eta\, p(z_t \,|\, X_t = x_k)\, \bar{p}_{k,t};$
end
Return $\{p_{k,t}\}$

Simple example – Belief & Measurement Model

Adapted from [PR]

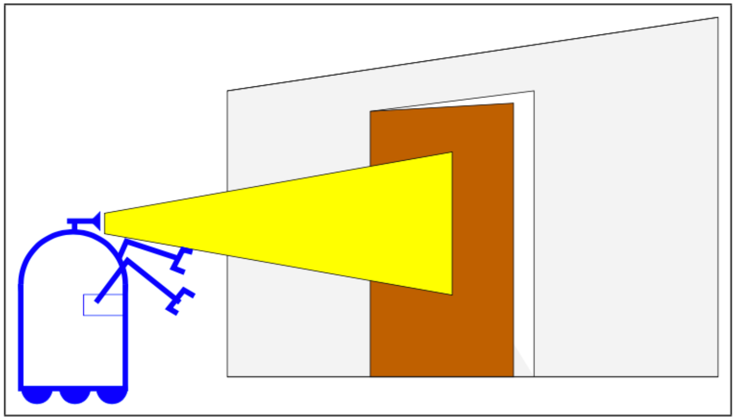


Figure 2.2 A mobile robot estimating the state of a door.

$$\begin{aligned} bel(X_0 = \textbf{open}) &= 0.5 \\ bel(X_0 = \textbf{closed}) &= 0.5 \end{aligned}$$

$$\begin{aligned} p(Z_t = \textbf{sense\_open} \mid X_t = \textbf{is\_open}) &= 0.6 \\ p(Z_t = \textbf{sense\_closed} \mid X_t = \textbf{is\_open}) &= 0.4 \end{aligned}$$

$$\begin{aligned} p(Z_t = \textbf{sense\_open} \mid X_t = \textbf{is\_closed}) &= 0.2 \\ p(Z_t = \textbf{sense\_closed} \mid X_t = \textbf{is\_closed}) &= 0.8 \end{aligned}$$

Simple example – Transition Model

Adapted from [PR]

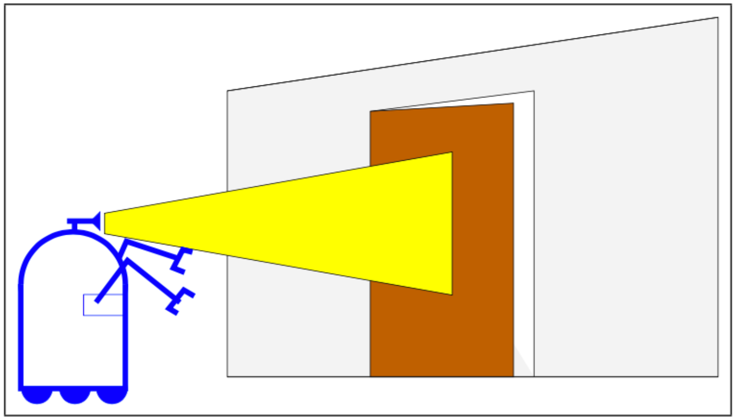


Figure 2.2 A mobile robot estimating the state of a door.

$$
\begin{aligned}
p(X_t = \textbf{is\_open} \mid U_t = \textbf{push}, X_{t\_1} = \textbf{is\_open}) &= 1 \\
p(X_t = \textbf{is\_closed} \mid U_t = \textbf{push}, X_{t\_1} = \textbf{is\_open}) &= 0 \\
p(X_t = \textbf{is\_open} \mid U_t = \textbf{push}, X_{t\_1} = \textbf{is\_closed}) &= 0.8 \\
p(X_t = \textbf{is\_closed} \mid U_t = \textbf{push}, X_{t\_1} = \textbf{is\_closed}) &= 0.2
\end{aligned}
$$

$$
\begin{aligned}
p(X_t = \textbf{is\_open} \mid U_t = \textbf{do\_nothing}, X_{t\_1} = \textbf{is\_open}) &= 1 \\
p(X_t = \textbf{is\_closed} \mid U_t = \textbf{do\_nothing}, X_{t\_1} = \textbf{is\_open}) &= 0 \\
p(X_t = \textbf{is\_open} \mid U_t = \textbf{do\_nothing}, X_{t\_1} = \textbf{is\_closed}) &= 0 \\
p(X_t = \textbf{is\_closed} \mid U_t = \textbf{do\_nothing}, X_{t\_1} = \textbf{is\_closed}) &= 1
\end{aligned}
$$

