ELEC 3210
Introduction to Mobile Robotics
Lecture 9

(Machine Learning and Infomation Processing for Robotics)

Huan YIN

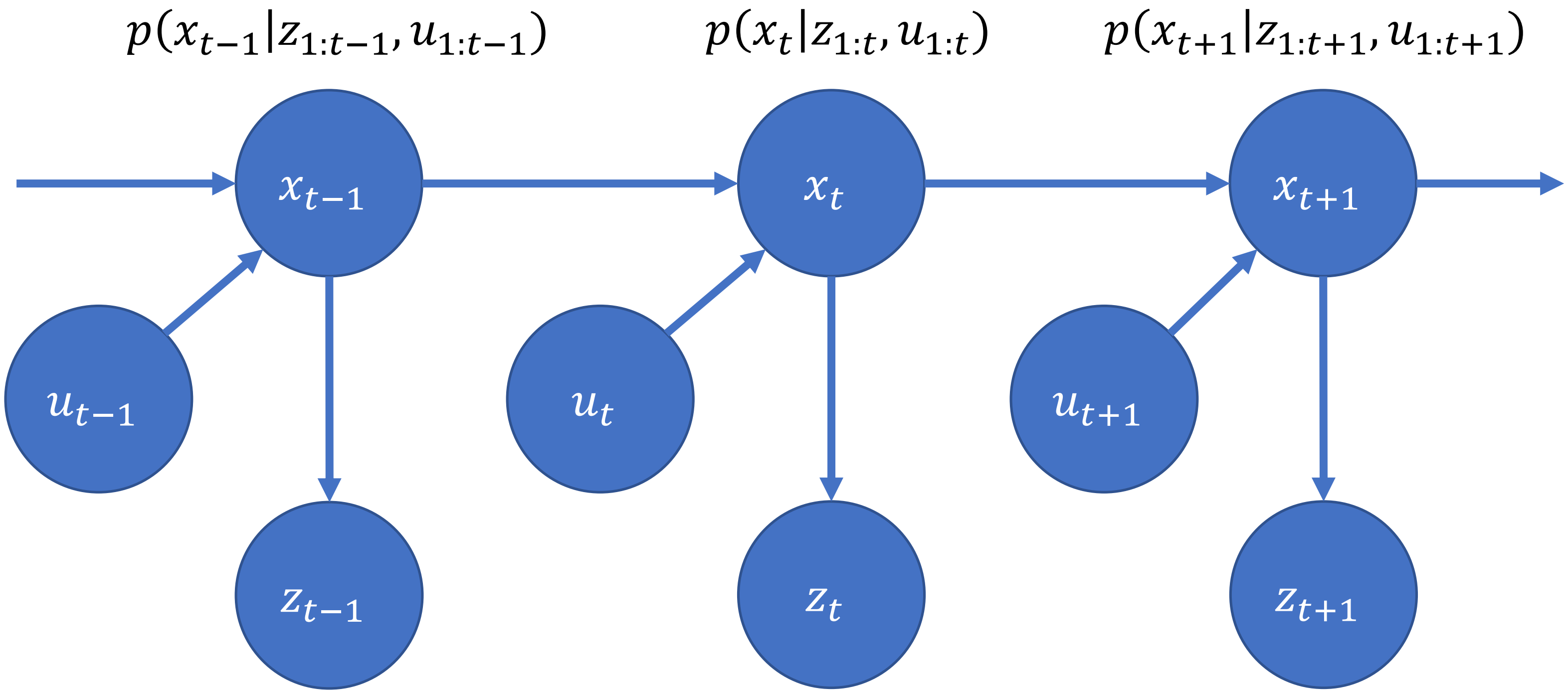
Research Assistant Professor, Dept. of ECE

eehyin@ust.hk




Recap L7 - Bayes Filter

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



Recap L8 - Particle Filter

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

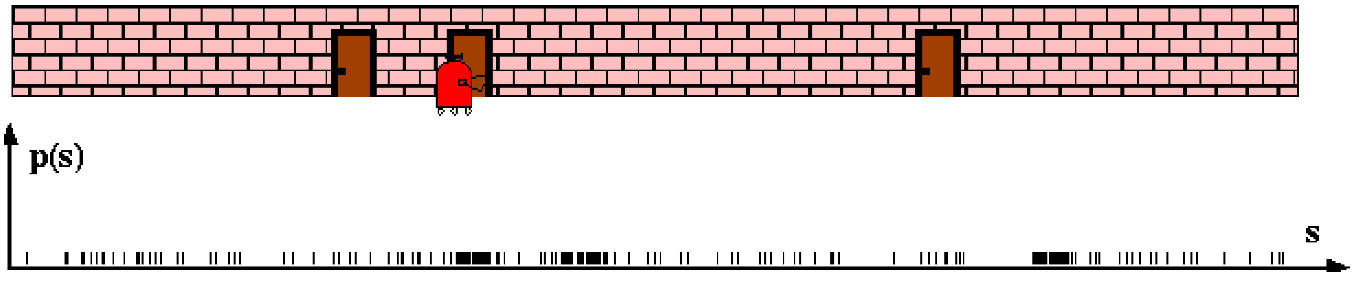
draw x^i_{t-1} from $Bel(x_{t-1})$

draw x^i_t from $p(x_t \mid x^i_{t-1}, u_{t-1})$

Importance factor for x^i_t:

$$w^i_t = \frac{\text{target distribution}}{\text{proposal distribution}}$$

$$\propto p(z_t \mid x_t)$$



Recap L8 - Particle Filter

- Pros (Compared to KF family)
  - Easy to implement
  - Able to handle nonlinear systems without linearization
  - Able to represent arbitrary distribution
- Cons
  - Particle degeneracy problem
  - Need lots of particles to represent high dimensional state space, computational complexity increases significantly w.r.t state dimension

Assumptions of Kalman Filter

- The prior state of the robot is represented by a **Gaussian distribution**
  - $p(x_0) \sim N(\mu_0, \Sigma_0)$
- The process model $g(x_t \mid x_{t-1}, u_t)$ is linear with **additive Gaussian white noise**
  - $x_t = A_t\, x_{t-1} + B_t\, u_t + n_t$
  - $n_t \sim N(0, Q_t)$
  - $x_t, n_t \in \mathbf{R}^n, u_t \in \mathbf{R}^m$, $A_t, Q_t \in \mathbf{R}^{n\times n}$, and $B_t \in \mathbf{R}^{n\times m}$
- The measurement model $h(z_t \mid x_t)$ is linear with **additive Gaussian white noise**
  - $z_t = C_t\, x_t + v_t$
  - $v_t \sim N(0, R_t)$
  - $z_t, v_t \in \mathbf{R}^p, C_t \in \mathbf{R}^{p\times n}$, and $R_t \in \mathbf{R}^{p\times p}$

Courtesy: Shaojie Shen

Rudolf E. Kálmán

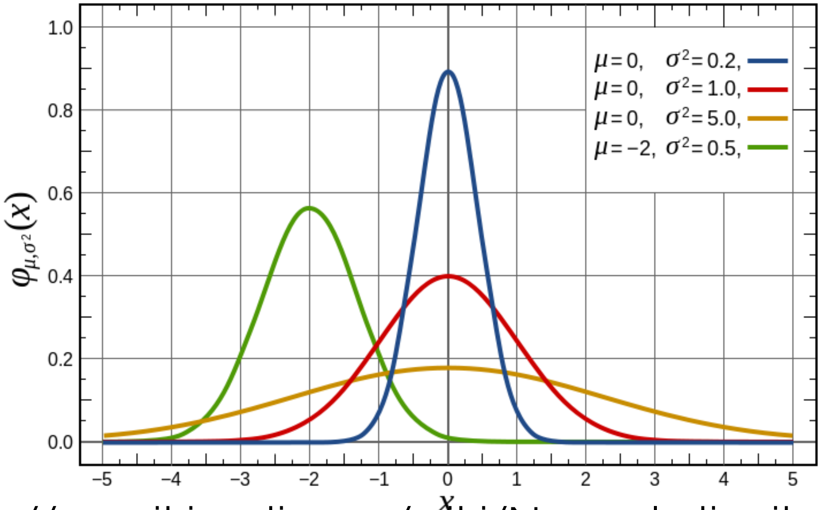
- US President Obama honors Prof. Rudolf Kalman (1930-2016) with National Medal of Science

