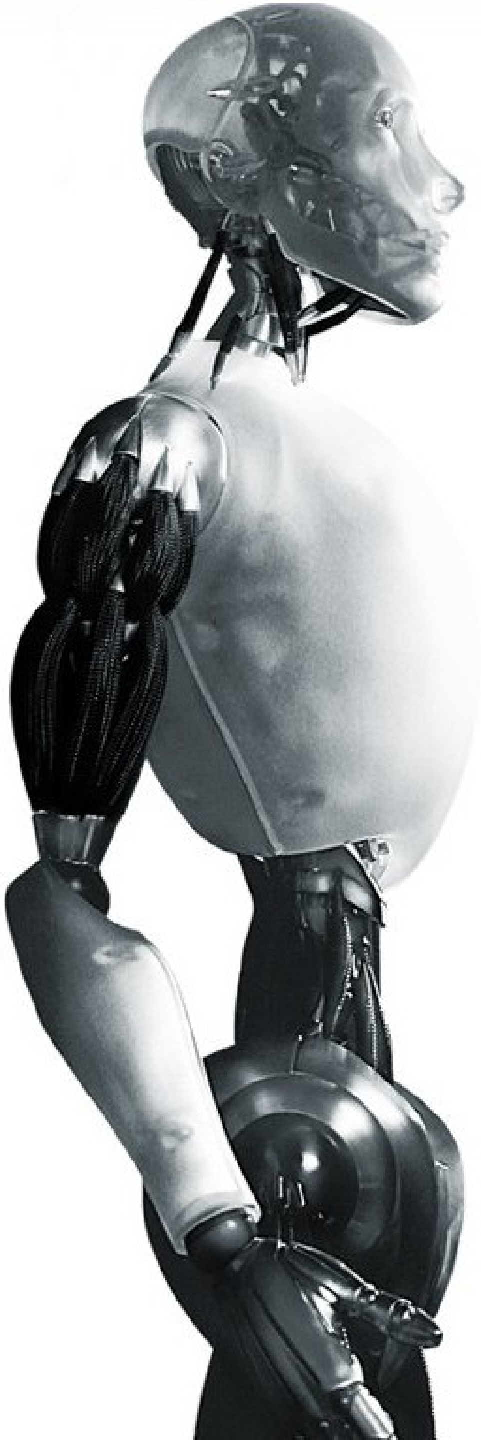
Disclaimer

These slides are intended as presentation aids for the lecture. They contain information that would otherwise be to difficult or time-consuming to reproduce on the board. But they are incomplete, not self-explanatory, and are not always used in the order they appear in this presentation. As a result, these slides should not be used as a script for this course. I recommend you take notes during class, maybe on the slides themselves. It has been shown that taking notes improves learning success.



Robotics

Recursive State Estimation

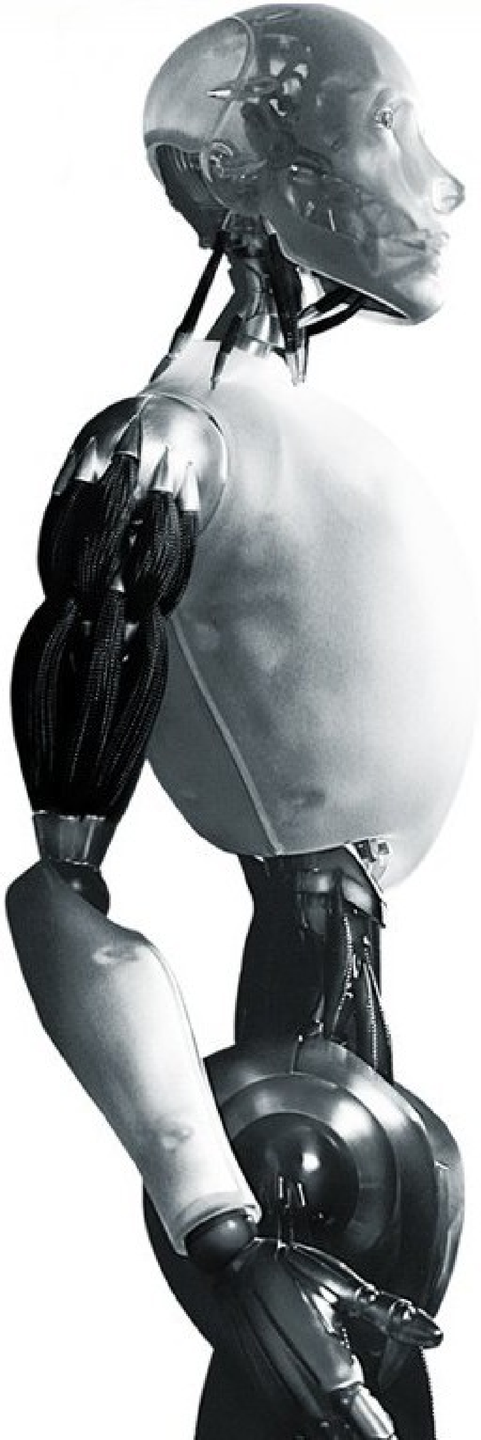
TU Berlin

Oliver Brock

Reading for this set of slides

- Probabilistic Robotics
  - Chapters 1-4, 7, 8-10 (please match the level of detail from the lectures, not all the material in these chapters is required)

Please note that this set of slides is intended as support for the lecture, not as a stand-alone script. If you want to study for this course, please use these slides in conjunction with the indicated chapters in the text books. The textbooks are available online or in the TUB library (many copies that can be checked out for the entire semester. There are also some aspects of the lectures that will not be covered in the text books but can still be part of the homework or exam. For those It is important that you attend class or ask somebody about what was covered in class.

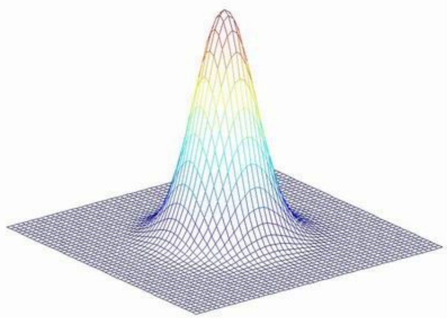


Robotics

Gaussian Bayes Filters

TU Berlin
Oliver Brock

Multivariate Gaussian



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

$$p(x) = \frac{1}{(2\pi)^{n/2}\det(\Sigma)^{1/2}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

Covariance Matrix Σ

$$\Sigma_{ij} = \mathrm{cov}(X_i, X_j) = \mathrm{E}\big[(X_i - \mu_i)(X_j - \mu_j)\big]$$

$$\Sigma = \begin{bmatrix} \mathrm{E}[(X_1 - \mu_1)(X_1 - \mu_1)] & \mathrm{E}[(X_1 - \mu_1)(X_2 - \mu_2)] & \cdots & \mathrm{E}[(X_1 - \mu_1)(X_n - \mu_n)] \\ \mathrm{E}[(X_2 - \mu_2)(X_1 - \mu_1)] & \mathrm{E}[(X_2 - \mu_2)(X_2 - \mu_2)] & \cdots & \mathrm{E}[(X_2 - \mu_2)(X_n - \mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n - \mu_n)(X_1 - \mu_1)] & \mathrm{E}[(X_n - \mu_n)(X_2 - \mu_2)] & \cdots & \mathrm{E}[(X_n - \mu_n)(X_n - \mu_n)] \end{bmatrix}.$$

Gaussian Bayesian Filters

- Kalman Filter
  - linear update of states based on action
    $$p(x_t | a_{t-1}, s_{t-1})$$
  - linear sensor model
    $$p(o_t | s_t)$$
  - belief can be described by a normal distribution
  - computationally efficient and elegant

Kalman Filter

- [Swerling 1958] [Kalman 1960]
- Assumptions:
  - State transition probability $p(x_t|u_t,x_{t-1})$ is a linear function with Gaussian noise

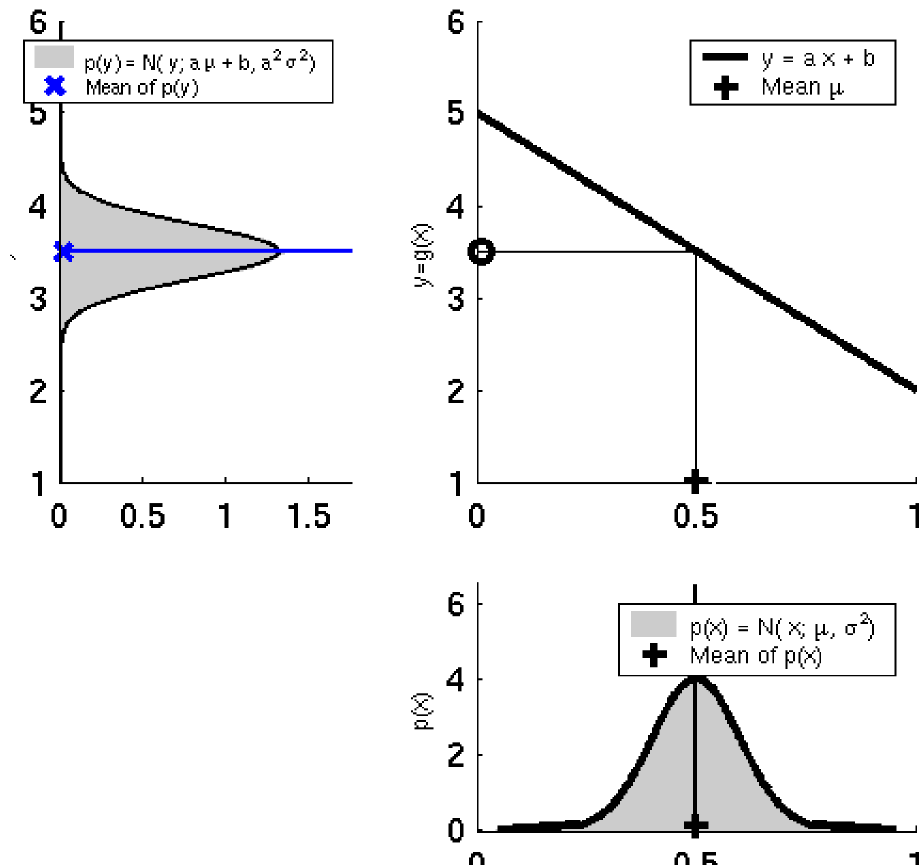
$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

  - Measurement probability $p(z_t|x_t)$ is linear with Gaussian noise

$$z_t = C_t x_t + \delta_t$$

  - Initial belief is normally distributed

Linear Transformation of a Gaussian



Falling Body Example (state equation)

$$y_t = y_0 + \dot{y}_0 t - \frac{g}{2} t^2$$

$$y_t = y_{t-1} + \dot{y}_{t-1} \Delta t - \frac{g}{2} \Delta t^2$$

$$x_t = \begin{bmatrix} y_t \\ \dot{y}_t \end{bmatrix}$$

$$\begin{aligned} x_t &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_{t-1} + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} (-g) \\ &= Ax + Bu + \varepsilon_t \end{aligned}$$

Falling Body Example (measurement)

$$
\begin{aligned}
z_t &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} x_t + \delta_t \\
&= C_t x_t + \delta_t
\end{aligned}
$$

Components of a Kalman Filter

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

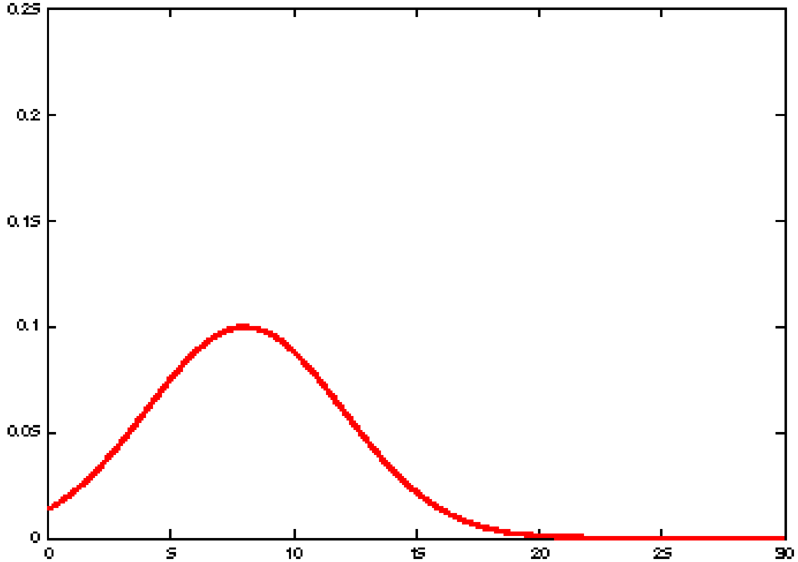
B_t Matrix (nxl) that describes how the control u_t changes the state from t to t-1.