Simple example – Prediction step

Adapted from [PR]

$$
\begin{aligned}
bel(X_0 = \textbf{open}) &= 0.5 \\
bel(X_0 = \textbf{closed}) &= 0.5
\end{aligned}
$$

Data: $\{p_{k,t-1}\}, u_t, z_t$
Result: $\{p_{k,t}\}$
foreach k **do**
  $\boxed{\overline{p}_{k,t} = \sum_i p(X_t = x_k \mid u_t, X_{t-1} = x_i)\, p_{i,t-1};}$
  $p_{k,t} = \eta\, p(z_t \mid X_t = x_k)\, \overline{p}_{k,t};$
end
Return $\{p_{k,t}\}$

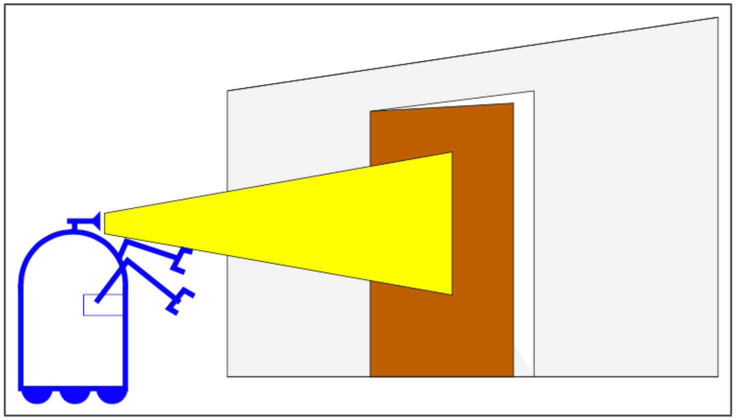


Figure 2.2 A mobile robot estimating the state of a door.

$$
\begin{aligned}
p(X_t = \textbf{is\_open} \mid U_t = \textbf{push}, X_{t\_1} = \textbf{is\_open}) &= 1 \\
p(X_t = \textbf{is\_closed} \mid U_t = \textbf{push}, X_{t\_1} = \textbf{is\_open}) &= 0 \\
p(X_t = \textbf{is\_open} \mid U_t = \textbf{push}, X_{t\_1} = \textbf{is\_closed}) &= 0.8 \\
p(X_t = \textbf{is\_closed} \mid U_t = \textbf{push}, X_{t\_1} = \textbf{is\_closed}) &= 0.2
\end{aligned}
$$

$$
\boxed{
\begin{aligned}
p(X_t = \textbf{is\_open} \mid U_t = \textbf{do\_nothing}, X_{t\_1} = \textbf{is\_open}) &= 1 \\
p(X_t = \textbf{is\_closed} \mid U_t = \textbf{do\_nothing}, X_{t\_1} = \textbf{is\_open}) &= 0 \\
p(X_t = \textbf{is\_open} \mid U_t = \textbf{do\_nothing}, X_{t\_1} = \textbf{is\_closed}) &= 0 \\
p(X_t = \textbf{is\_closed} \mid U_t = \textbf{do\_nothing}, X_{t\_1} = \textbf{is\_closed}) &= 1
\end{aligned}
}
$$

$$\overline{bel}(x_{t+1}) = \{0.5, 0.5\}$$

Simple Example – Update Step

Adapted from [PR]

- Is door open or not?

$$\overline{bel}(x_t) = \{0.5, 0.5\}$$

Data: $\{p_{k,t-1}\}, u_t, z_t$
Result: $\{p_{k,t}\}$
foreach k **do**
  $\bar{p}_{k,t} = \sum_i p(X_t = x_k \mid u_t, X_{t-1} = x_i)\, p_{i,t-1};$
  $\boxed{p_{k,t} = \eta\, p(z_t \mid X_t = x_k)\, \bar{p}_{k,t};}$
end
Return $\{p_{k,t}\}$

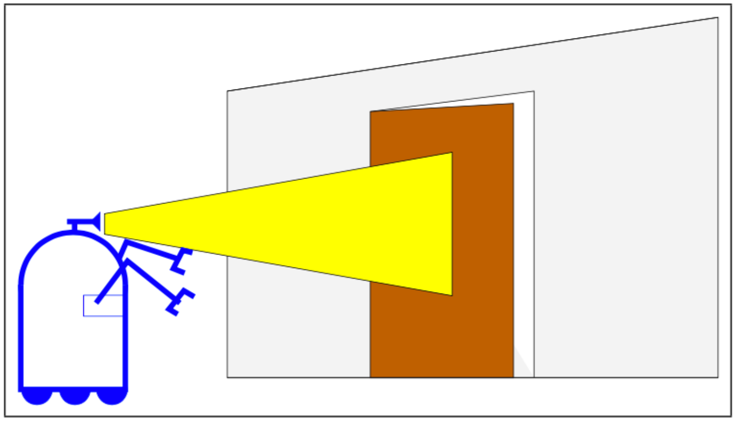


Figure 2.2 A mobile robot estimating the state of a door.