Courtesy: ETH Life

Gaussian Random Variables

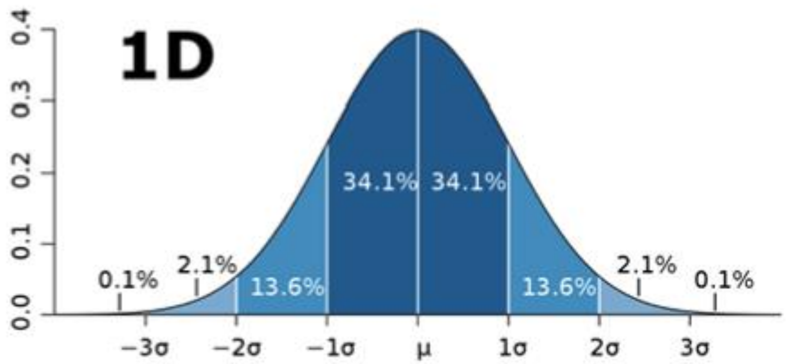
Multivariate Normal (Gaussian) Distribution

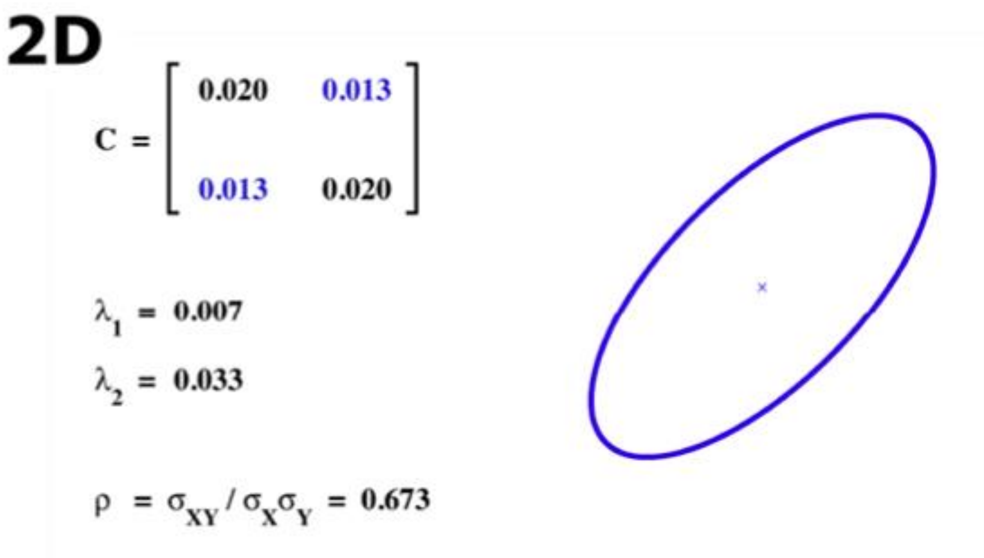
- Let X be a vector of n random variables
- A multivariate normal distribution takes the form
- $f_X(x) = \frac{1}{(2\pi)^{n/2}\sqrt{\det(\Sigma)}} e^{\frac{-(x-\mu)^T\Sigma^{-1}(x-\mu)}{2}}$
- where mean $\mu \in \mathbf{R}^n$ and covariance $\Sigma \in \mathbf{R}^{n\times n}$
- Fully parameterized by μ, Σ

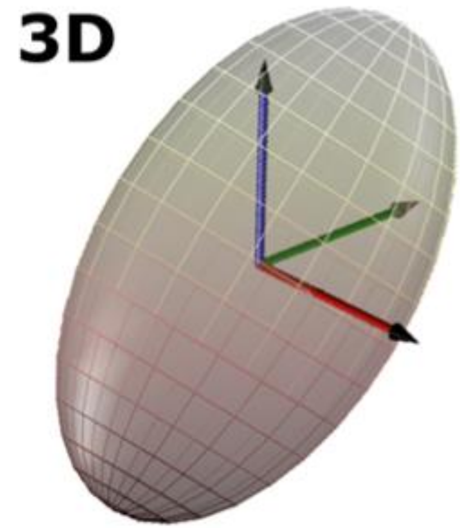


[http://en.wikipedia.org/wiki/Normal_distribution]

Gaussians





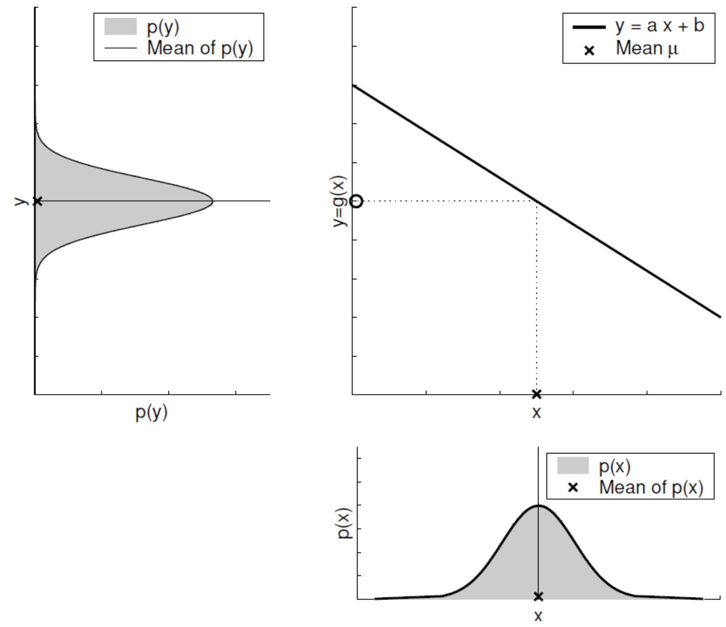


Courtesy: Wolfram Burgard

Linear Transformations

- Linear transformation of Gaussian distributions are Gaussian
- If $X \sim N(\mu_X, \Sigma_X)$ and $Y = AX + b$ then $Y \sim N(\mu_Y, \Sigma_Y)$ where
- $\mu_Y = A\,\mu_X + b$ and $\Sigma_Y = A\,\Sigma_X\,A^T$

- **Example:**
- $x_t = A_t\,x_{t-1} + B_t\,u_t + n_t$



Linear Transformations

- **Fact:**
  - Expectation is a linear operator of x
  - $E[X] = \int p(x)\, x\, dx$

$$
\begin{aligned}
\mu_Y &= E[Y] \\
&= E[AX + b] \\
&= A\, E[X] + b \\
&= A\, \mu_X + b
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_Y &= E[(Y - \mu_Y)(Y - \mu_Y)^T] \\
&= E[(AX + b - A\mu_X - b)(AX + b - A\mu_X - b)^T] \\
&= E[(A(X - \mu_X))(A(X - \mu_X))^T] \\
&= A\, E[(X - \mu_X)(X - \mu_X)^T]\, A^T \\
&= A\, \Sigma_{\mathrm{X}}\, A^T
\end{aligned}
$$