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

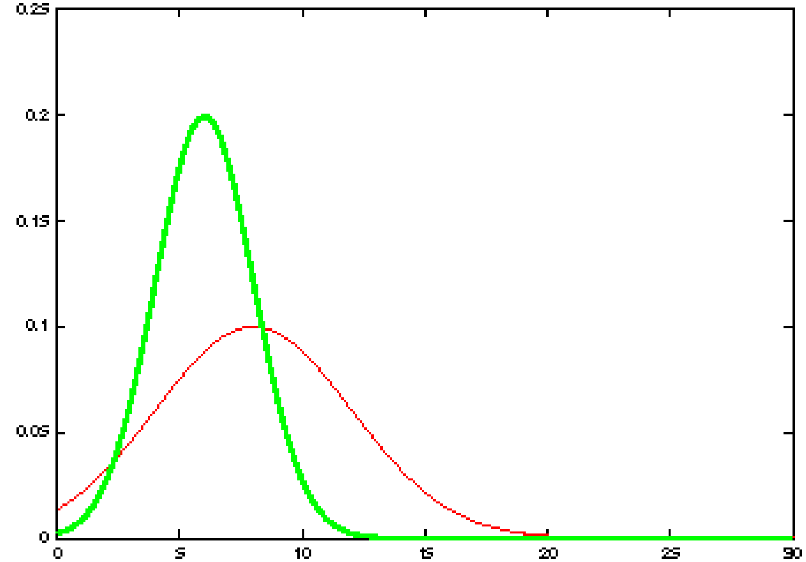
ε_t

δ_t

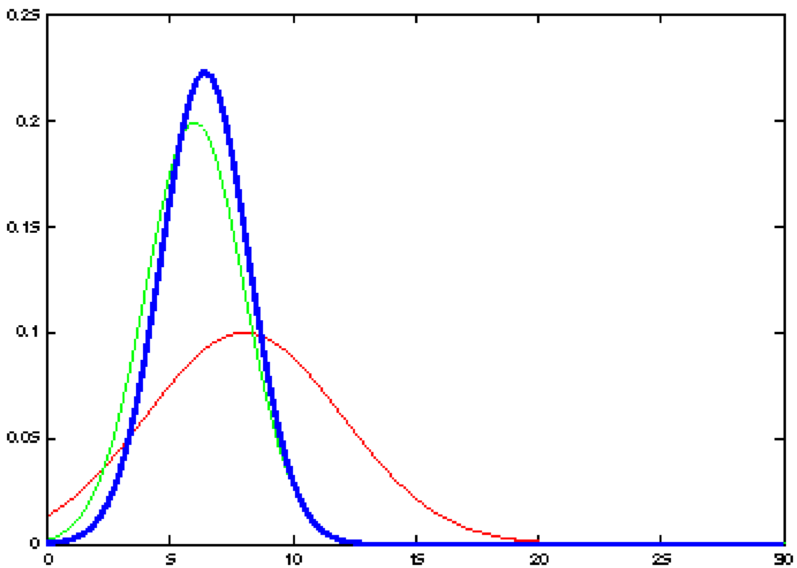
Random variables representing the process and measurement noise that are assumed to be independent and normally distributed with covariance R_t and Q_t respectively.



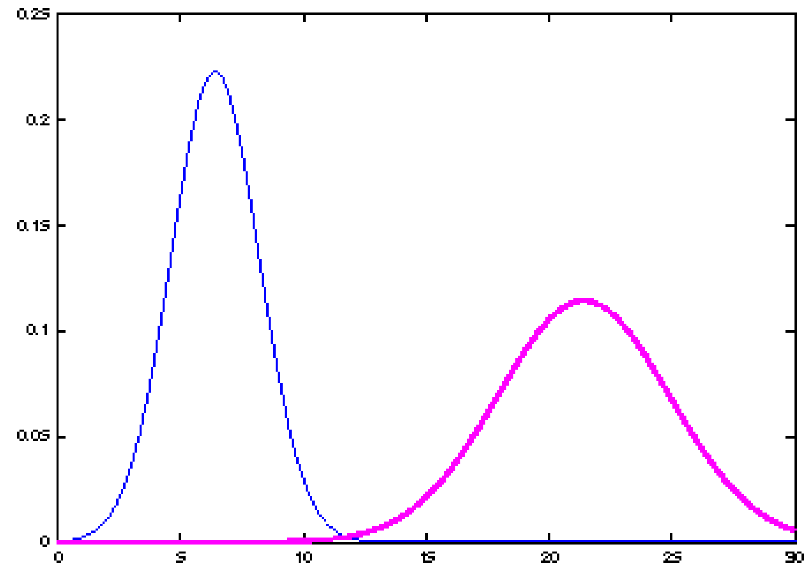
Initial estimate



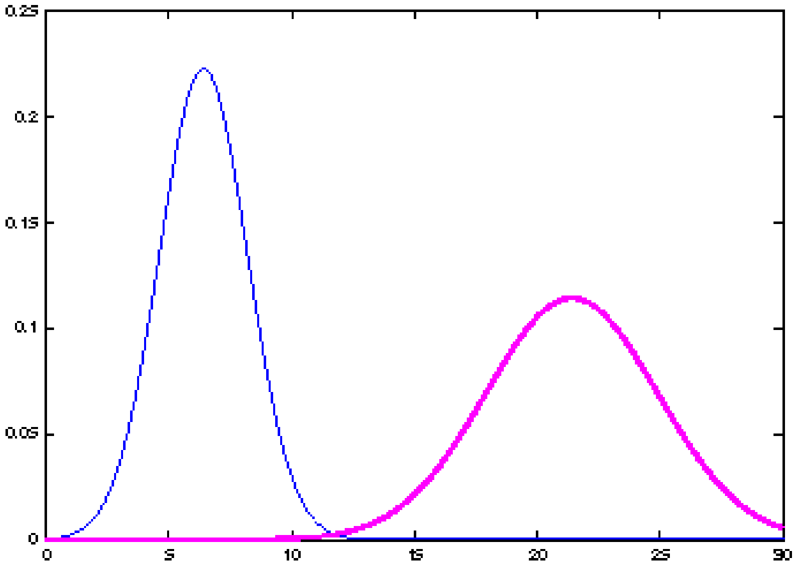
Sensing



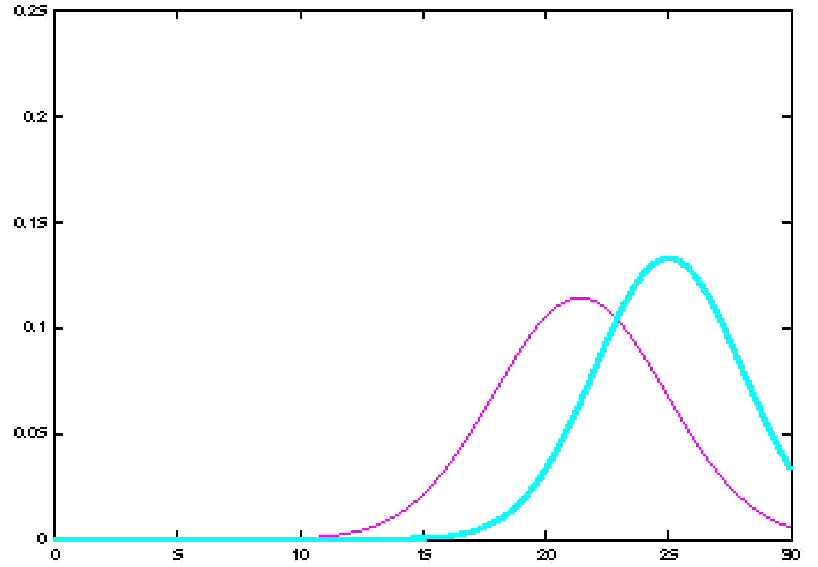
Correction



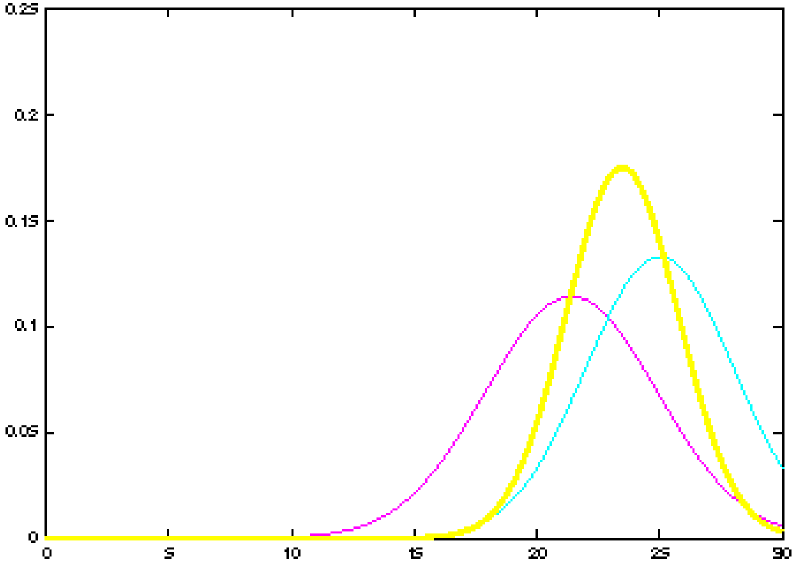
Prediction



Prediction



Sensing



Correction

Kalman Filter (KF)

Kalman Filter $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$

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

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

Prediction

Kalman gain

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

Innovation

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

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

Correction

return μ_t, Σ_t

Kalman Filter Cycle

Kalman Filter $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$

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

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

Prediction

Kalman gain $$K_t = \bar{\Sigma}_t \, C_t^T \, (C_t \, \bar{\Sigma}_t \, C_t^T + Q_t)^{-1}$$

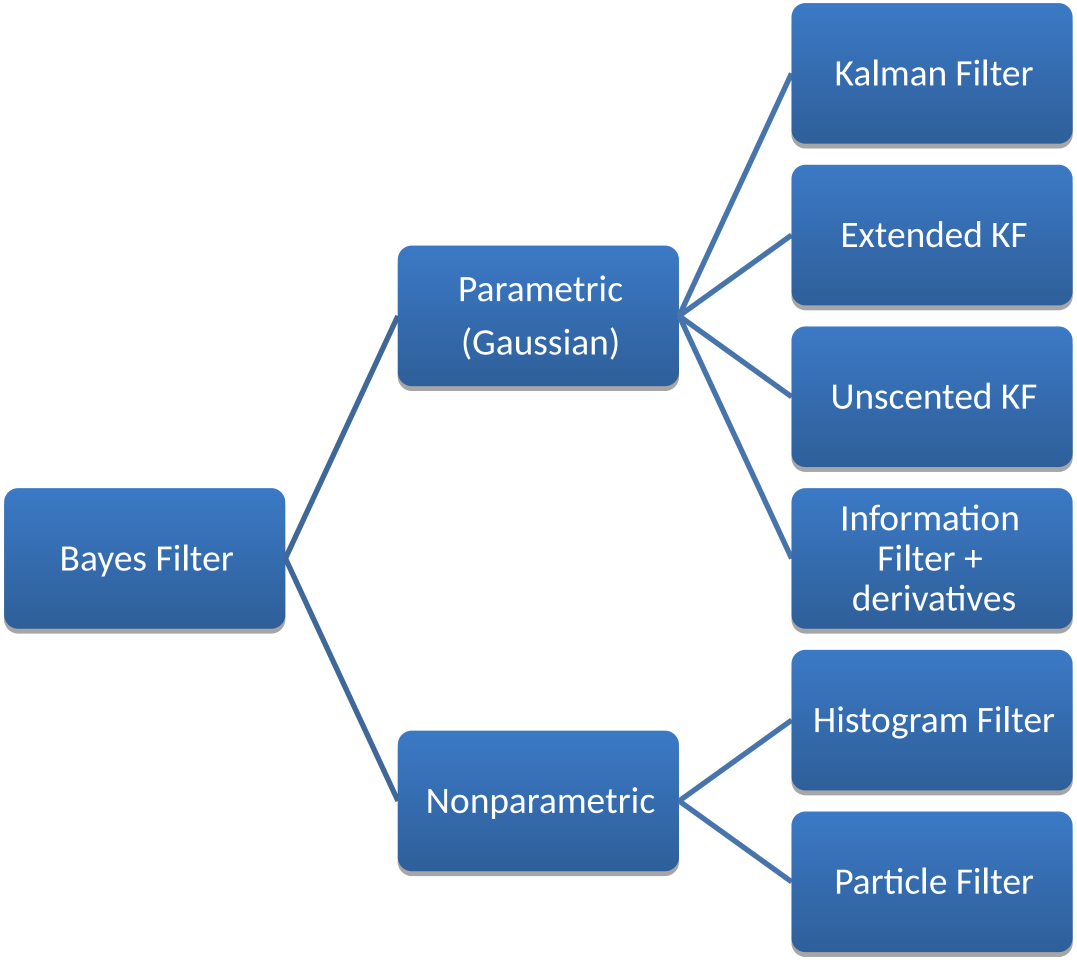
Innovation $$\mu_t = \bar{\mu}_t + K_t \, (z_t - C_t \, \bar{\mu}_t)$$

