Gaussian Filters*

Bayes Filter Implementations

Prof. Mohan Sridharan
Chair in Robot Systems
University of Edinburgh, UK
https://homepages.inf.ed.ac.uk/msridhar/
m.sridharan@ed.ac.uk

*Revised original slides that accompany the book: PR by Thrun, Burgard and Fox.

Bayes Filter Reminder

- Prediction:

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

- Correction:

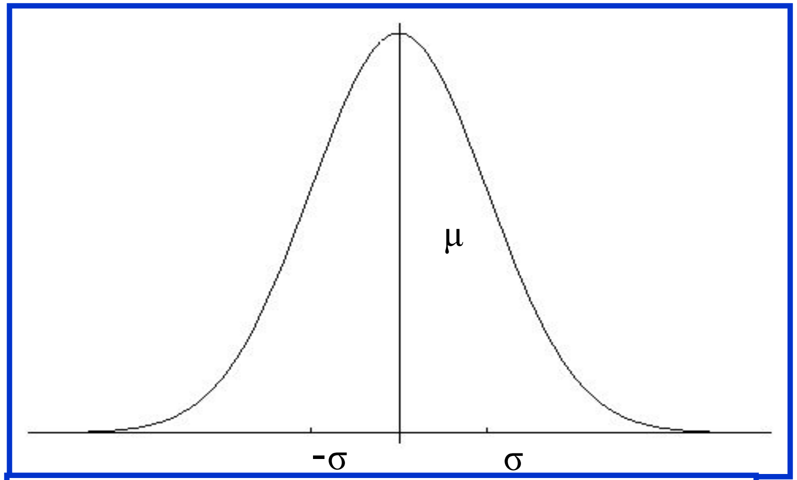
$$bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t)$$

Gaussians (1D and ND)

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

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

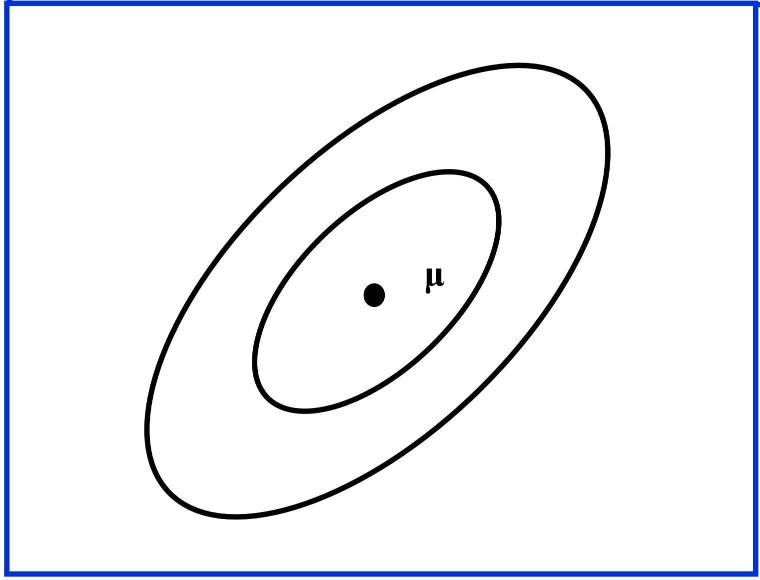
Univariate



$$p(x) \sim N(\mu, \Sigma):$$

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

Multivariate



Properties of Gaussians

$$\begin{pmatrix} X \sim N(\mu, \sigma^2) \\ Y = aX + b \end{pmatrix} \Rightarrow Y \sim N(a\mu + b, a^2\sigma^2)$$

$$\begin{pmatrix} X_1 \sim N(\mu_1, \sigma_1^2) \\ X_2 \sim N(\mu_2, \sigma_2^2) \end{pmatrix} \rightarrow \mathrm{p}(X_1).\mathrm{p}(X_2) \sim \mathrm{N}\ \left(\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\mu_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\mu_2, \frac{1}{\sigma_1^{-2} + \sigma_2^{-2}}\right)$$

Multivariate Gaussians

$$\begin{pmatrix} X \sim N(\mu, \Sigma) \\ Y = AX + B \end{pmatrix} \Rightarrow \; Y \sim N(A\mu + B, A\Sigma A^T)$$

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

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

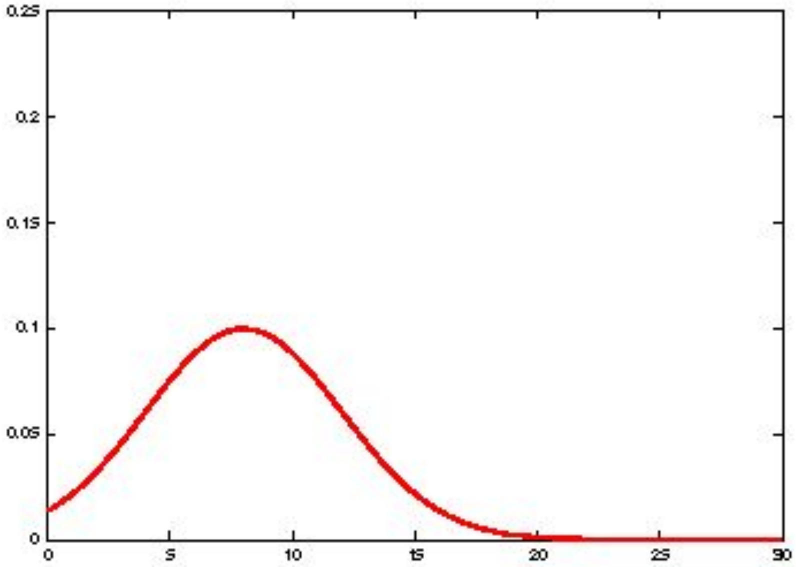
C_t Matrix (kxn) that describes how to map the state x_t to an observation z_t.