Independence

- Let $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ where X_1, X_2 are uncorrelated, i.e., the covariance is of the form $\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$ where $\Sigma_{12} = \Sigma_{21} = 0$
- Then X_1, X_2 are independent and $f_X(X) = f_{X_1}(X_1)\, f_{X_2}(X_2)$
- **Note:** The converse is always true, i.e., if two random variables are independent then they are uncorrelated
- **Example:** We assume that the noise is independent of the state of the system

Sum of Independent Gaussians

- Let X, Y be independent multivariate Gaussian random variables with mean μ_X, μ_Y and covariance Σ_X, Σ_Y
- The sum $Z = X + Y$ is also Gaussian with mean $\mu_Z = \mu_X + \mu_Y$ and covariance $\Sigma_Z = \Sigma_X + \Sigma_Y$

- **Example:**
  - $x_t = x_{t-1} + n_t$
  - $z_t = x_t + v_t$

Courtesy: Shaojie Shen

Jointly Normal Random Vectors

- Let X be a multivariate Gaussian random variable and let $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$
- Then X_1, X_2 are both (multivariate) Gaussian random variables and are jointly normally distributed

- **Note:** If X_1, X_2 are both (multivariate) Gaussian random variables then it does *not* necessarily imply that $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ is also Gaussian
- **Note:** If X_1, X_2 are independent (multivariate) Gaussian random variables then $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ is also Gaussian

Courtesy: Shaojie Shen

Conditional Distributions

- Let $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ be a multivariate Gaussian with mean $\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$ and covariance $\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$
- Then the conditional density $f_{X_1|X_2}(x_1|X_2 = x_2)$ is a multivariate normal distribution with
  - mean $\mu_{X_1|X_2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)$
  - covariance $\Sigma_{X_1|X_2} = \Sigma_{11} - \Sigma_{12}\,\Sigma_{22}^{-1}\,\Sigma_{21}$
- **Note:** $\Sigma_{X_1|X_2}$ is the Schur complement of Σ_{22} (Not used in L9)

- Further readings: http://fourier.eng.hmc.edu/e161/lectures/gaussianprocess/node7.html

Kalman Filter

System Modeling

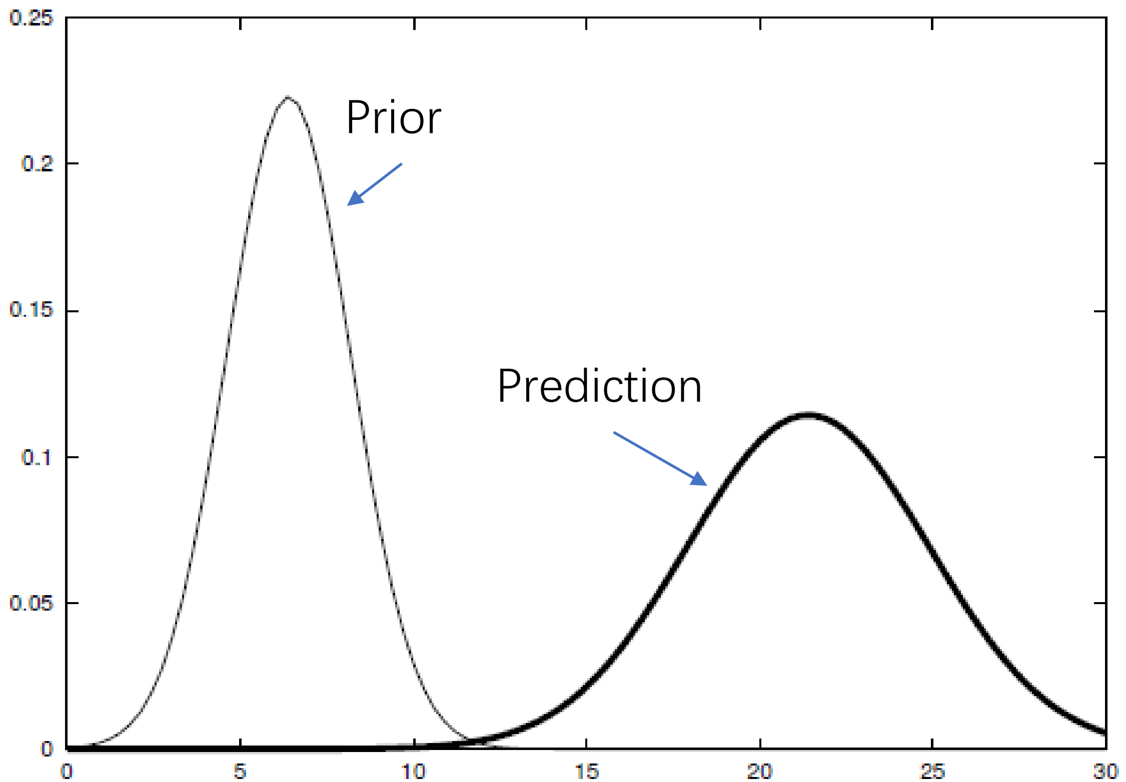
- The prior state of the robot is represented by a Gaussian distribution
  - $p(x_0) \sim N(\mu_0, \Sigma_0)$
- The process model $g(x_t \mid x_{t-1}, u_t)$ is linear with additive Gaussian white noise
  - $x_t = A_t\, x_{t-1} + B_t\, u_t + n_t$
  - $n_t \sim N(0, Q_t)$
- The measurement model $h(z_t \mid x_t)$ is linear with additive Gaussian white noise
  - $z_t = C_t\, x_t + v_t$
  - $v_t \sim N(0, R_t)$

Courtesy: Shaojie Shen

Kalman Filter – Prediction

- Bayes:

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



Courtesy: [Thrun, Burgard, and Fox. Probabilistic Robotics. 2005]

Kalman Filter – Prediction

- Bayes:

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

- $x_t = A_t\, x_{t-1} + B_t u_t + n_t$
- $n_t \sim N(0, Q_t)$
- Prior: $p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1}) \sim N(\mu_{t-1}, \Sigma_{t-1})$
- Prediction:
  - $\bar{\mu}_t = A\, \mu_{t-1} + B\, u_t$
  - $\bar{\Sigma}_t = A\, \Sigma_{t-1}\, A^T + Q$