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

Correction

return μ_t, Σ_t

A family of methods



Extended Kalman Filter

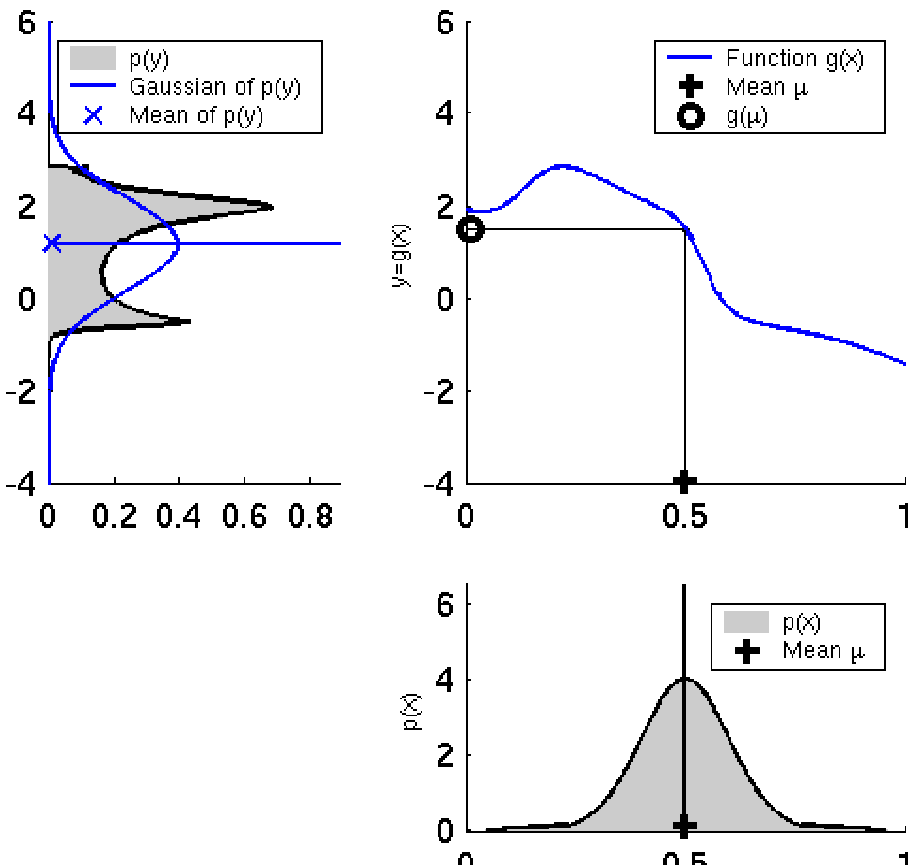
- Assumptions relaxed:
  - State transition probability $p(x_t|u_t,x_{t-1})$ is a **nonlinear** function with Gaussian noise

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t$$

  - Measurement probability $p(z_t|x_t)$ is a **nonlinear** with Gaussian noise

$$z_t = h(x_t) + \delta_t$$

  - Initial belief is normally distributed


p(y)
Gaussian of p(y)
Mean of p(y)
Function g(x)
Mean μ
g(μ)
y=g(x)
p(x)
Mean μ

Linearization with Taylor Expansion

$$
\begin{aligned}
g(u_t, x_{t-1}) \quad &\approx \quad g(u_t, \mu_{t-1}) + g'(u_t, \mu_{t-1})(x_{t-1} - \mu_{t-1}) \\
&= \quad g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1})
\end{aligned}
$$

G_t is the Jacobian!

Extended Kalman Filter (EKF)

Extended Kalman Filter $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$

$$\bar{\mu}_t = g(u_t, \mu_{t-1})$$

$$\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, F_t^T + R_t$$

Prediction

Kalman gain

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

Innovation

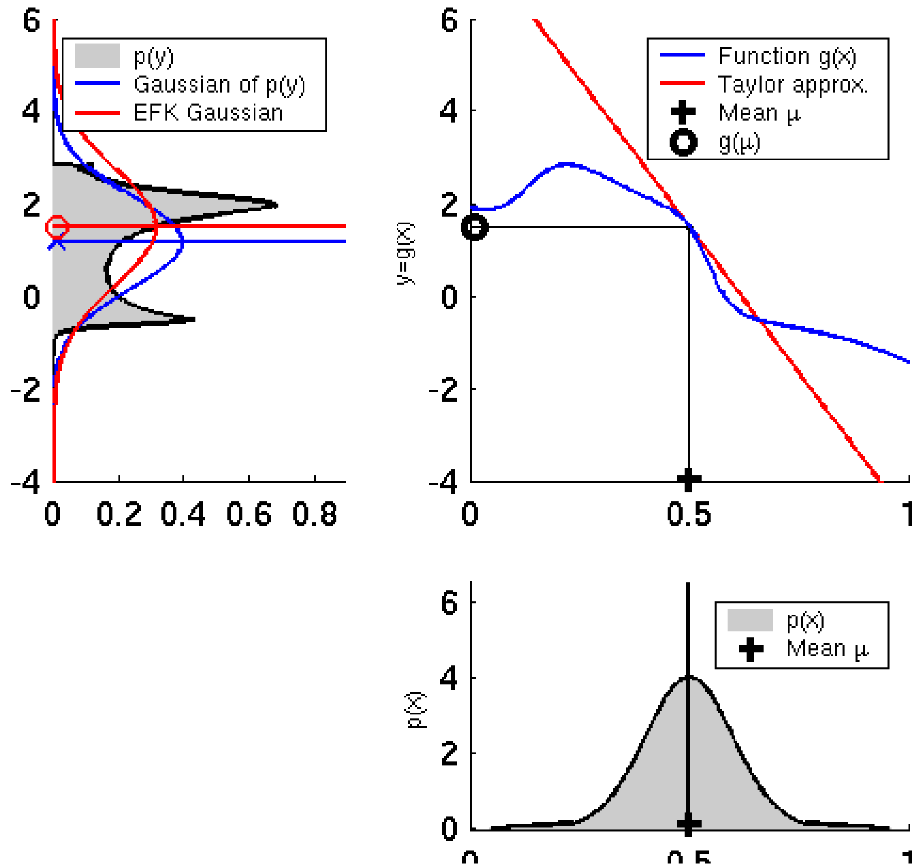
$$\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))$$

$$\Sigma_t = (I - K_t \, H_t) \bar{\Sigma}_t$$

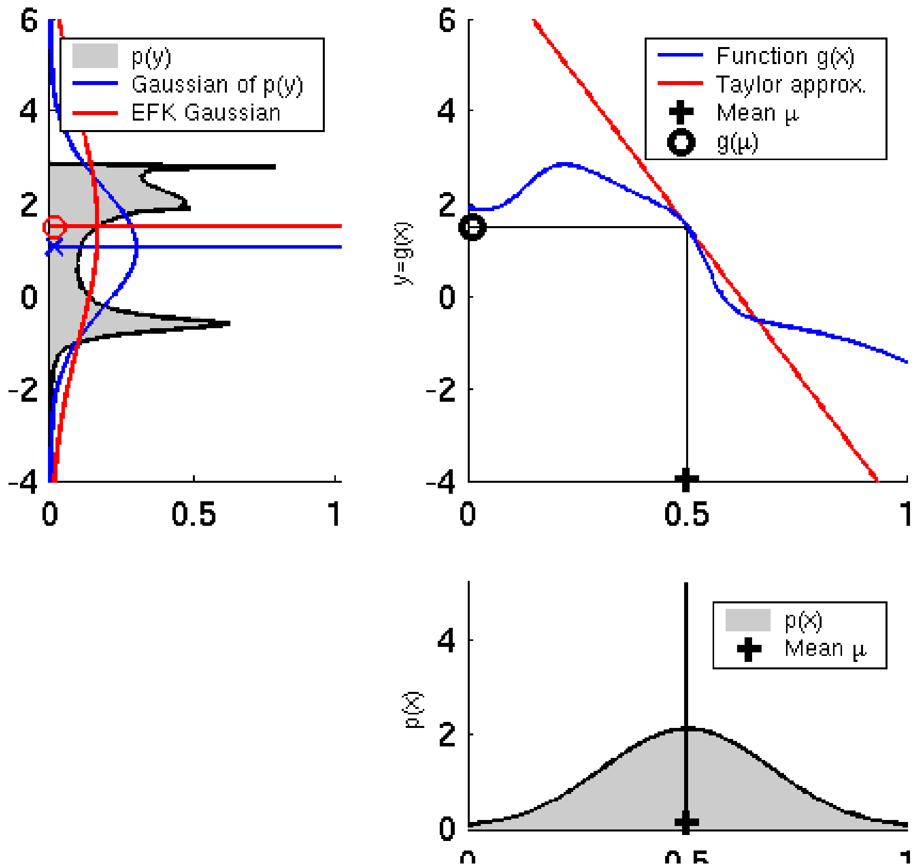
Correction

return μ_t, Σ_t

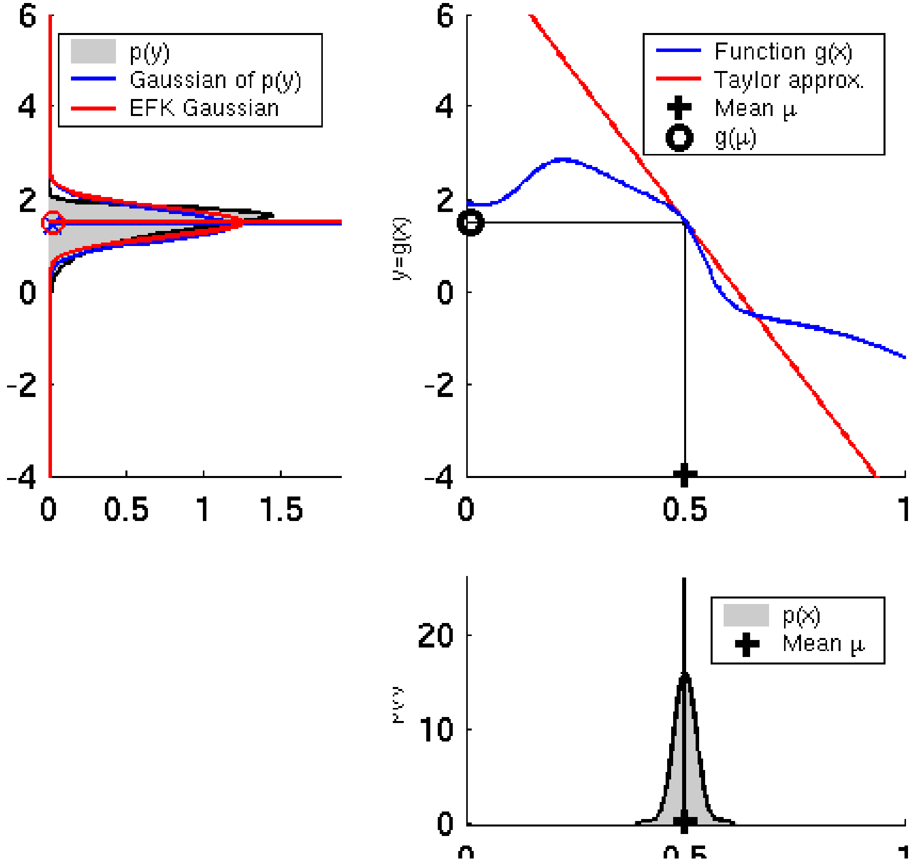
EKF depends on uncertainty (1)



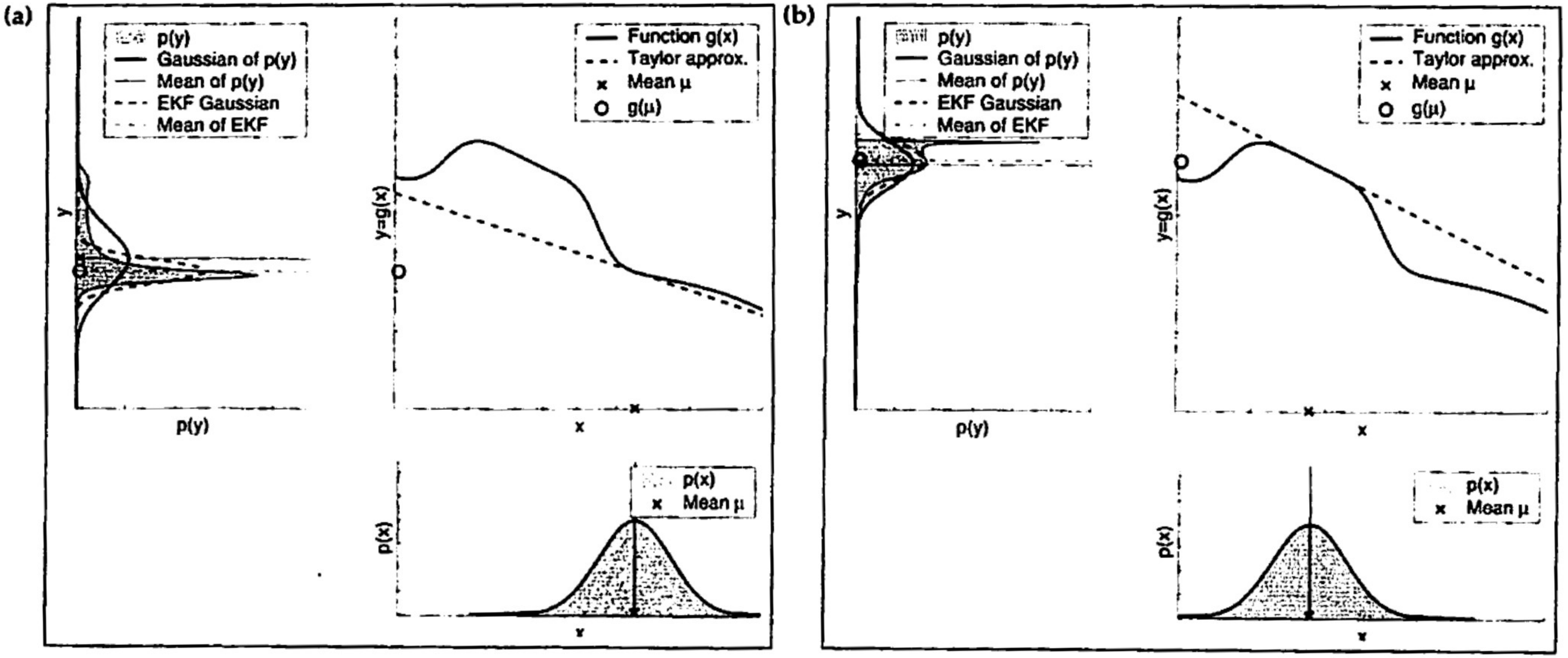
EKF depends on uncertainty (2)



EKF depends on uncertainty (3)