Measurement Model

$$\begin{aligned}
p(Z_t = \textbf{sense_open} \mid X_t = \textbf{is_open}) &= 0.6 \\
p(Z_t = \textbf{sense_closed} \mid X_t = \textbf{is_open}) &= 0.4
\end{aligned}$$

$$\begin{aligned}
p(Z_t = \textbf{sense_open} \mid X_t = \textbf{is_closed}) &= 0.2 \\
p(Z_t = \textbf{sense_closed} \mid X_t = \textbf{is_closed}) &= 0.8
\end{aligned}$$

Transition Model for do_nothing

$$\begin{aligned}
p(X_t = \textbf{is_open} \mid U_t = \textbf{do_nothing}, X_{t\text{-}1} = \textbf{is_open}) &= 1 \\
p(X_t = \textbf{is_closed} \mid U_t = \textbf{do_nothing}, X_{t\text{-}1} = \textbf{is_open}) &= 0 \\
p(X_t = \textbf{is_open} \mid U_t = \textbf{do_nothing}, X_{t\text{-}1} = \textbf{is_closed}) &= 0 \\
p(X_t = \textbf{is_closed} \mid U_t = \textbf{do_nothing}, X_{t\text{-}1} = \textbf{is_closed}) &= 1
\end{aligned}$$

Simple Example – Update Step

Adapted from [PR]

- Is door open or not?

$$\overline{bel}(x_t) = \{0.5, 0.5\}$$

Received Sensor Measurement:

$$bel(x_1) = \eta\, p(Z_1 = \textbf{sense\_open} \mid x_1)\, \overline{bel}(x_1)$$

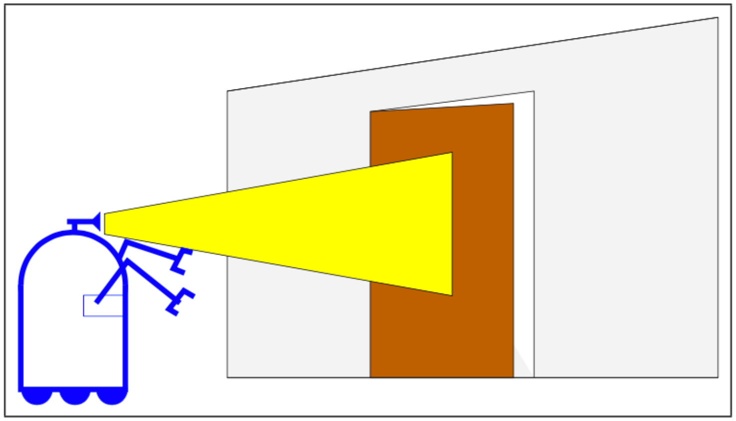


Figure 2.2 A mobile robot estimating the state of a door.

Measurement Model

$$\boxed{p(Z_t = \textbf{sense\_open} \mid X_t = \textbf{is\_open}) = 0.6}$$
$$p(Z_t = \textbf{sense\_closed} \mid X_t = \textbf{is\_open}) = 0.4$$
$$\boxed{p(Z_t = \textbf{sense\_open} \mid X_t = \textbf{is\_closed}) = 0.2}$$
$$p(Z_t = \textbf{sense\_closed} \mid X_t = \textbf{is\_closed}) = 0.8$$

Transition Model for do_nothing

$$\begin{aligned}
p(X_t = \textbf{is_open} \mid U_t = \textbf{do_nothing}, X_{t_1} = \textbf{is_open}) &= 1 \\
p(X_t = \textbf{is_closed} \mid U_t = \textbf{do_nothing}, X_{t_1} = \textbf{is_open}) &= 0 \\
p(X_t = \textbf{is_open} \mid U_t = \textbf{do_nothing}, X_{t_1} = \textbf{is_closed}) &= 0 \\
p(X_t = \textbf{is_closed} \mid U_t = \textbf{do_nothing}, X_{t_1} = \textbf{is_closed}) &= 1
\end{aligned}$$

$\boldsymbol{bel(x_t) = (0.75, 0.25)}$ Don't forget normalization!