Courtesy: Shaojie Shen

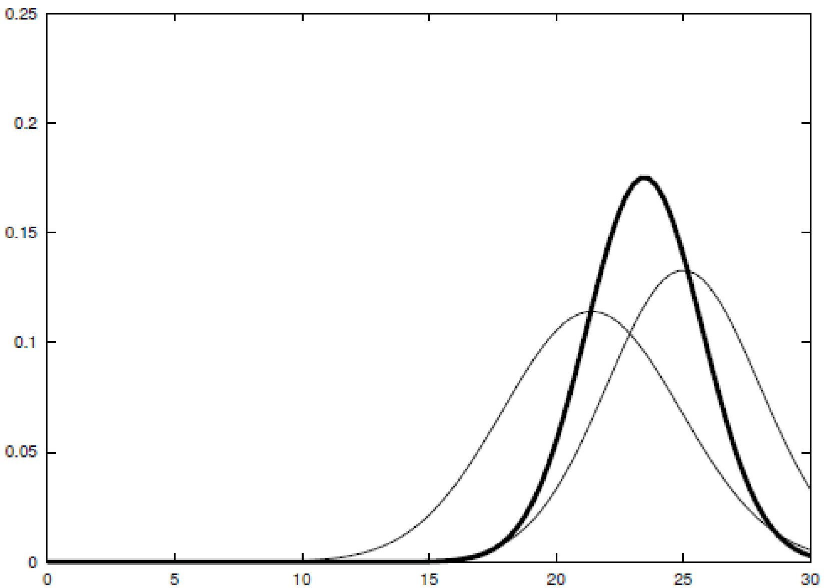
Kalman Filter - Update

- Bayes:
$$p(x_t \mid z_{1:t}, u_{1:t}) = \eta h(z_t \mid x_t) p(x_t \mid z_{1:t-1}, u_{1:t})$$
- The measurement model is $z_t = C_t \bar{x}_t + v_t, \quad v_t \sim N(0, R_t)$
- The best update without a measurement is to set $x_t = \bar{x}_t$
- $\begin{bmatrix} x_t \\ z_t \end{bmatrix} = \begin{bmatrix} I & 0 \\ C & I \end{bmatrix} \begin{bmatrix} \bar{x}_t \\ v_t \end{bmatrix}$
- **Question:** Is this a jointly normal distribution?
- $\mu = \begin{bmatrix} \bar{\mu}_t \\ C\bar{\mu}_t \end{bmatrix}$
- $\Sigma = \begin{bmatrix} I & 0 \\ C & I \end{bmatrix} \begin{bmatrix} \bar{\Sigma}_t & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} I & C^T \\ 0 & I \end{bmatrix} = \begin{bmatrix} \bar{\Sigma}_t & \bar{\Sigma}_t C^T \\ C\bar{\Sigma}_t & C\bar{\Sigma}_t C^T + R \end{bmatrix}$

Kalman Filter - Update

- The distribution of x_t conditioned on z_t is thus normal with
- $\mu_{x_t|z_t} = \bar{\mu}_t + \bar{\Sigma}_t C^T (C\bar{\Sigma}_t C^T + R)^{-1} (z_t - C\bar{\mu}_t)$
- $\Sigma_{x_t|z_t} = \bar{\Sigma}_t - \bar{\Sigma}_t C^T (C\bar{\Sigma}_t C^T + R)^{-1} C\bar{\Sigma}_t$

- Define the Kalman gain K_t
- $K_t = \bar{\Sigma}_t C^T (C\bar{\Sigma}_t C^T + R)^{-1}$
- $\mu_t = \bar{\mu}_t + K_t (z_t - C\bar{\mu}_t)$
- $\Sigma_t = \bar{\Sigma}_t - K_t C\bar{\Sigma}_t$



Courtesy: Shaojie Shen and *Probabilistic Robotics*

Kalman Gain

- $K_t = \bar{\Sigma}_t \, C^T \, (C \, \bar{\Sigma}_t \, C^T + R)^{-1}$
- **Intuition:** How much to trust the sensor vs. the prediction
- **Example:**
  - Perfect sensor $R = 0$
    - $K_t = \bar{\Sigma}_t \, C^T \, (C \, \bar{\Sigma}_t \, C^T + R)^{-1} = C^{-1}$
    - $\mu_t = \bar{\mu}_t + K_t(z_t - C \, \bar{\mu}_t) = C^{-1} z_t$
    - $\Sigma_t = \bar{\Sigma}_t - K_t \, C \, \bar{\Sigma}_t = 0$
  - Horrible sensor $R \to \infty$
    - $K_t = \bar{\Sigma}_t \, C^T \, (C \, \bar{\Sigma}_t \, C^T + R)^{-1} \to 0$
    - $\mu_t = \bar{\mu}_t + K_t(z_t - C \, \bar{\mu}_t) \to \bar{\mu}_t$
    - $\Sigma_t = \bar{\Sigma}_t - K_t \, C \, \bar{\Sigma}_t \to \bar{\Sigma}_t$

Courtesy: Shaojie Shen

Kalman Filter

- Prior:
  - $p(x_0) \sim N(\mu_0, \Sigma_0)$
- Process model:
  - $x_t = A_t\ x_{t-1} + B_t\ u_t + n_t$
  - $n_t \sim N(0, Q_t)$
- Measurement model:
  - $z_t = C_t\ x_t + v_t$
  - $v_t \sim N(0, R_t)$

- Prior:
  - μ_{t-1}, Σ_{t-1}
- Prediction:
  - $\bar{\mu}_t = A_t\ \mu_{t-1} + B_t\ u_t$
  - $\bar{\Sigma}_t = A_t\ \Sigma_{t-1}\ A_t^T + Q_t$
- Update:
  - $K_t = \bar{\Sigma}_t\ C_t^T\ (C_t\ \bar{\Sigma}_t\ C_t^T + R_t)^{-1}$
  - $\mu_t = \bar{\mu}_t + K_t(z_t - C_t\ \bar{\mu}_t)$
  - $\Sigma_t = \bar{\Sigma}_t - K_t\ C_t\ \bar{\Sigma}_t$