EKF depends on quality of approximation



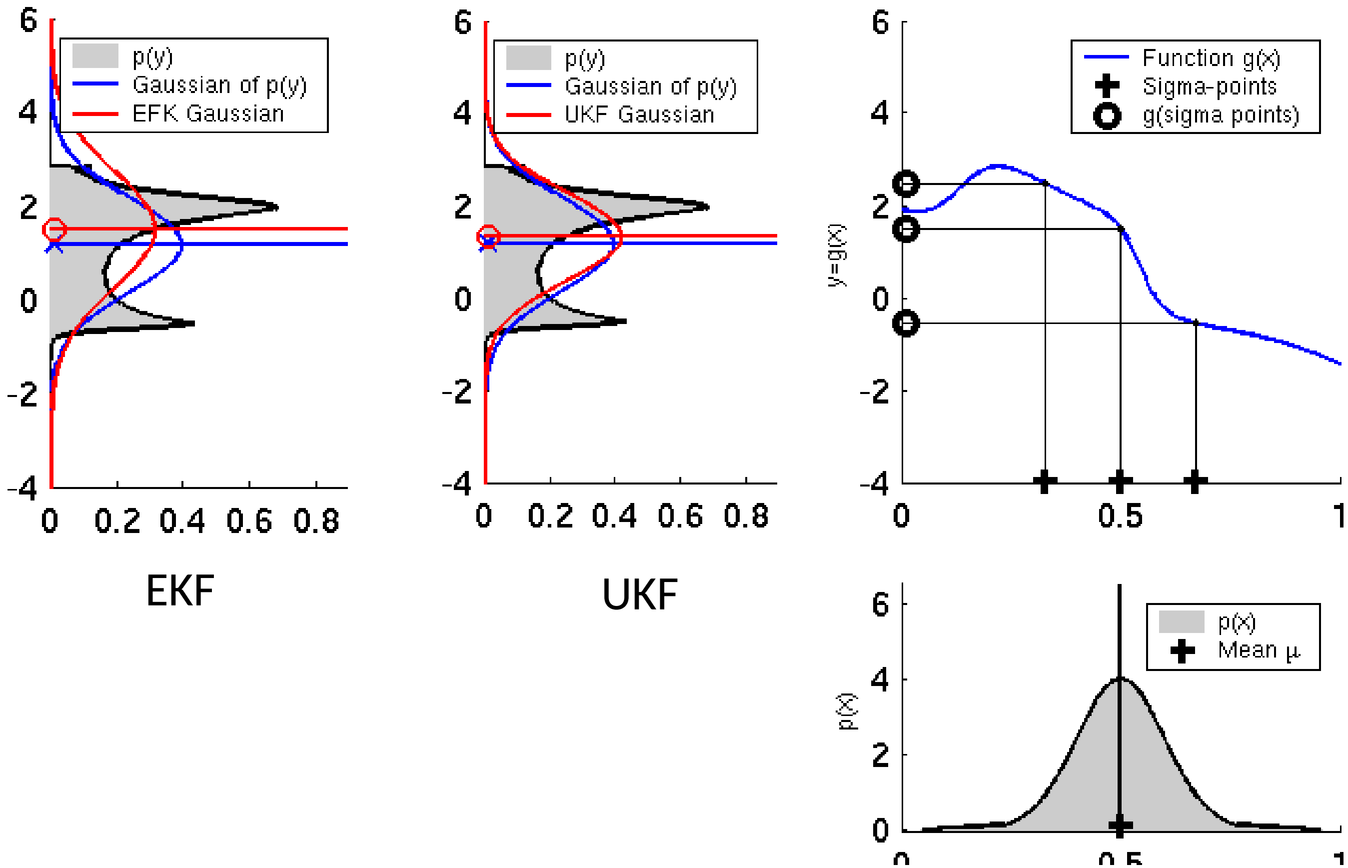
EKF Summary

- Highly efficient: Polynomial in measurement dimensionality *k* and state dimensionality *n*:

  $O(k^{2.376} + n^2)$

- Not optimal!
- Can diverge if nonlinearities are large!
- Works surprisingly well even when all assumptions are violated!

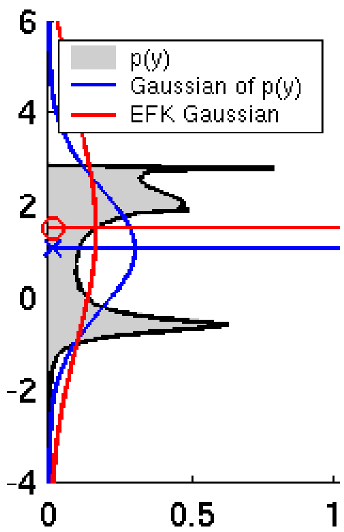
Unscented Kalman Filter (UKF)



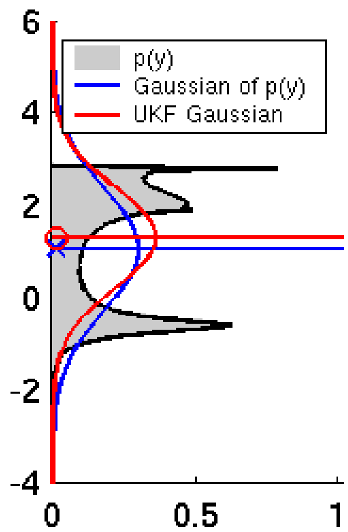
EKF

UKF

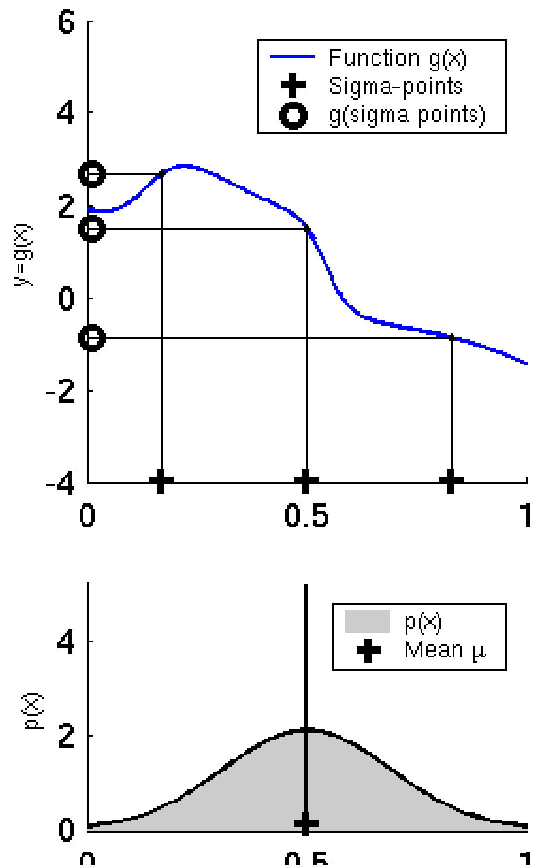
UKF Sigma-Point Estimate (2)



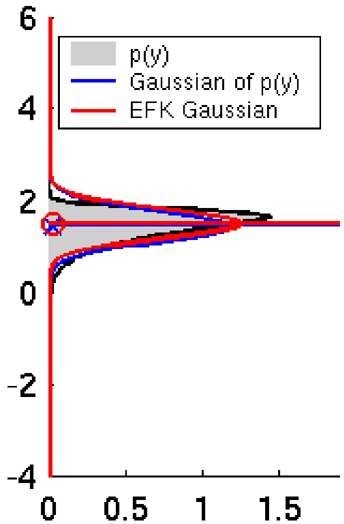
EKF



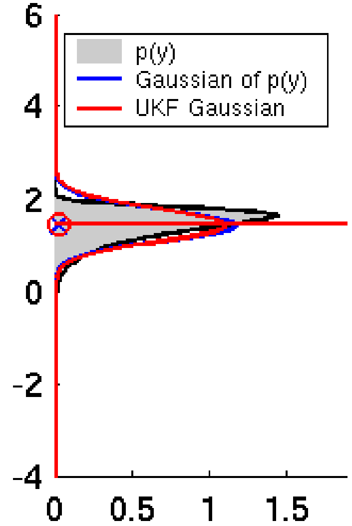
UKF



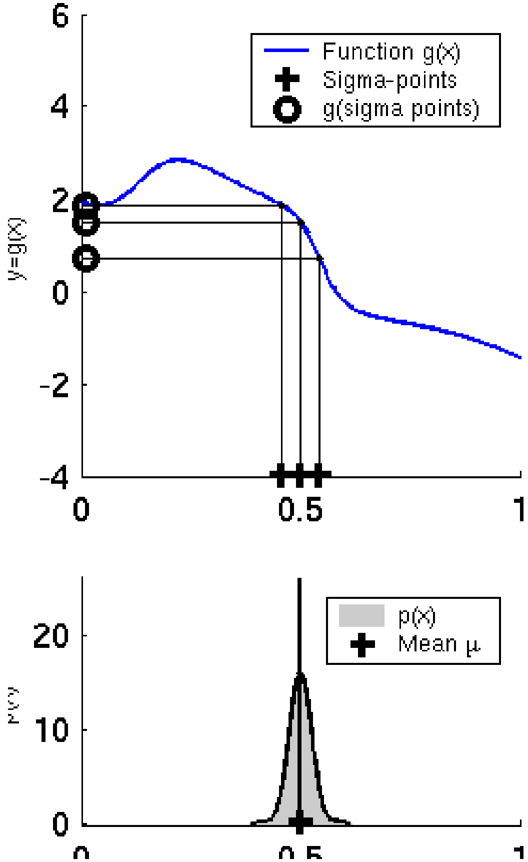
UKF Sigma-Point Estimate (3)



EKF



UKF



Unscented Transform

Sigma points

$$\chi^0 = \mu$$

$$\chi^i = \mu \pm \left(\sqrt{(n+\lambda)\Sigma}\right)_i$$

Weights

$$w_m^0 = \frac{\lambda}{n+\lambda} \qquad w_c^0 = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta)$$

$$w_m^i = w_c^i = \frac{1}{2(n+\lambda)} \qquad \text{for } i = 1,\ldots,2n$$

Pass sigma points through nonlinear function

$$\psi^i = g(\chi^i)$$

Recover mean and covariance

$$\mu' = \sum_{i=0}^{2n} w_m^i \psi^i$$

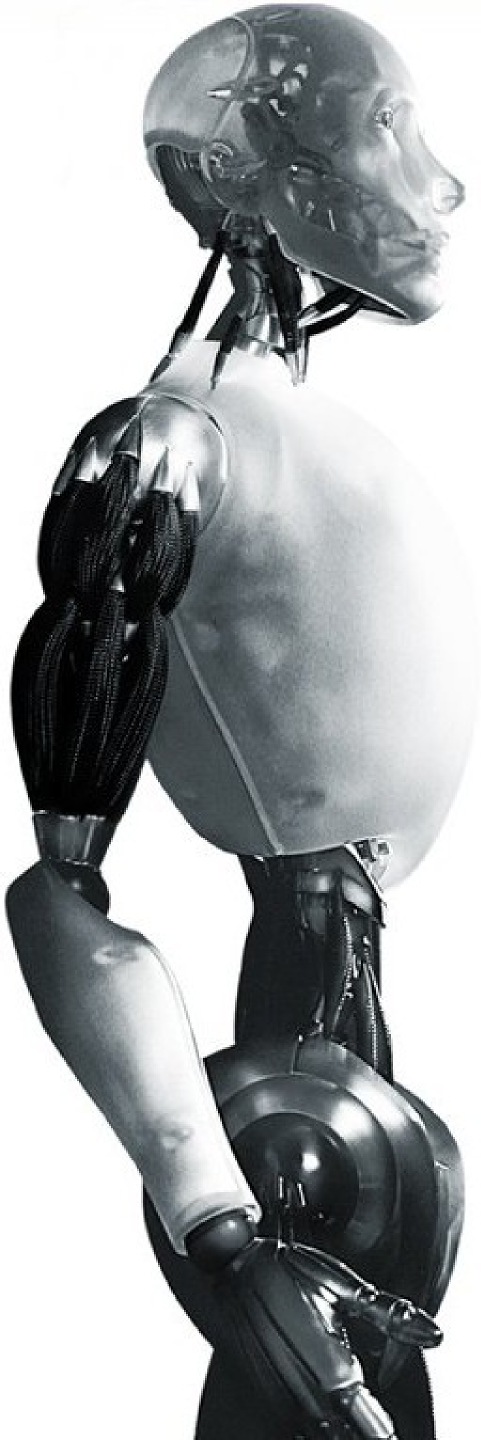
$$\Sigma' = \sum_{i=0}^{2n} w_c^i (\psi^i - \mu)(\psi^i - \mu)^T$$

UKF Summary

- Highly efficient: Same complexity as EKF, with a constant factor slower in typical practical applications
- Better linearization than EKF: Accurate in first two terms of Taylor expansion (EKF only first term)
- Derivative-free: No Jacobians needed
- Still not optimal!