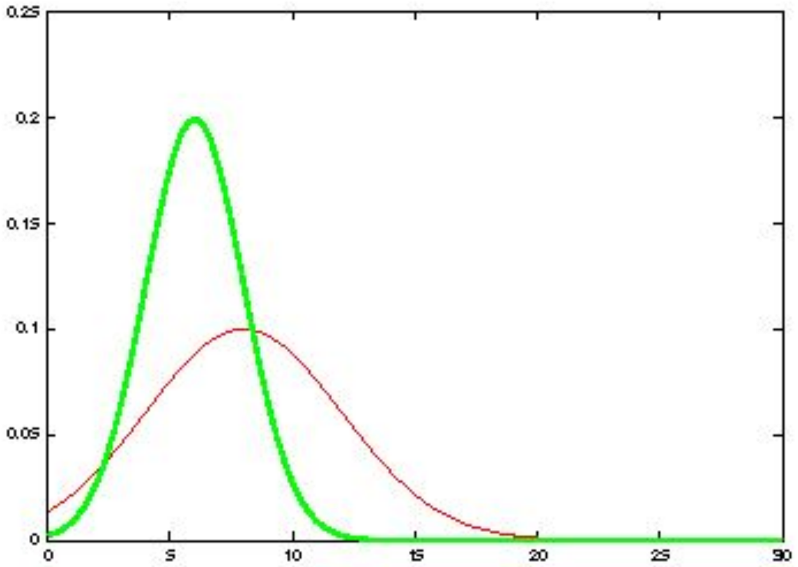
ε_t

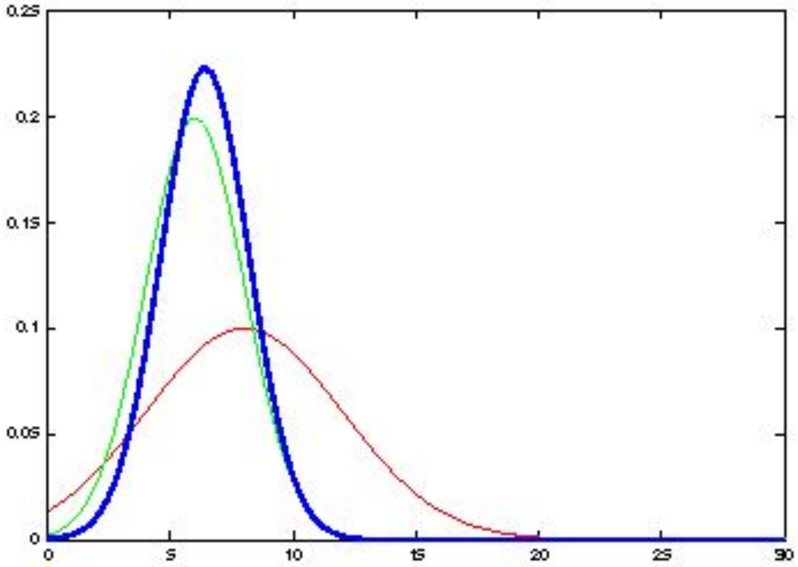
δ_t

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 Updates in 1D



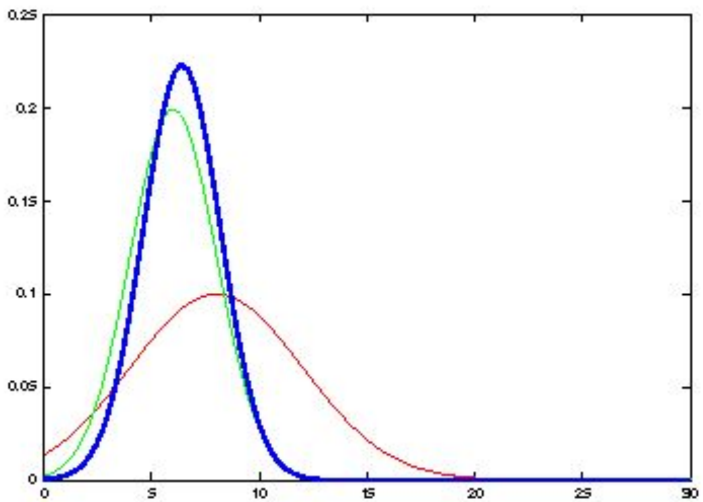


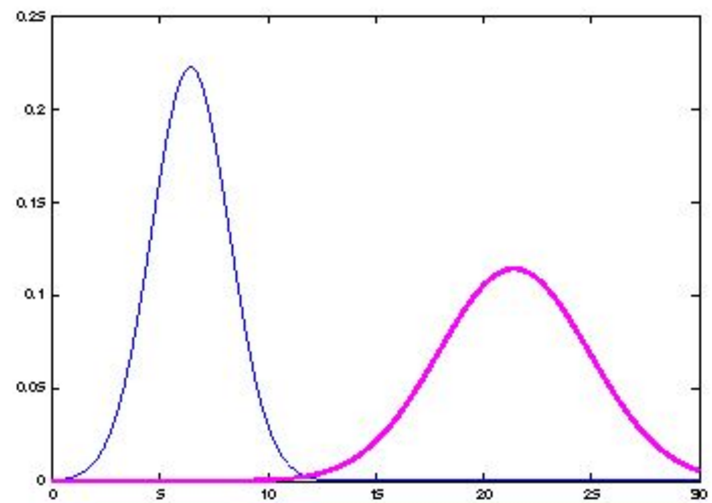


Kalman Filter Control Updates

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

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

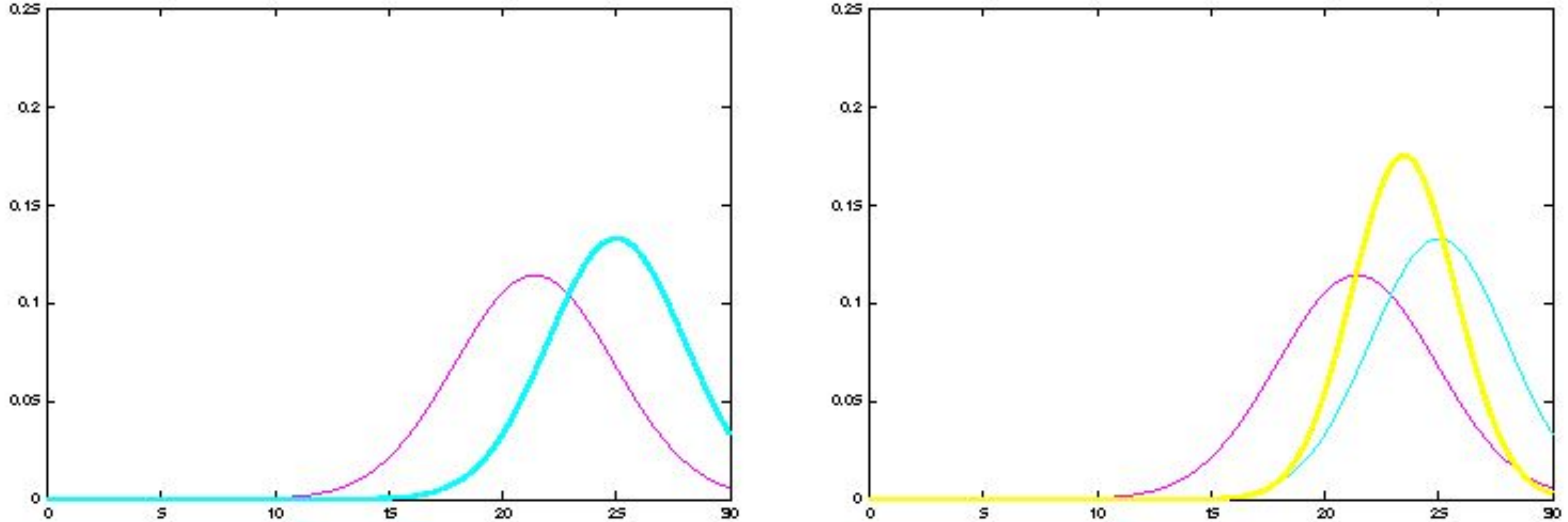




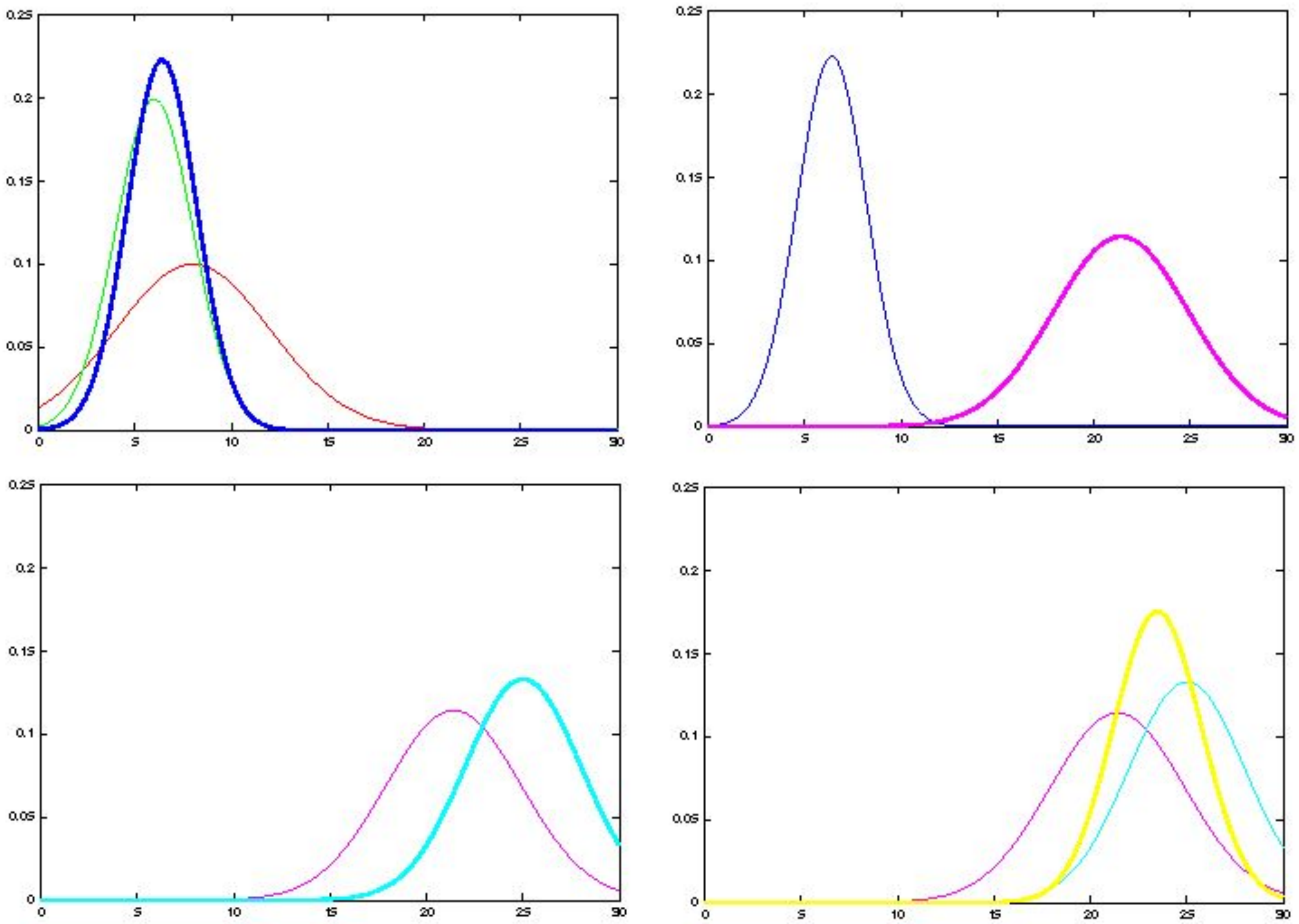
Kalman Filter Measurement Updates

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

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