Courtesy: Shaojie Shen

Kalman Filter

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

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

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

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

9. Return μ_t, Σ_t

Courtesy: Wolfram Burgard

Example Problem

$$x_t = x_{t-1} + u_t + n_t$$

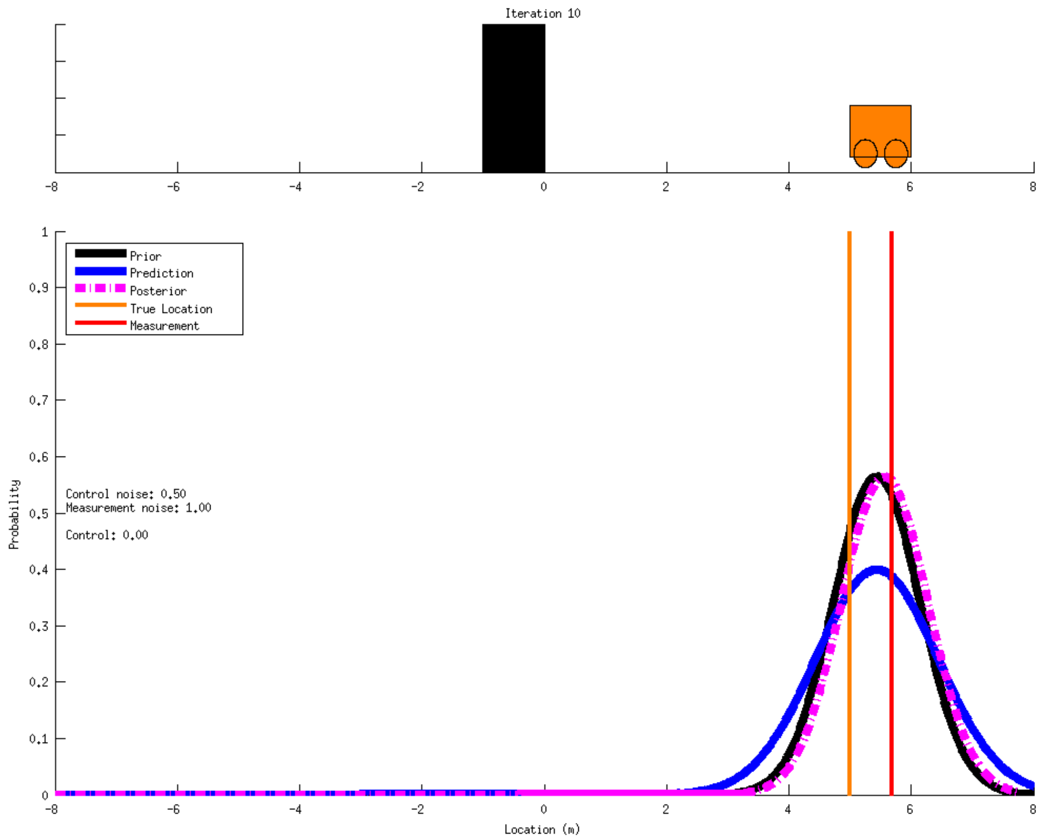
$$Q_t = 0.5$$

$$A_t = B_t = 1$$

$$z_t = x_t + v_t$$

$$R_t = 1.0$$

$$C_t = 1$$

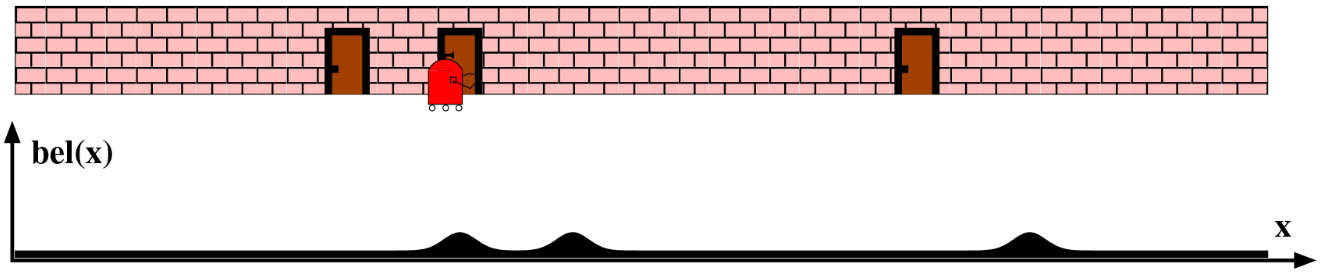


Kalman Filter Facts

- Everything stays Gaussian
- The variance never increases due to receiving a measurement
- Prediction and update can happen in arbitrary order as long as they are temporally sorted
- If the distribution is not Gaussian, the Kalman filter is the minimum variance linear estimator (Kalman Filter can also work)

Summary

- Kalman filter is a weighted mean with Gaussians
- **Pros:**
  - Simple
  - Purely matrix operations
    - Computationally efficient, even for high dimensional systems
- **Cons:**
  - Assumes everything is linear and Gaussian
  - Unimodal distribution
    - Cannot handle multiple hypotheses (Particle Filter Can)



Problem of Kalman Filter

- We live in a nonlinear world
- Most robotic sytems are nonlinear

$$\cancel{x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t} \qquad \cancel{z_t = C_t x_t + \delta_t}$$

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

$$x_t = g(u_t, x_{t-1}) \qquad\qquad z_t = h(x_t)$$

Extended (to handle nonlinear systems) Kalman Filter

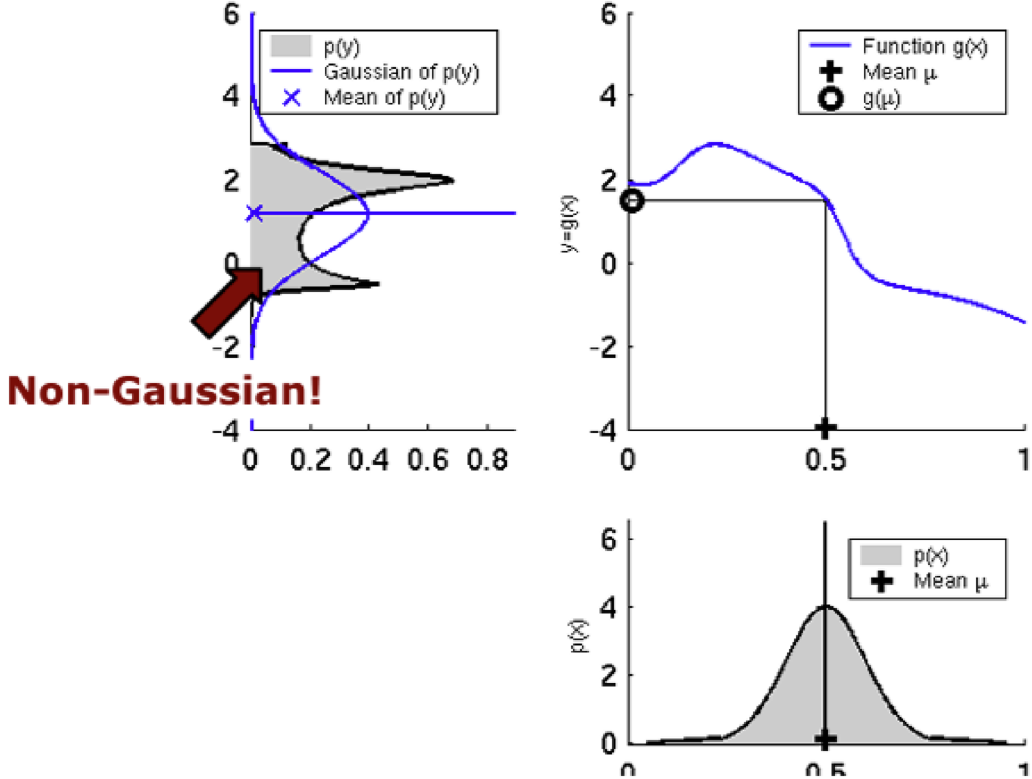
Assumptions for EKF

- The prior state of the robot is represented by a Gaussian distribution
  - $p(x_0) \sim N(\mu_0, \Sigma_0)$
- The process model is:
  - Nonlinear $x_t = g(x_{t-1}, u_t, n_t)$
  - $n_t \sim N(0, Q_t)$ is Gaussian white noise
- The measurement model is:
  - Nonlinear $z_t = h(x_t, v_t)$
  - $v_t \sim N(0, R_t)$ is Gaussian white noise
- $x_t = \bar{x}_t$ in some pages and literatures

(careful when reading resources)

Non-Linear functions

- The non-linear functions lead to nonGaussian distributions
- Kalman filter is not applicable anymore
- **What can be done to resolve this? Local linearization**



Courtesy: Wolfram Burgard

Linearization: Taylor Expansion

- First Order:

$$f(x) \approx f(a) + \left.\frac{\partial f(x)}{\partial x}\right|_a (x - a)$$

- Prediction:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} (x_{t-1} - \mu_{t-1})$$

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t (x_{t-1} - \mu_{t-1})$$

- Update:

$$h(x_t) \approx h(\bar{\mu}_t) + \frac{\partial h(\bar{\mu}_t)}{\partial x_t} (x_t - \bar{\mu}_t)$$

$$h(x_t) \approx h(\bar{\mu}_t) + H_t (x_t - \bar{\mu}_t)$$

Jacobians

Courtesy: Wolfram Burgard

Math - Jacobian Matrix

- It is a non-square matrix $\mathrm{m} \times n$ in general
- Differs for different points in times
- Given a vector-valued function

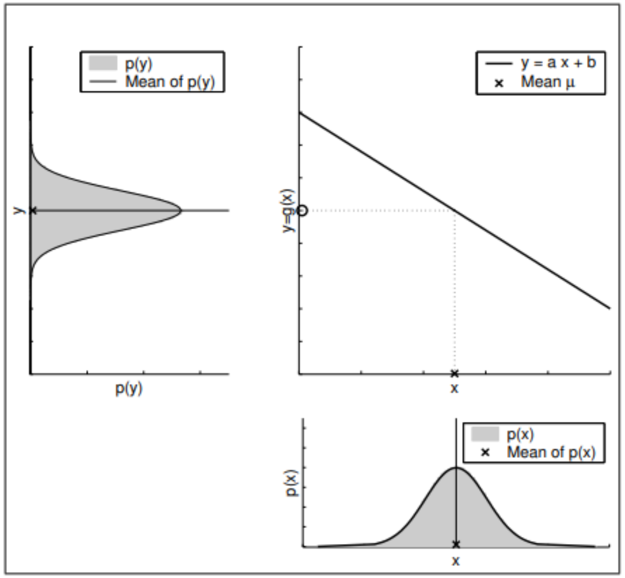
$$g(x) = \begin{pmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_m(x) \end{pmatrix}$$