Information Filter

- Dual of KF
- Also Extended Information Filter



Robotics

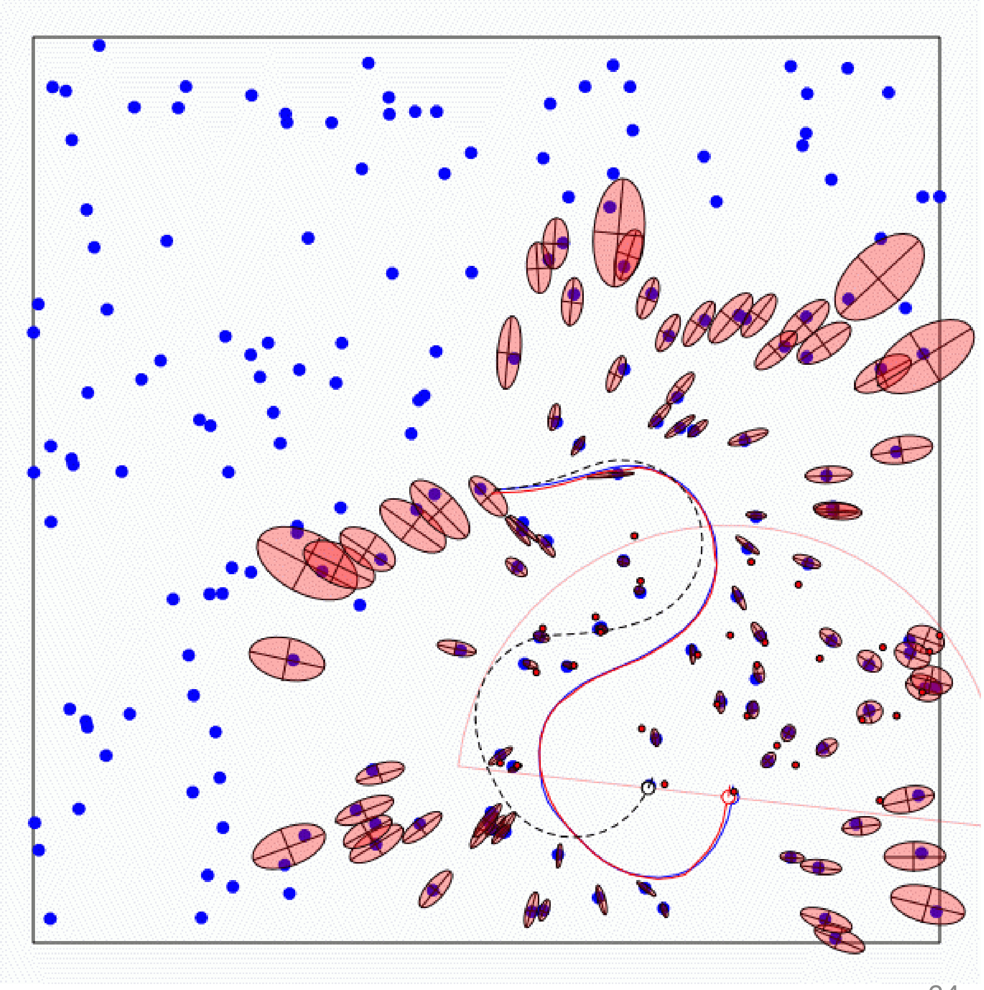
Simultaneous Mapping and Localization (SLAM)

TU Berlin

Oliver Brock

The SLAM Problem

A robot moving though an unknown, static environment



Given:

- The robot's controls
- Observations of nearby features

Estimate:

- Map of features
- Path of the robot

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

SLAM:
Simultaneous Localization and Mapping

- Full SLAM:

Estimates entire path and map!

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

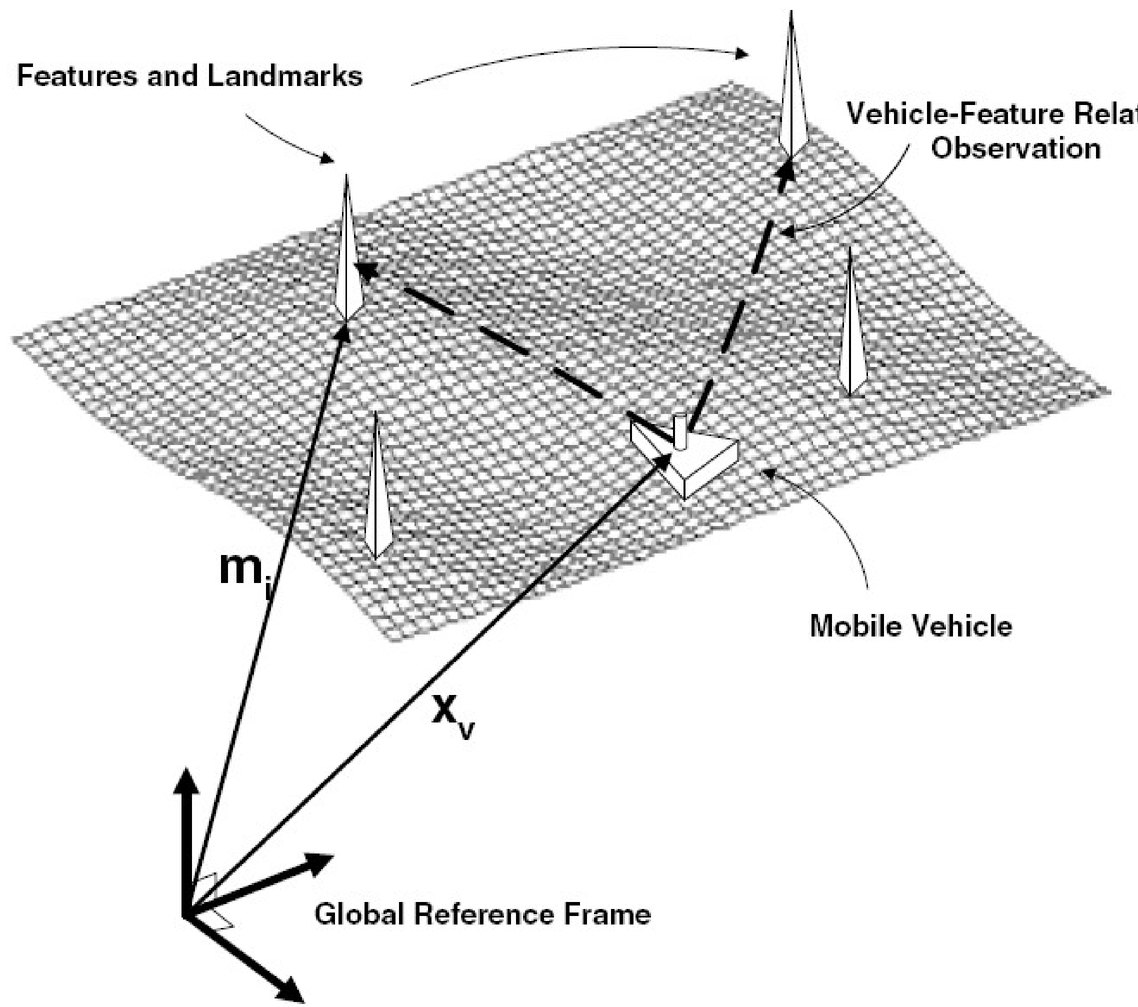
- Online SLAM:

$$p(x_t, m \mid z_{1:t}, u_{1:t}) = \int\int \ldots \int p(x_{1:t}, m \mid z_{1:t}, u_{1:t})\, dx_1 dx_2 \ldots dx_{t-1}$$

Integrations typically done one at a time

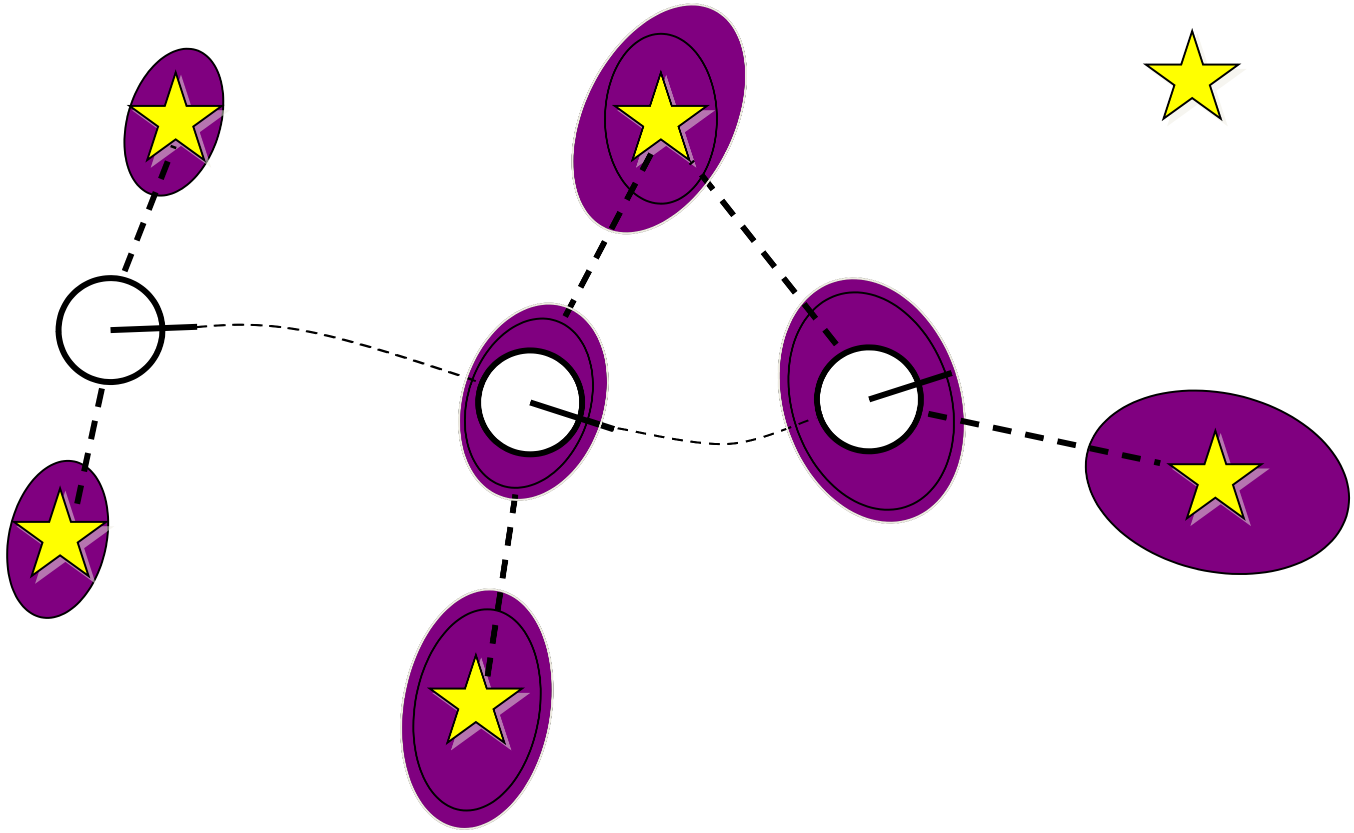
Estimates most recent pose and map!

Structure of the Landmark-based SLAM-Problem



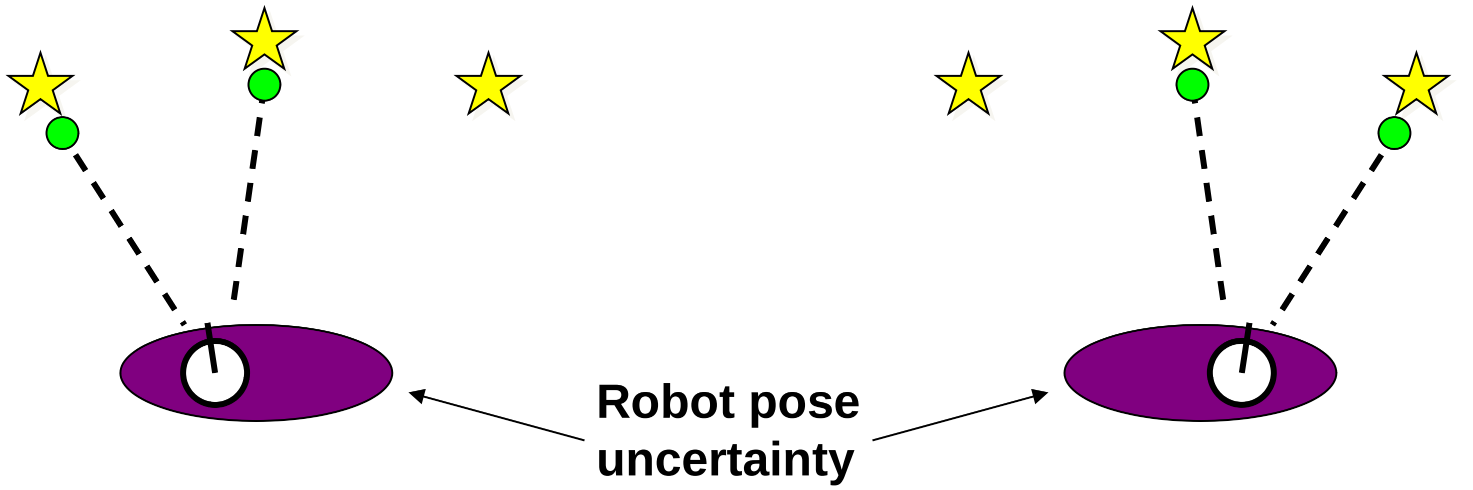
Why is SLAM a hard problem?