Kalman Filter Updates



Linear Gaussian system: Initialize

- Initial belief is normally distributed:

$$bel(x_0) = N(x_0; \mu_0, \Sigma_0)$$

Kalman Filter Algorithm

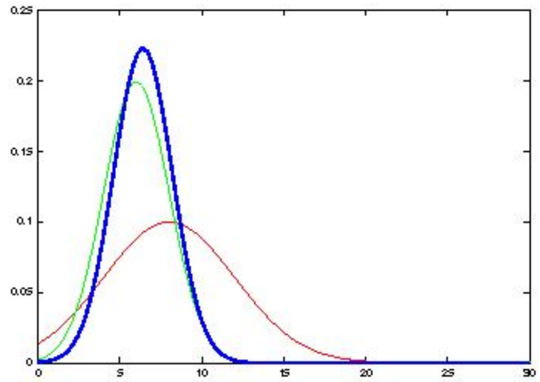
1. Algorithm **Kalman_filter**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

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

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

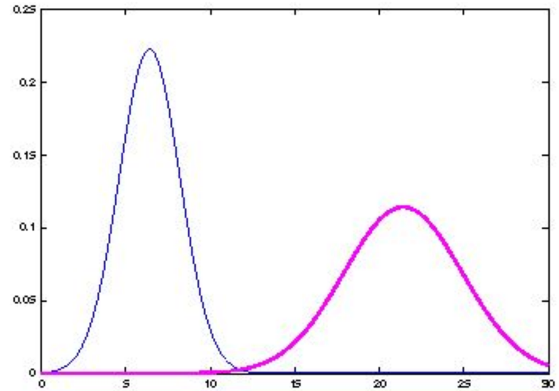
The Prediction-Correction Cycle

Prediction

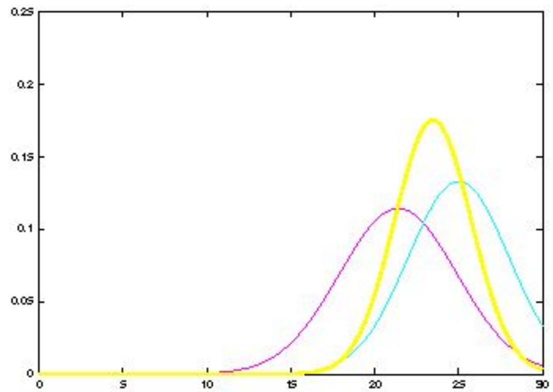


$$\overline{bel}(x_t) = \left\{ \begin{array}{l} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{array} \right\}$$

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

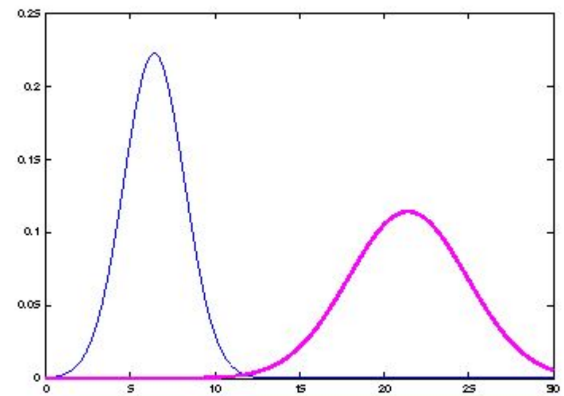


The Prediction-Correction Cycle



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

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



Correction

The Prediction-Correction Cycle

Prediction

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

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

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

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

Correction

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!