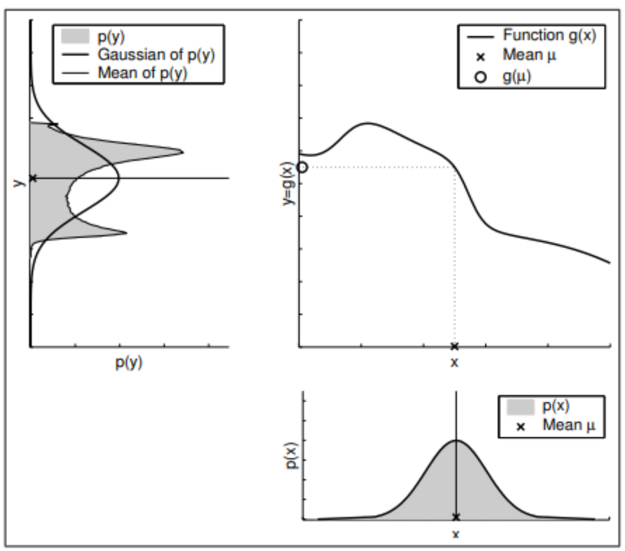
- The Jacobian matrix is defined as

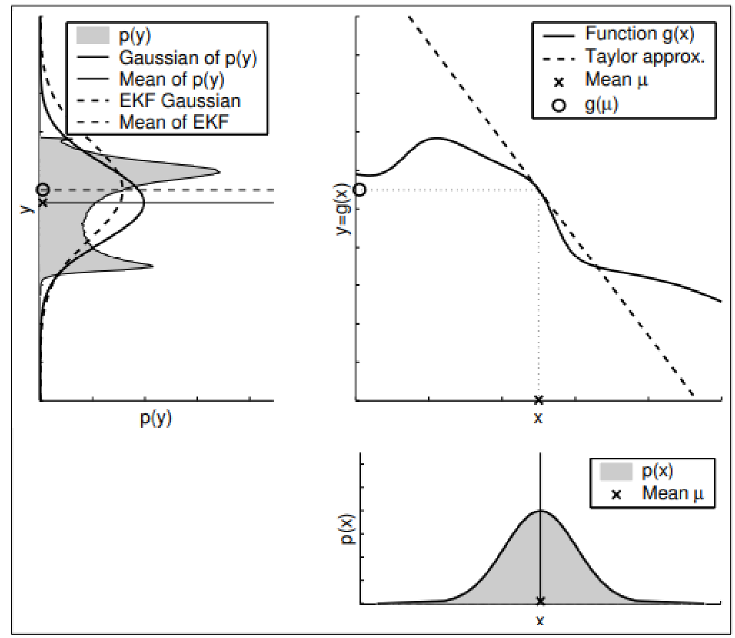
$$G_x = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \frac{\partial g_m}{\partial x_2} & \cdots & \frac{\partial g_m}{\partial x_n} \end{pmatrix}$$

Courtesy: Cyrill Stachniss

Gaussian Approximation







EKF aims to generate Gaussian Approximation of the random variable under nonlinear function

Courtesy: Shaojie Shen

Linearized Models

- G_t replace A_t in Kalman Filter

$$x_t = g\left(u_t, x_{t-1}\right) \approx g\left(u_t, \mu_{t-1}\right) + G_t\left(x_{t-1} - \mu_{t-1}\right)$$

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

- H_t replace C_t in Kalman Filter

$$z_t = h\left(x_t\right) \approx h\left(\bar{\mu}_t\right) + H_t\left(x_t - \bar{\mu}_t\right)$$

$$z_t = C_t\, x_t + v_t$$

Extended Kalman Filter

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

2. Prediction:
3. $\bar{\mu}_t = g(u_t, \mu_{t-1})$ $\longleftarrow$ $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
4. $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + Q_t$ $\longleftarrow$ $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t$

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

9. Return μ_t, Σ_t

$$H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t} \qquad G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}$$

Courtesy: Wolfram Burgard

Math. Derivation of KF and KF

- Probabilistic Robotics Chapter 3.2.4 and Chapter 3.3.3

3.2.4 Mathematical Derivation of the KF

This section derives the Kalman filter algorithm in Table 3.1. The section can safely be skipped at first reading.

Part 1: Prediction. Our derivation begins with Lines 2 and 3 of the algorithm, in which the belief $\overline{bel}(x_t)$ is calculated from the belief one time step earlier, $bel(x_{t-1})$. Lines 2 and 3 implement the update step described in Equation (2.61), restated here for the reader's convenience:

$$\overline{bel}(x_t) \;=\; \int \underbrace{p(x_t \mid x_{t-1}, u_t)}_{\sim\mathcal{N}(x_t;A_t x_{t-1}+B_t u_t,R_t)} \quad \underbrace{bel(x_{t-1})}_{\sim\mathcal{N}(x_{t-1};\mu_{t-1},\Sigma_{t-1})} \quad dx_{t-1} \tag{3.7}$$

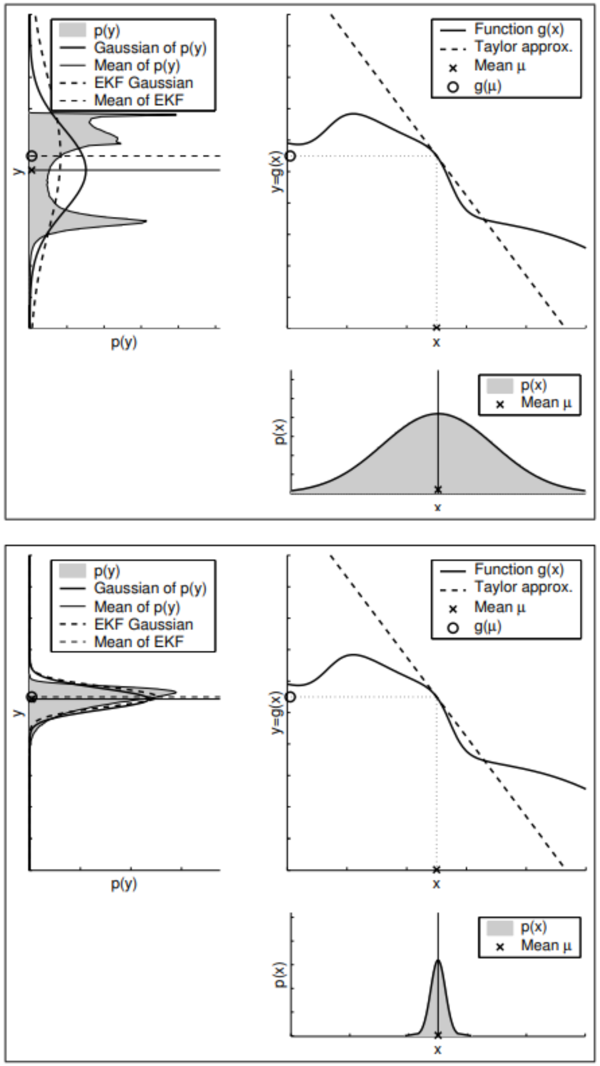
The "prior" belief $bel(x_{t-1})$ is represented by the mean μ_{t-1} and the covariance Σ_{t-1}. The state transition probability $p(x_t \mid x_{t-1}, u_t)$ was given in (3.4) as a normal distribution over x_t with mean $A_t x_{t-1} + B_t u_t$ and covariance R_t. As we shall show now, the outcome of (3.7) is again a Gaussian with mean $\bar{\mu}_t$ and covariance $\bar{\Sigma}_t$ as stated in Table 3.1.