SLAM: robot path and map are both **unknown**



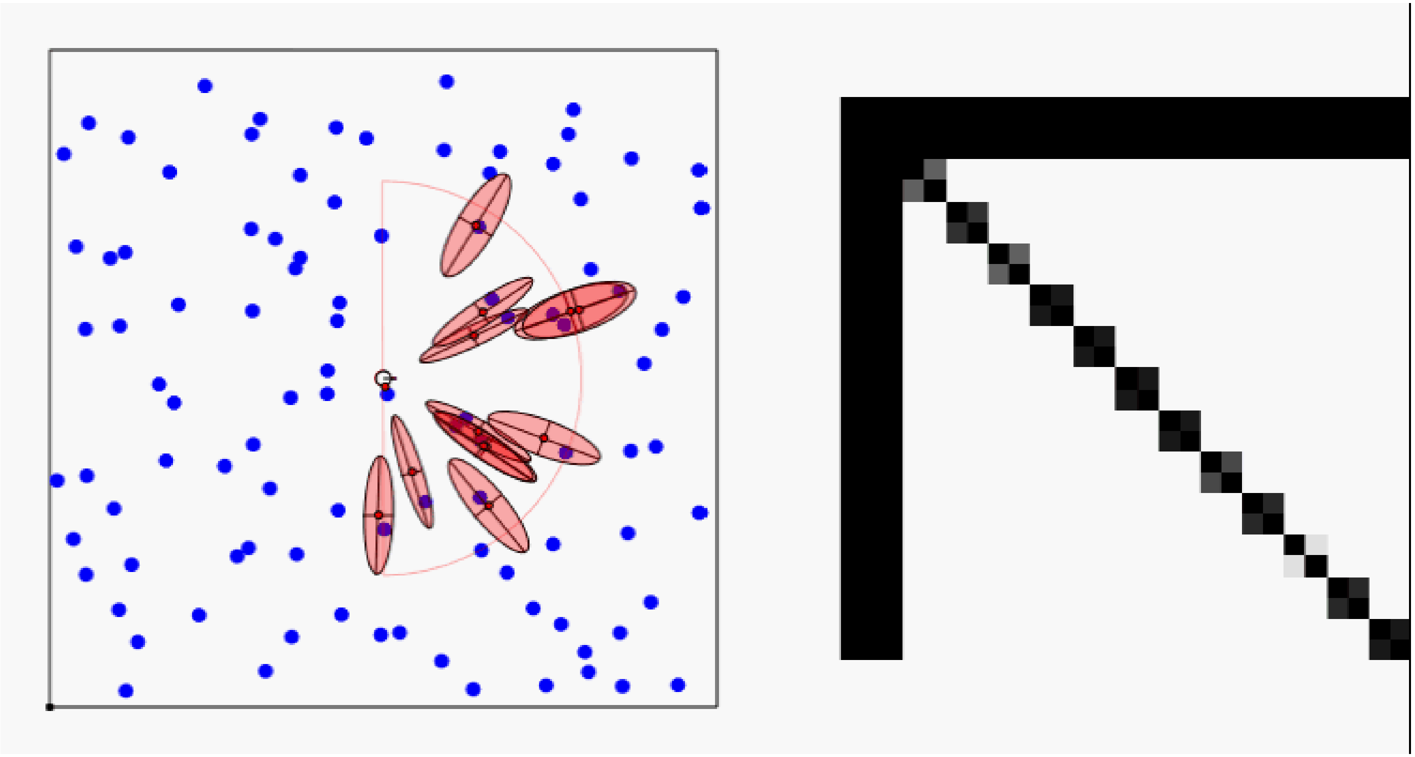
Robot path error correlates errors in the map

Why is SLAM a hard problem?



- In the real world, the mapping between observations and landmarks is unknown
- Picking wrong data associations can have catastrophic consequences
- Pose error correlates data associations

Classical Solution – The EKF



- Approximate the SLAM posterior with a high-dimensional Gaussian [Smith & Cheesman, 1986] …
- Single hypothesis data association

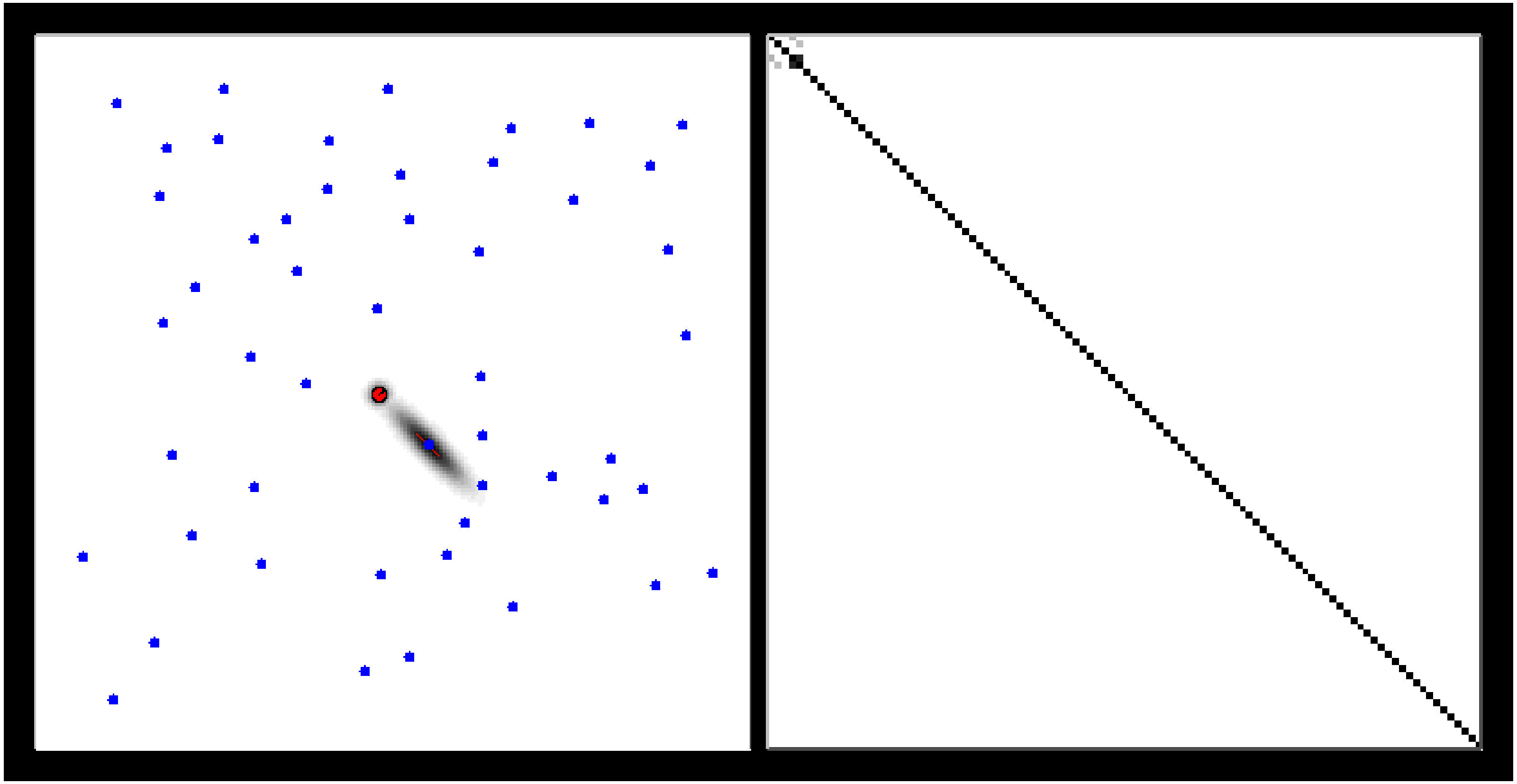
(E)KF-SLAM

- Map with N landmarks:(3+2N)-dimensional Gaussian

$$Bel(x_t, m_t) = \left\langle \begin{pmatrix} x \\ y \\ \theta \\ l_1 \\ l_2 \\ \vdots \\ l_N \end{pmatrix}, \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} & \sigma_{xl_1} & \sigma_{xl_2} & \cdots & \sigma_{xl_N} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{y\theta} & \sigma_{yl_1} & \sigma_{yl_2} & \cdots & \sigma_{yl_N} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_\theta^2 & \sigma_{\theta l_1} & \sigma_{\theta l_2} & \cdots & \sigma_{\theta l_N} \\ \sigma_{xl_1} & \sigma_{yl_1} & \sigma_{\theta l_1} & \sigma_{l_1}^2 & \sigma_{l_1 l_2} & \cdots & \sigma_{l_1 l_N} \\ \sigma_{xl_2} & \sigma_{yl_2} & \sigma_{\theta l_2} & \sigma_{l_1 l_2} & \sigma_{l_2}^2 & \cdots & \sigma_{l_2 l_N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{xl_N} & \sigma_{yl_N} & \sigma_{\theta l_N} & \sigma_{l_1 l_N} & \sigma_{l_2 l_N} & \cdots & \sigma_{l_N}^2 \end{pmatrix} \right\rangle$$