- Limiting assumptions:
  - Observations are linear functions of state. State transition are linear.
  - Unimodal beliefs.

- Most robotics systems are nonlinear and beliefs are multimodal!

Extended Kalman Filter (EKF)

- Most realistic robotic problems involve nonlinear functions.

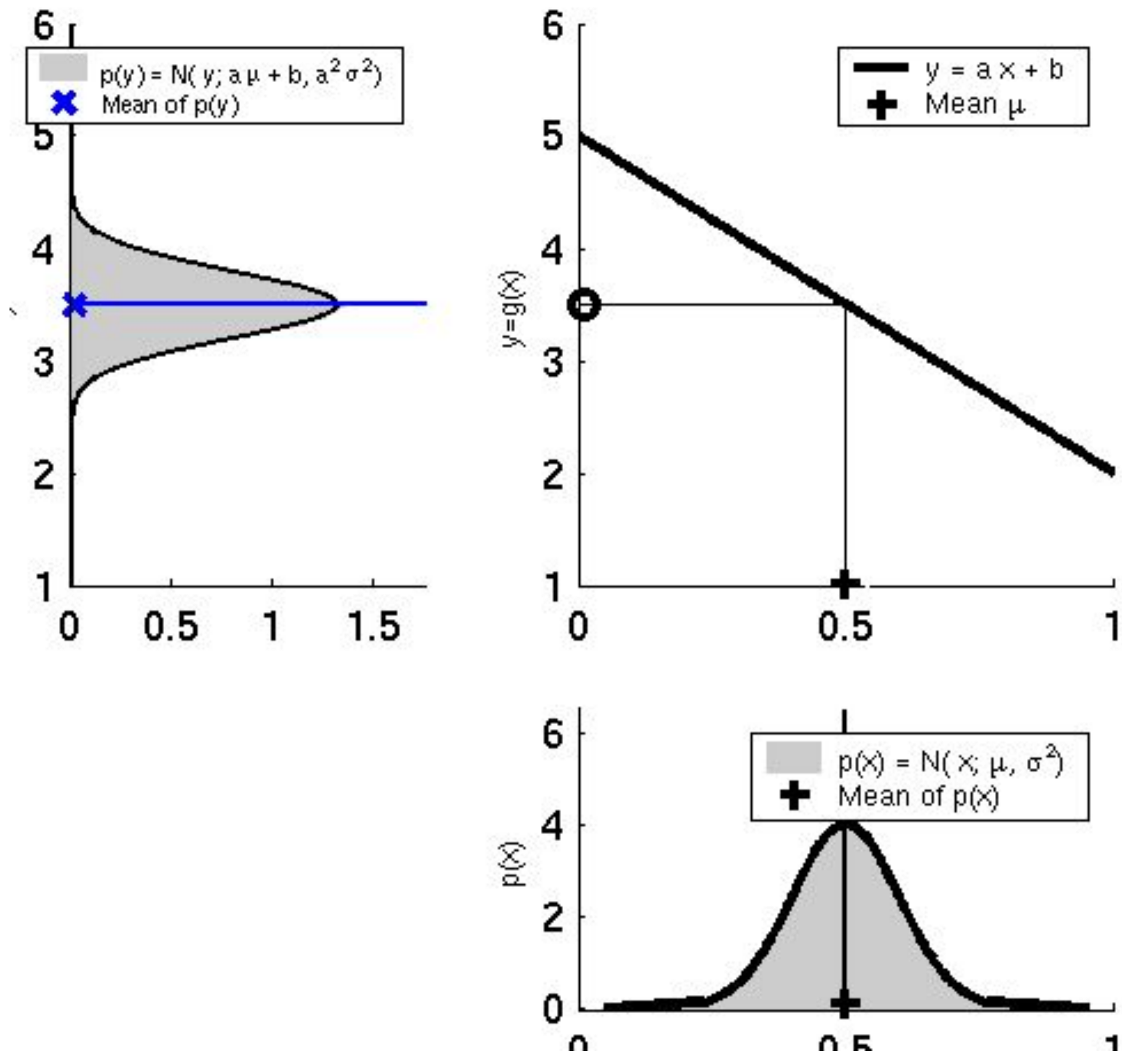
- EKF supports such non-linear functions; relaxes linearity assumption.