We begin by writing (3.7) in its Gaussian form:

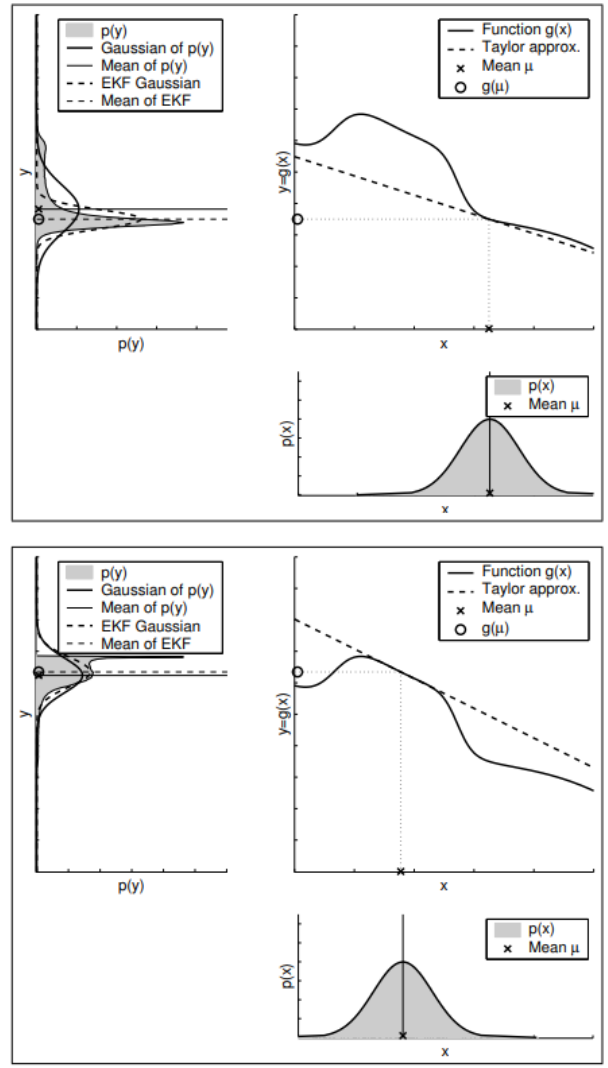
$$\begin{aligned}\overline{bel}(x_t) \;=\; \eta \int & \exp\left\{-\tfrac{1}{2}\,(x_t - A_t\, x_{t-1} - B_t\, u_t)^T\, R_t^{-1}\,(x_t - A_t\, x_{t-1} - B_t\, u_t)\right\} \\ & \exp\left\{-\tfrac{1}{2}\,(x_{t-1} - \mu_{t-1})^T \Sigma_{t-1}^{-1}(x_{t-1} - \mu_{t-1})\right\}\; dx_{t-1}\,. \end{aligned} \tag{3.8}$$

Courtesy: *Probabilistic Robotics*

More on Linearization



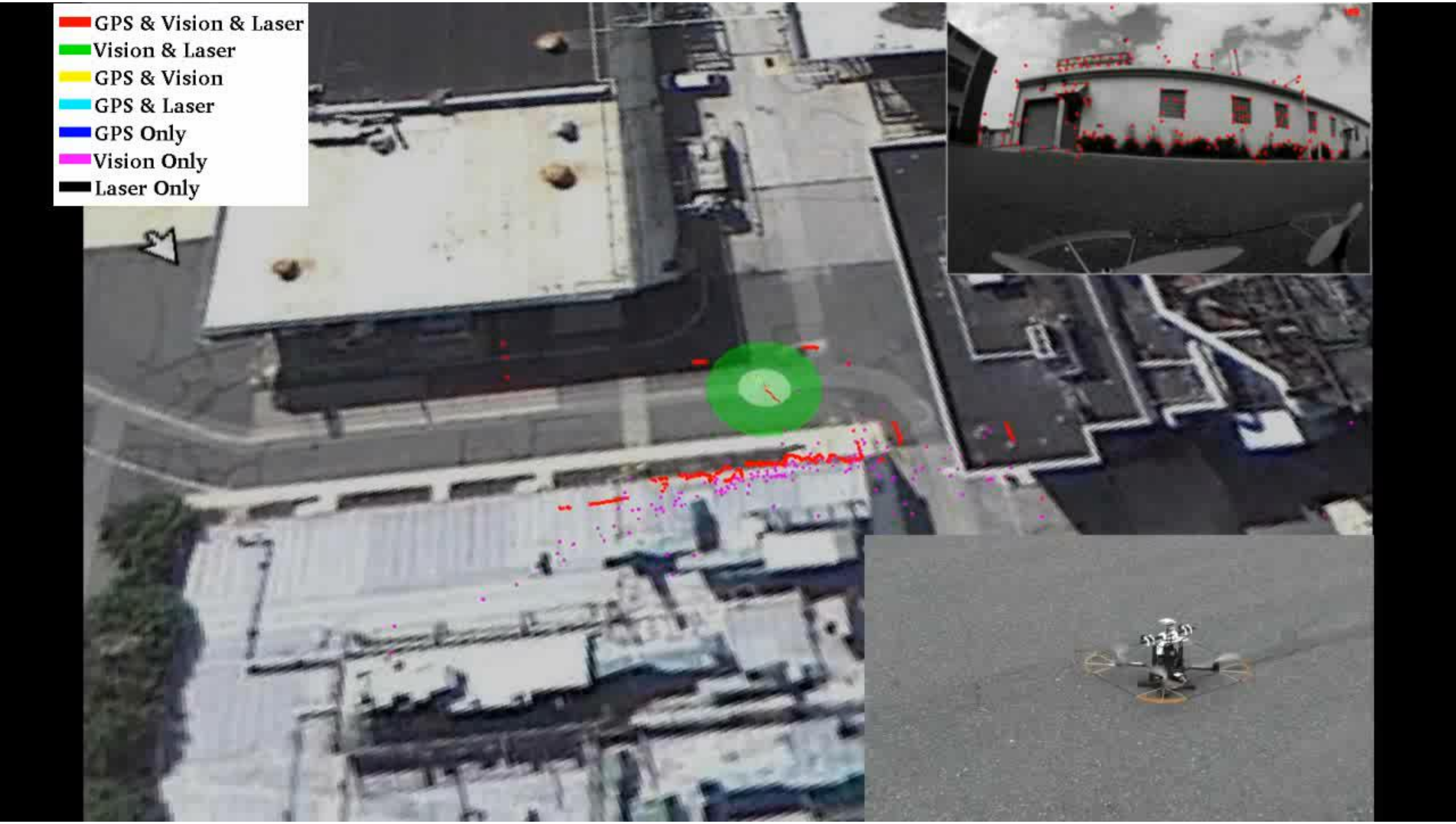
Different uncertainties of the random variable



Different nonlinearities of the function

Courtesy: Shaojie Shen

Multi-Sensor Fusion



Courtesy: Shaojie Shen

Our work with KF

- Yin, Huan, Runjian Chen, Yue Wang, and Rong Xiong. "Rall: end-to-end radar localization on lidar map using differentiable measurement model." 2021

RaLL: End-to-end Radar Localization on Lidar Map Using Differentiable Measurement Model