- Can handle hundreds of dimensions

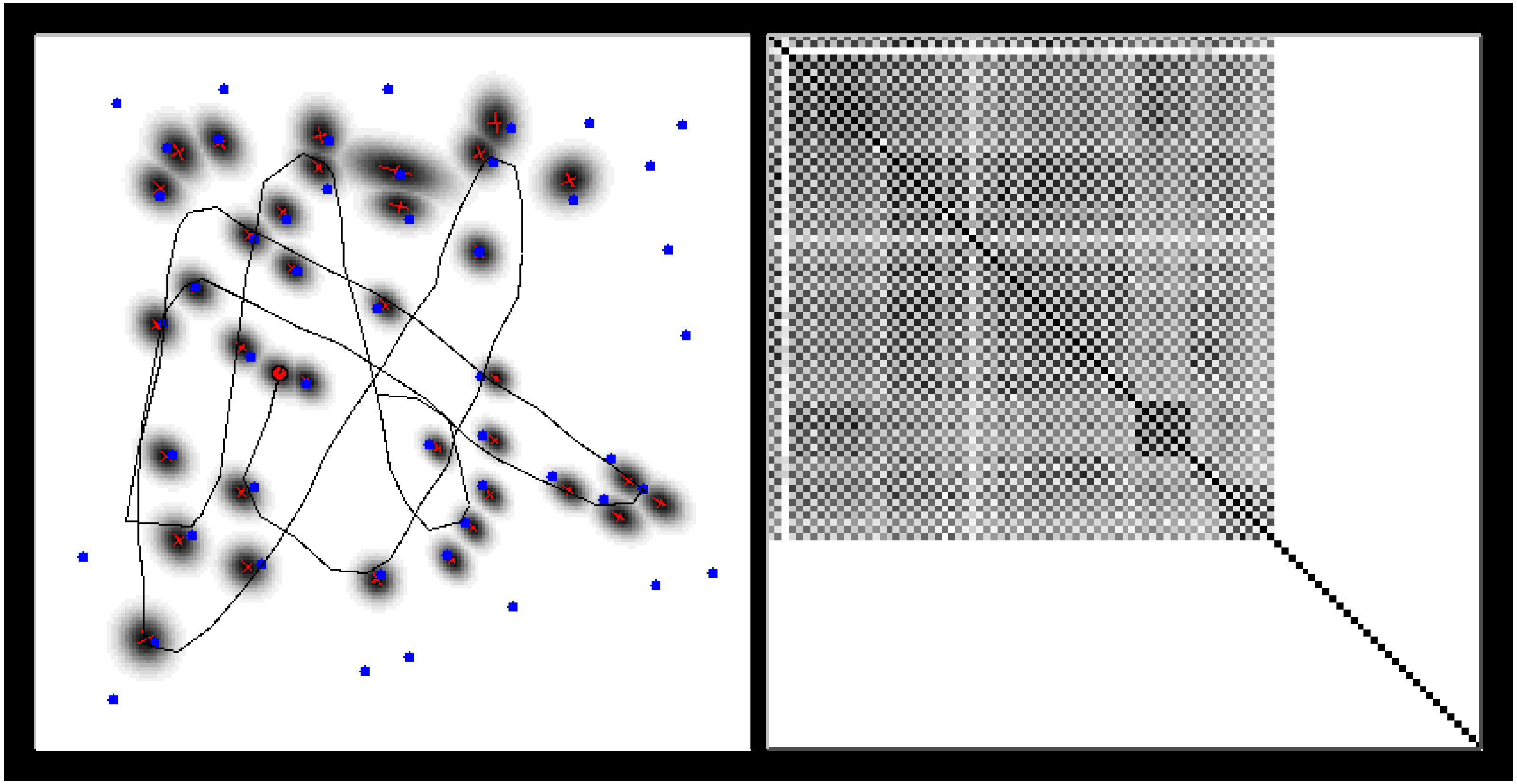
EKF-SLAM



Map

Correlation matrix

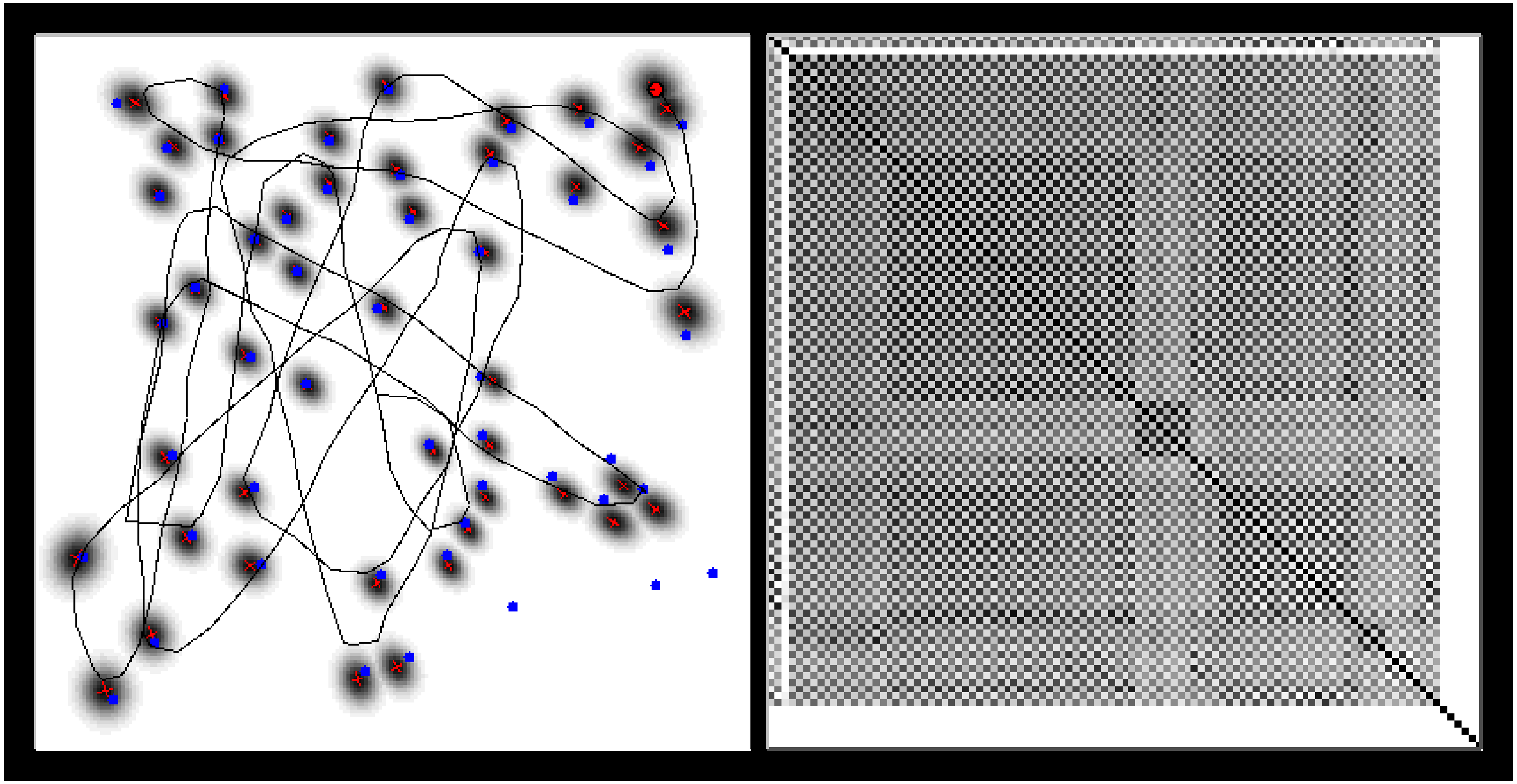
EKF-SLAM



Map

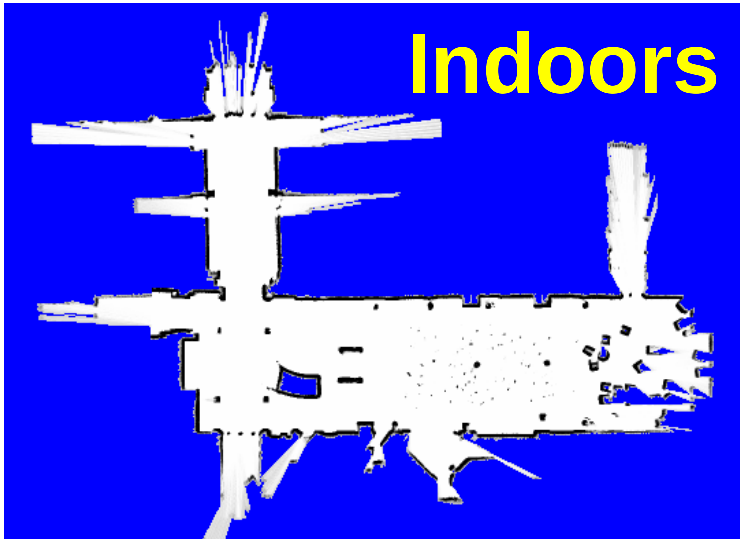
Correlation matrix

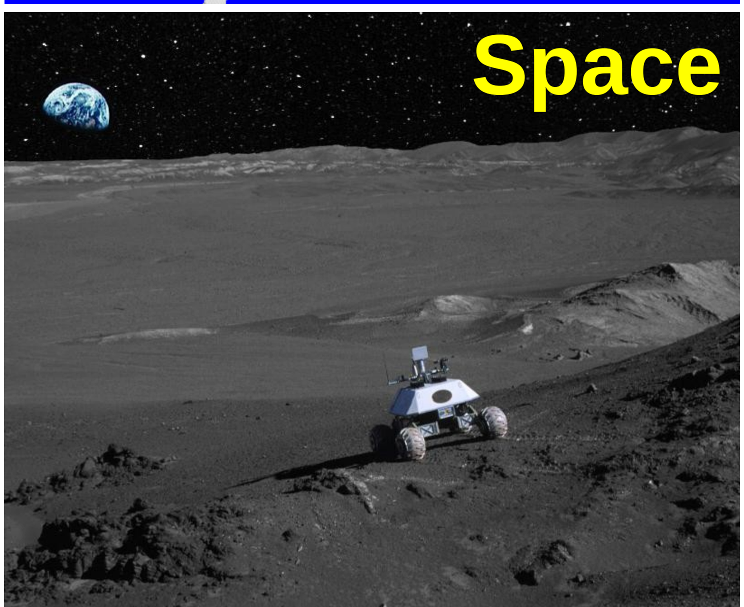
EKF-SLAM



Map Correlation matrix

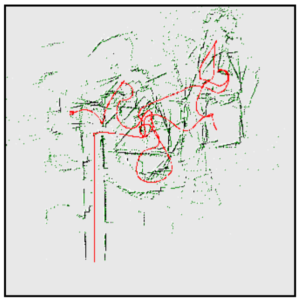
SLAM Applications

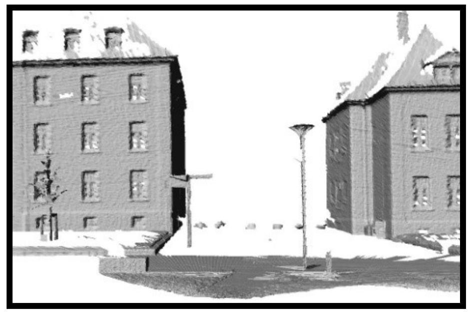


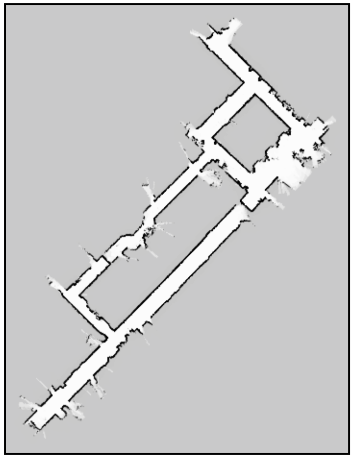


Representations

- Grid maps or scans

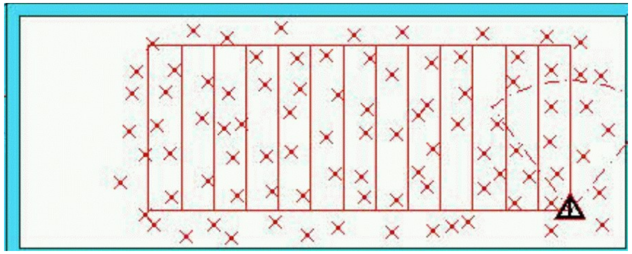


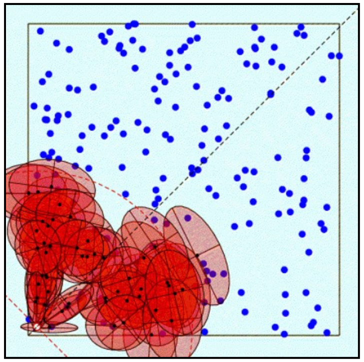


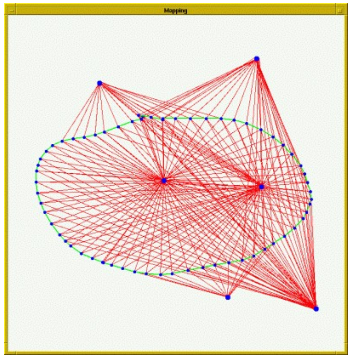


[Lu & Milios, 97; Gutmann, 98: Thrun 98; Burgard, 99; Konolige & Gutmann, 00; Thrun, 00; Arras, 99; Haehnel, 01;...]

- Landmark-based







[Leonard et al., 98; Castelanos et al., 99: Dissanayake et al., 2001; Montemerlo et al., 2002;...

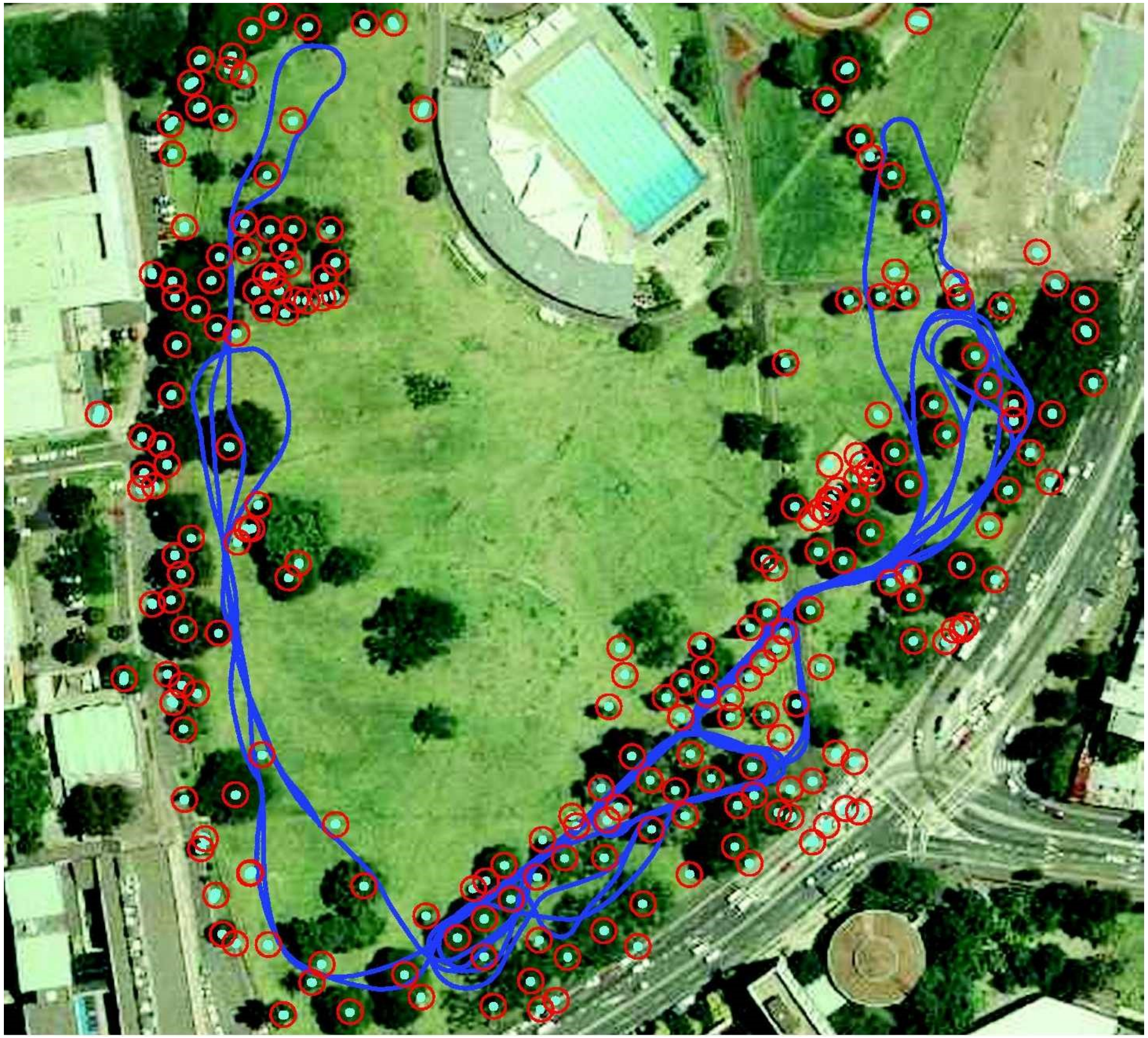
Victoria Park Data Set Vehicle

[courtesy by E. Nebot]

Data Acquisition

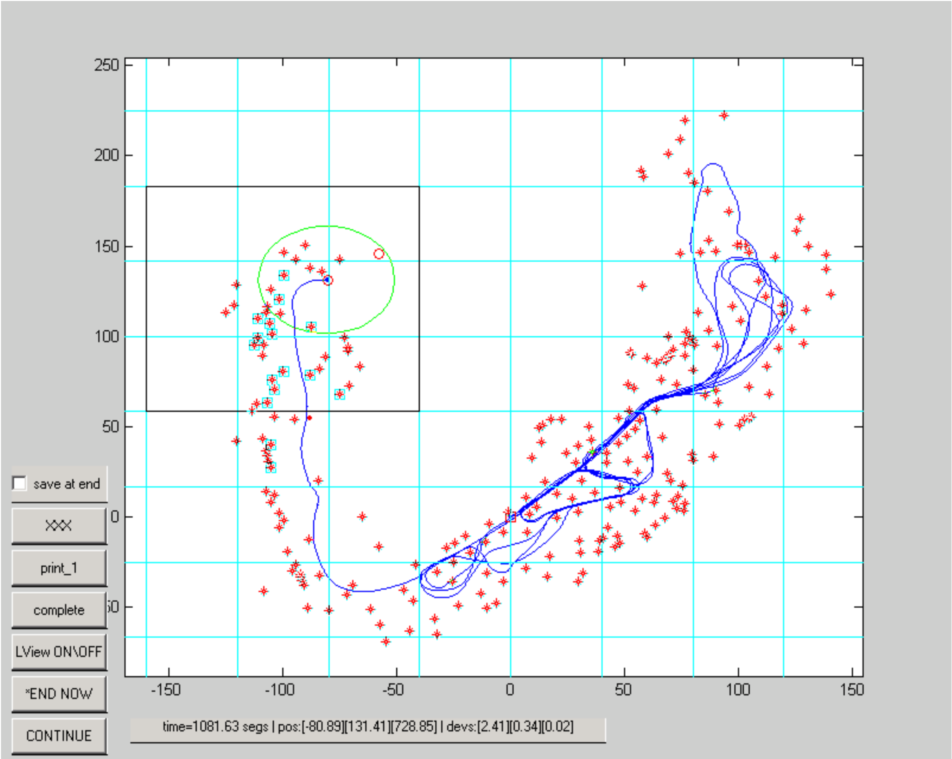
[courtesy by E. Nebot]