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

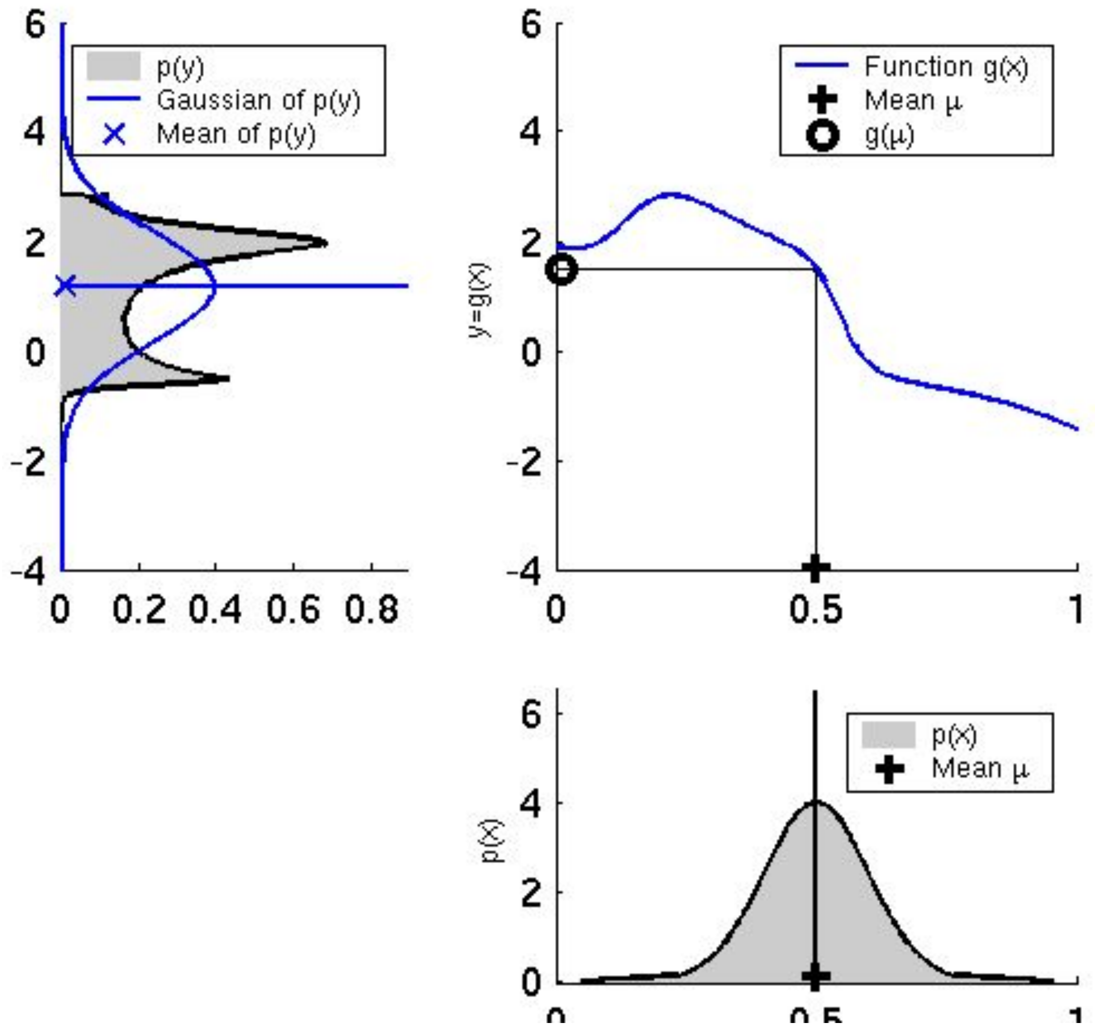
$$z_t = h(x_t)$$

- However, beliefs are no longer Gaussian ☹

Linearity Assumption Revisited



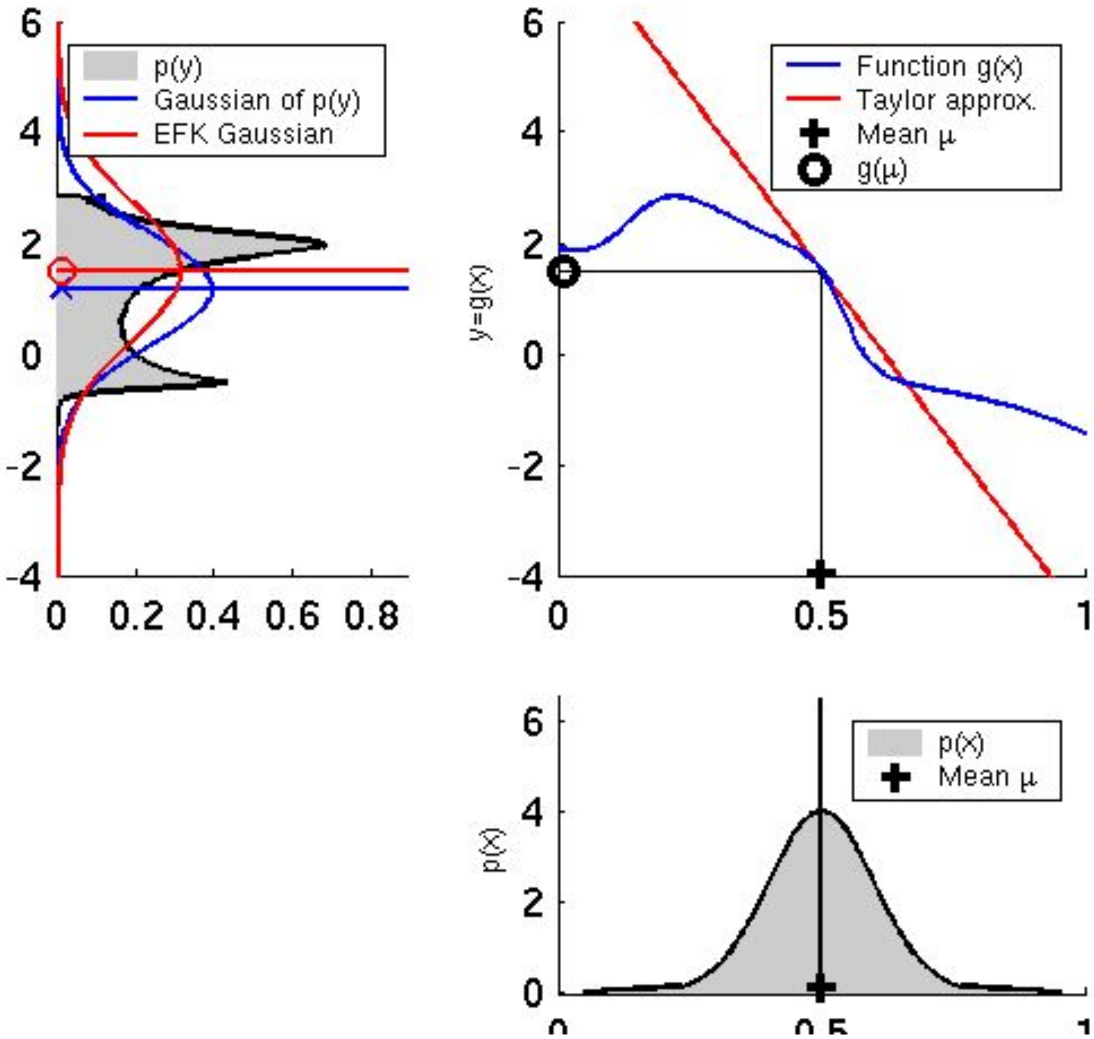
Non-linear Function



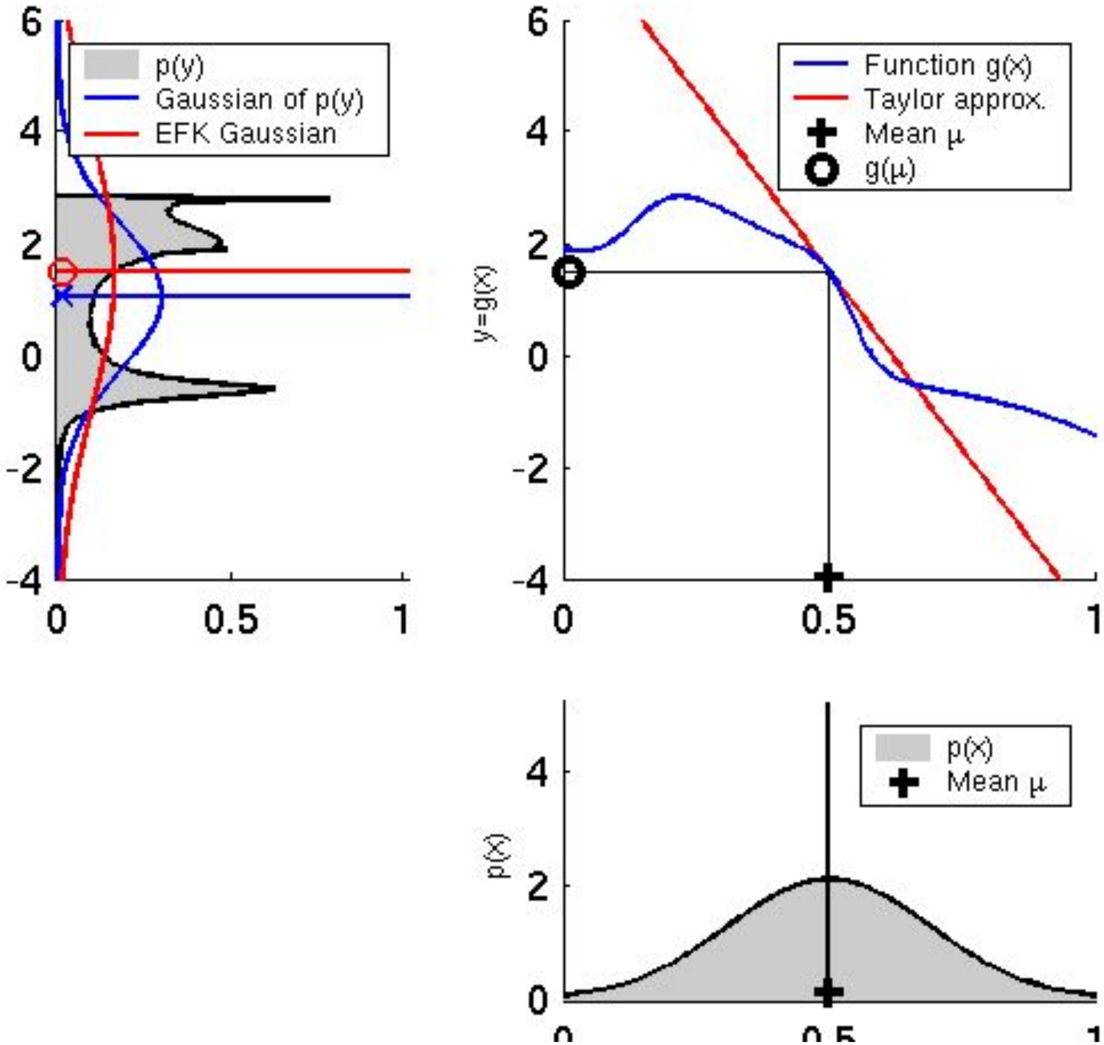
Linearization in EKF

- Sequence of steps for linearization in EKF.
- Compute tangent to function *g()* at mean.
- Consider the tangent as the linearized approximation of *g()*.
- Project *p(x)* through linear approximation.
- Compute mean and covariance of *y*. This defines the Gaussian approximation of the underlying non-linear transformation.