Instantiating the Bayes' filter

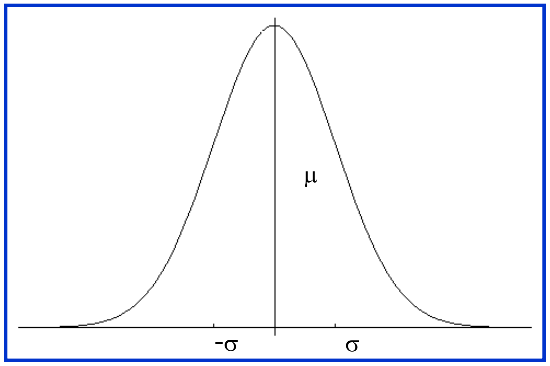
- Tractable implementations of Bayes' filter exploit structure and / or approximations; two main classes
  - Discrete Bayesian filter (a.k.a., Histogram filter)
  - Parametric filters: e.g., Kalman Filter, EKF, UKF, etc.
  - Non parametric filters: e.g., particle filter, etc.

Gaussian distributions

- Key idea: belief represented as multivariate normal distribution

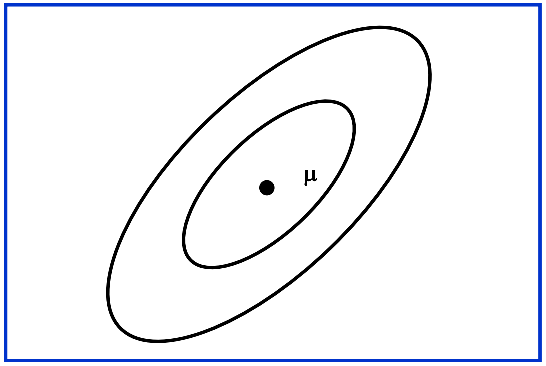
Univariate

$$p(x) = (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right)$$
$$\sim \mathcal{N}(x; \mu, \sigma^2)$$



Multivariate

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$
$$\sim \mathcal{N}(\mu, \Sigma)$$



Key properties of Gaussian random variables

- If $X \sim \mathcal{N}(\mu,\, \Sigma)$ then

$$Y = AX + b \sim \mathcal{N}(A\mu + b,\, A\Sigma A^T)$$

- The sum of two independent Gaussian RVs

$$X_i \sim \mathcal{N}(\mu_i,\, \Sigma_i), \qquad i = 1, 2$$

is Gaussian, specifically

$$X_1 + X_2 \sim \mathcal{N}(\mu_1 + \mu_2,\, \Sigma_1 + \Sigma_2)$$

- The product of Gaussian pdf is also Gaussian

Kalman filter (KF)

- Assumption #1: linear dynamics

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

  - Gaussian white noise process ϵ_t is $\mathcal{N}(0, R_t)$
- Assumption #1 implies that the transition distribution is Gaussian

$$p(x_t \mid u_t, x_{t-1}) = \det(2\pi R_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1}(x_t - A_t x_{t-1} - B_t u_t)\right)$$

Kalman filter (KF)

- Assumption #2: linear measurement model

$$z_t = C_t x_t + \delta_t$$

  - Independent measurement noise δ_t is $\mathcal{N}(0, Q_t)$

- Assumption #2 implies that the measurement probability is Gaussian

$$p(z_t \mid x_t) = \det(2\pi Q_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1} (z_t - C_t x_t)\right)$$

Kalman filter (KF)

- Assumption #3: the initial belief is Gaussian

$$bel(x_0) = p(x_0) = \det(2\pi\Sigma_0)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_0 - \mu_0)^T \Sigma_0^{-1}(x_0 - \mu_0)\right)$$

- Key fact: These three assumptions ensure that the posterior $bel(x_t)$ is Gaussian for all *t*, i.e., $bel(x_t) = \mathcal{N}(\mu_t, \Sigma_t)$
- Note:
  - KF implements belief computation for continuous states
  - Gaussians are unimodal -> commitment to single-hypothesis filtering

Recap – Bayes Filter

Data: $bel(x_{t-1}), u_t, z_t$
Result: $bel(x_t)$
foreach x_t **do**

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

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

end
Return $bel(x_t)$

Kalman filter: algorithm

Prediction

Project state ahead

$$\overline{\mu}_t = A_t \mu_{t-1} + B_t u_t$$

Project covariance ahead

$$\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

Correction

Compute Kalman gain

$$K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}$$

Update estimate with new measurement

$$\mu_t = \overline{\mu}_t + K_t (z_t - C_t \overline{\mu}_t)$$

Update covariance

$$\Sigma_t = (I - K_t C_t) \overline{\Sigma}_t$$

$$\textbf{Data: } \overbrace{(\mu_{t-1}, \Sigma_{t-1})}^{bel(x_{t-1})}, u_t, z_t$$

$$\textbf{Result: } (\mu_t, \Sigma_t)$$

Prediction: $\overline{bel}(x_t)$

$$\overline{\mu}_t = A_t \mu_{t-1} + B_t u_t \;;$$
$$\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t;$$