Victoria Park Data Set



[courtesy by E. Nebot]

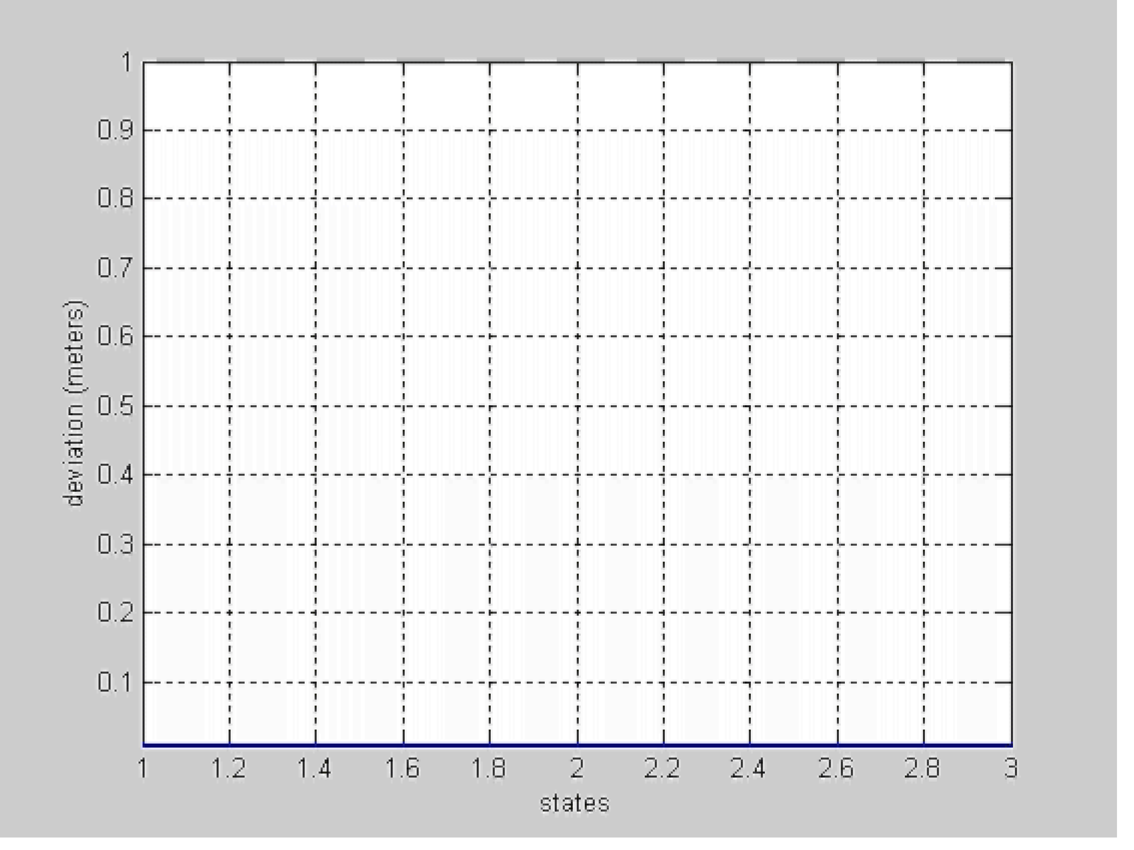
Map and Trajectory



Landmarks Covariance

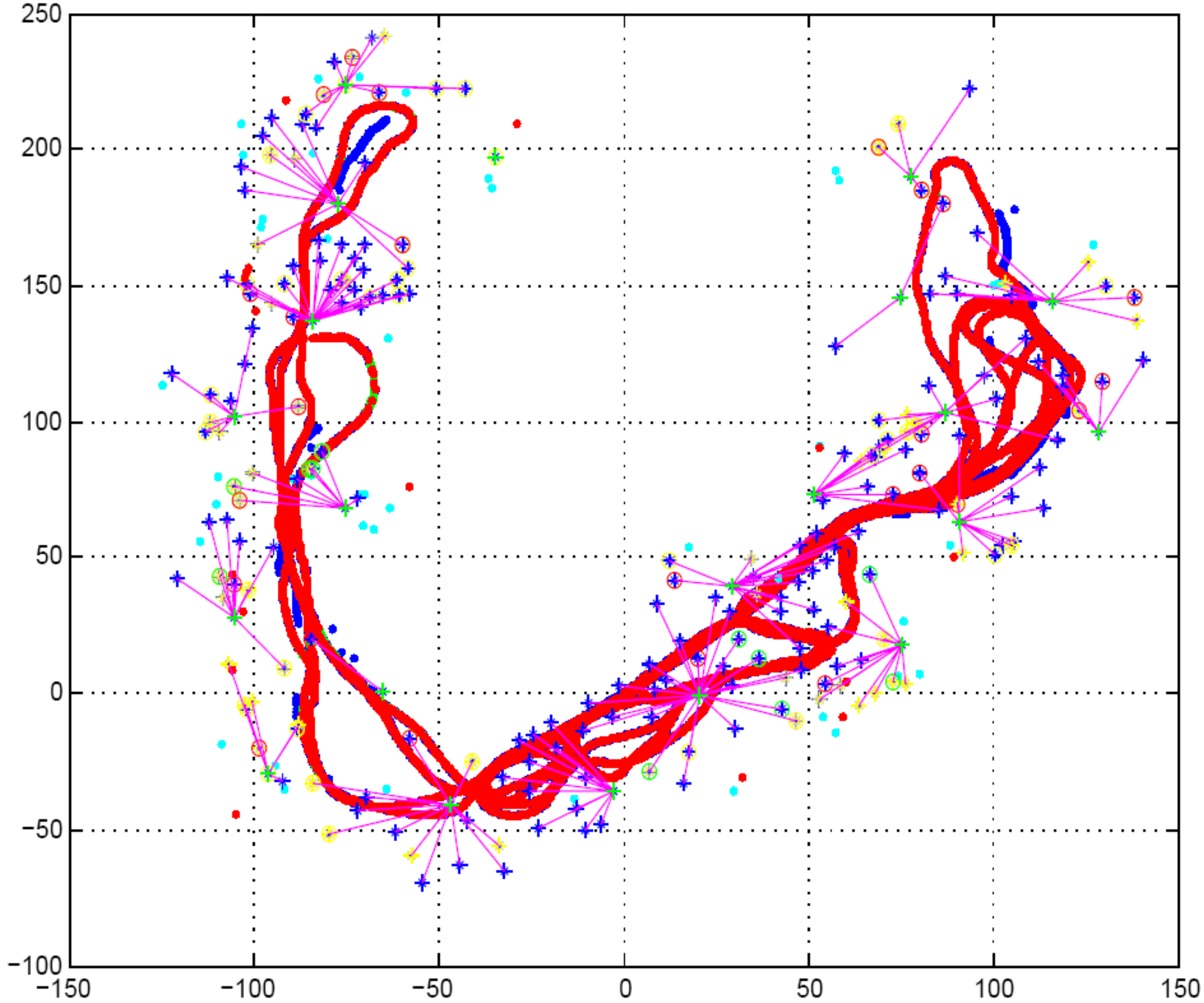
[courtesy by E. Nebot]

Landmark Covariance



[courtesy by E. Nebot]

Estimated Trajectory

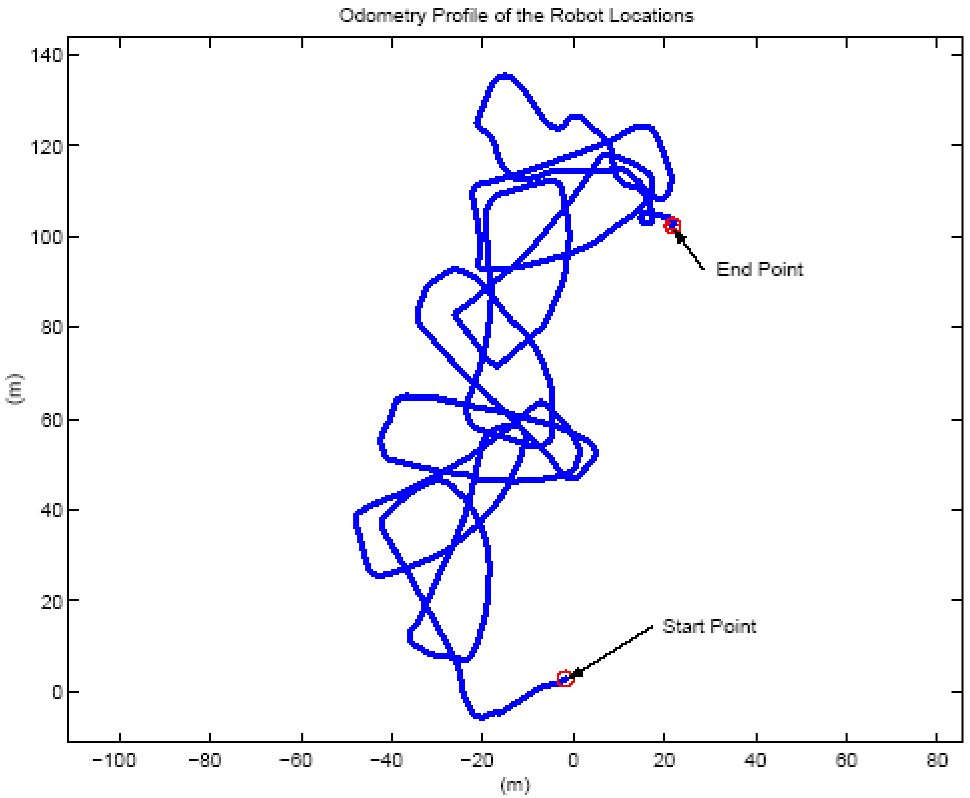


[courtesy by E. Nebot]

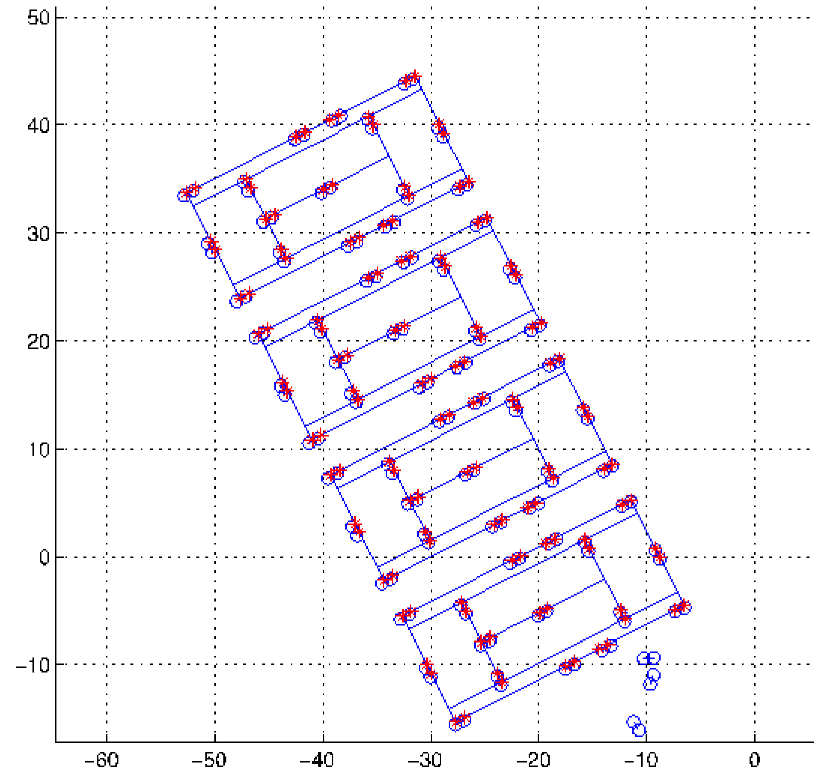
EKF SLAM Application

[courtesy by John Leonard]

EKF SLAM Application



odometry



estimated trajectory

[courtesy by John Leonard]

Approximations for SLAM

- Local submaps
  [Leonard et al.99, Bosse et al. 02, Newman et al. 03]
- Sparse links (correlations)
  [Lu & Milios 97, Guivant & Nebot 01]
- Sparse extended information filters
  [Frese et al. 01, Thrun et al. 02]
- Thin junction tree filters
  [Paskin 03]
- Rao-Blackwellisation (FastSLAM)
  [Murphy 99, Montemerlo et al. 02, Eliazar et al. 03, Haehnel et al. 03]

EKF-SLAM Summary

- Quadratic in the number of landmarks: $O(n^2)$
- Convergence results for the linear case.
- Can diverge if nonlinearities are large!
- Have been applied successfully in large-scale environments.
- Approximations reduce the computational complexity.