Huan Yin, *IEEE Member*, Yue Wang, *IEEE Member*, Runjian Chen, and Rong Xiong, *IEEE Member*

Zhejiang University

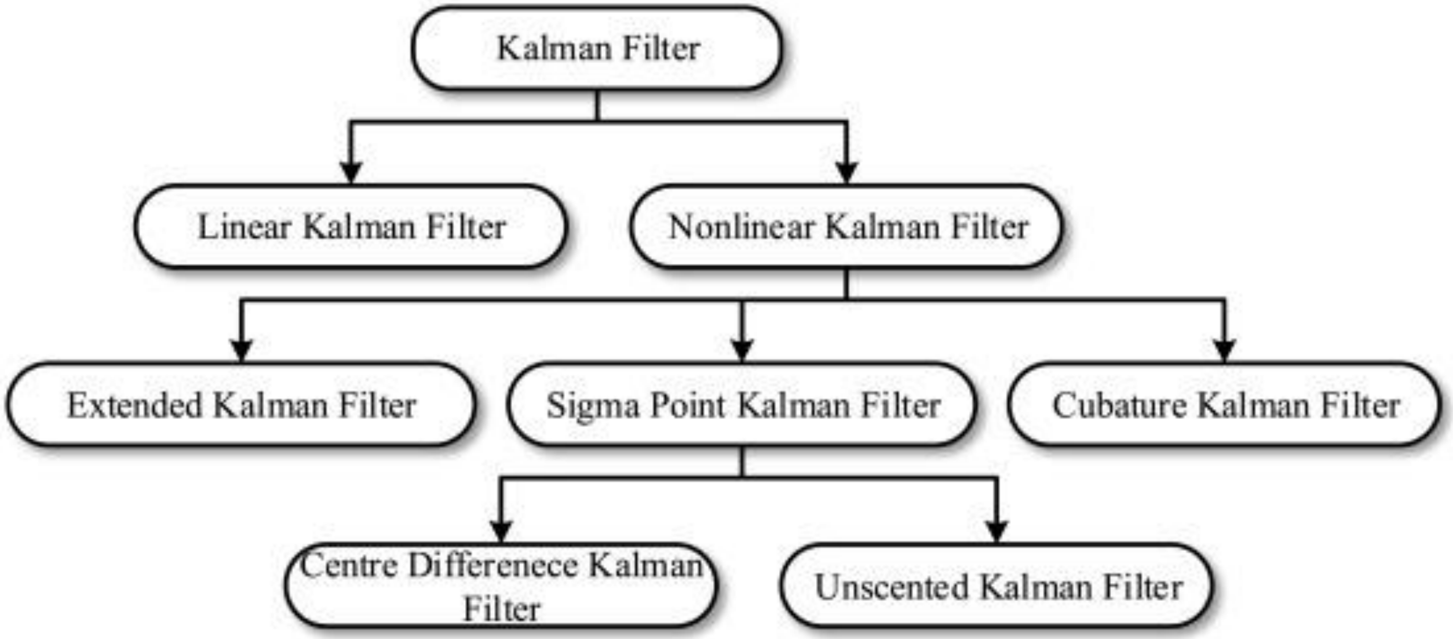
Extended KF Summary

- Extension of the Kalman Filter
- One way to handle the non-linearities
- Perform Local linearizations via Taylor Expansions
- Work well in moderate non-linearities

Courtesy: Cyrill Stachniss

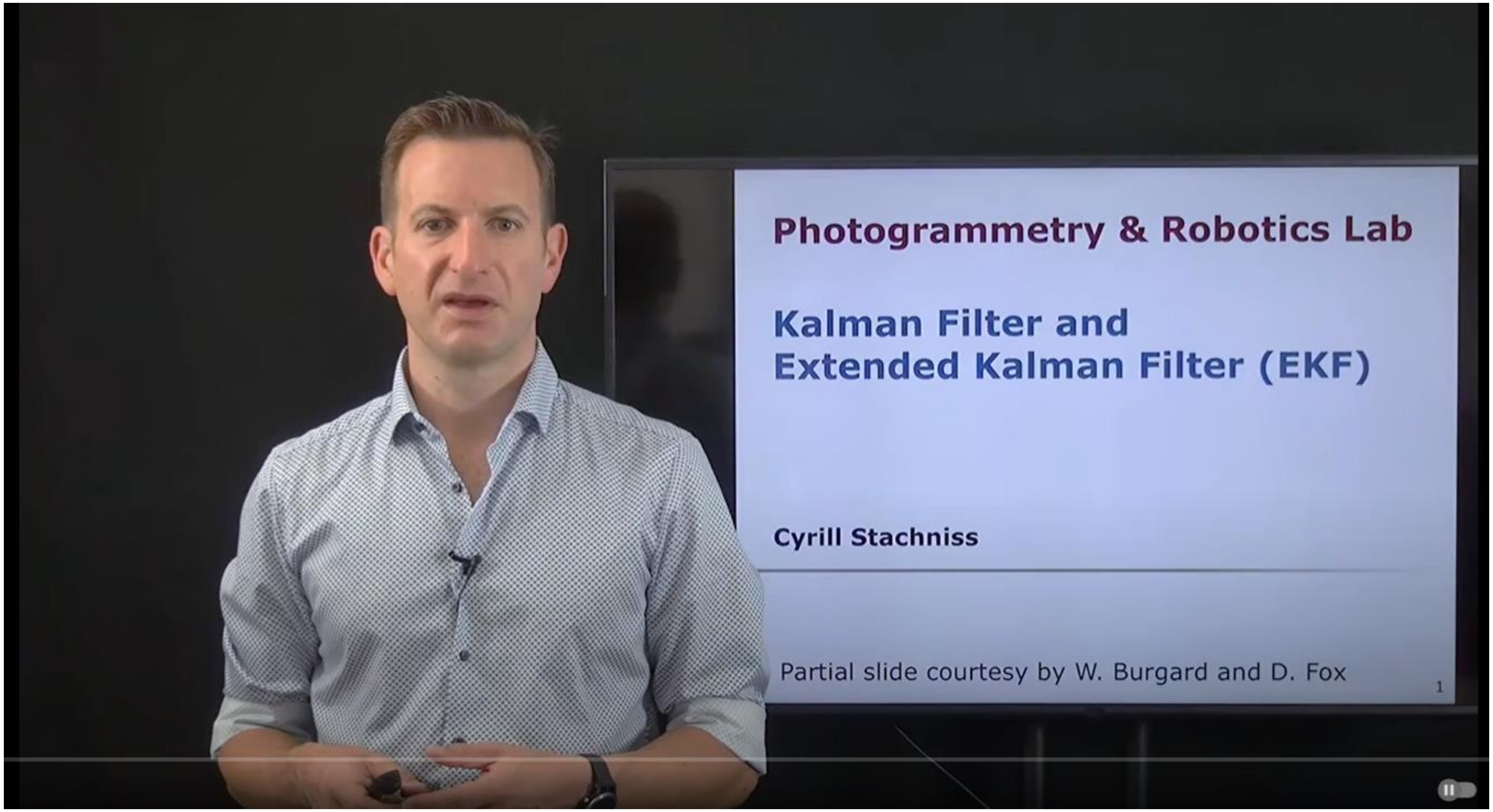
What if EKF could not work well?

- Other family members
  - Unscented Kalman Filter
  - Invariant Kalman Filter
  - etc.



Resources

- *Probabilistic Robotics* Chapter 3 and 7
- Prof. Cyrill Stachniss



Next Lecture

- EKF SLAM
  - Project 2, a classical case of EKF
  - Lecture 3 - SLAM
  - Lecture 6 - Global Feature Map