Correction: $bel(x_t)$

$$K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1};$$
$$\mu_t = \overline{\mu}_t + K_t (z_t - C_t \overline{\mu}_t);$$
$$\Sigma_t = (I - K_t C_t) \overline{\Sigma}_t;$$

$$\text{Return } \underbrace{(\mu_t, \Sigma_t)}_{bel(x_t)}$$

Kalman filter: derivation (sketch)

- Prediction

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

$$\mathcal{N}(A_t x_{t-1} + B_t u_t,\, R_t) \qquad \mathcal{N}(\mu_{t-1},\, \Sigma_{t-1})$$

- Recalling that $x_t = A_t x_{t-1} + B_t u_t + \epsilon_t$

$$\overline{bel}(x_t) = \mathcal{N}(\overline{\mu}_t,\, \overline{\Sigma}_t)$$

with

$$\overline{\mu}_t = A_t \mu_{t-1} + B_t u_t$$
$$\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

Kalman filter: derivation (sketch)

- Correction

$$bel(x_t) = \eta \; \underbrace{p(z_t \mid x_t)}_{\mathcal{N}(C_t x_t,\, Q_t)} \quad \cdot \quad \underbrace{\overline{bel(x_t)}}_{\mathcal{N}(\bar{\mu}_t,\, \bar{\Sigma}_t)}$$

- After some algebraic manipulations

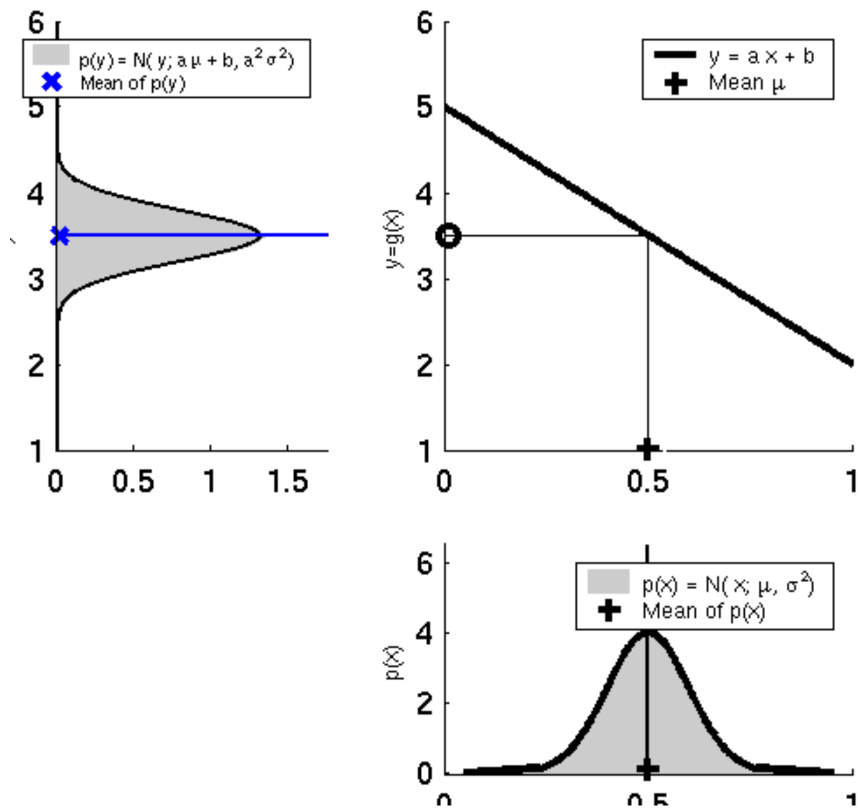
$$bel(x_t) = \mathcal{N}(\mu_t,\, \Sigma_t)$$

with

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

- Other derivations are possible; see, e.g., R. E. Kalman, A new approach to linear filtering and prediction problems. Journal of Basic Engineering, 82(1), 35-45, 1960.

Revisiting linearity assumption



- KF crucially exploits the property that a linear transformation of a Gaussian RV results in a Gaussian RV
- However, linearity assumptions are severely restrictive for robotics applications