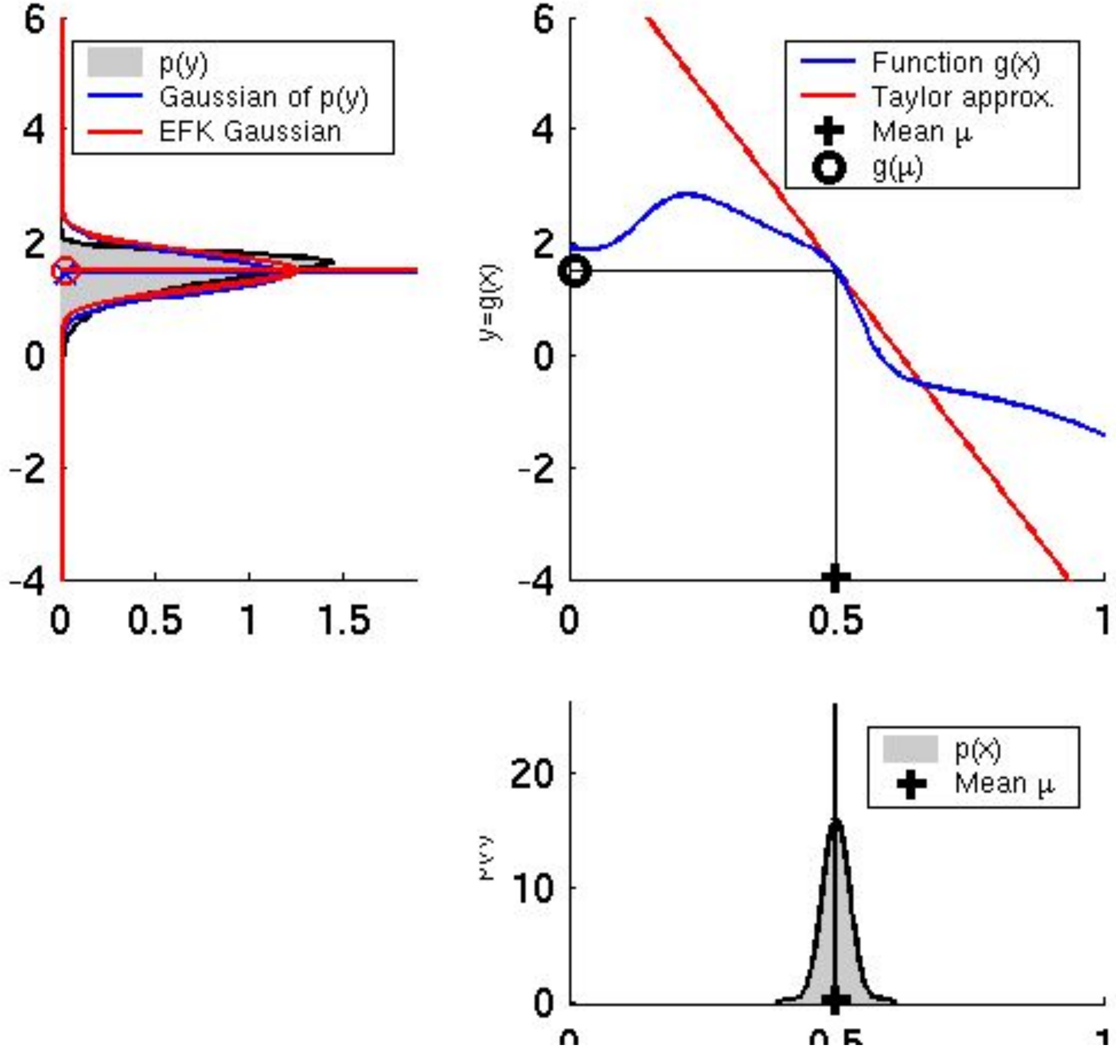
EKF Linearization (1)



EKF Linearization (2)



EKF Linearization (3)



Why Linearize?

- Remember limiting assumptions of KF:
  - Observations are linear functions of state. State transition are linear.
  - Unimodal beliefs.

- Assumptions do not hold in practice.

- Relax linearity assumption. However, makes beliefs non-Gaussian ☹

- EKF computes Gaussian approximation of true belief through linearization of non-linear functions *g()* and *h()*.

- Achieve linearization through (first-order) Taylor expansion (Section 3.3.2, PR).

EKF Linearization: First Order Taylor Series Expansion

- Prediction:

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

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

- Correction:

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

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

- Derivation of EKF (Section 3.3.4, PR).

EKF Algorithm

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

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

4. Correction:
5. $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$ ← $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
6. $\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))$ ← $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
7. $\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$ ← $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$
8. Return μ_t, Σ_t

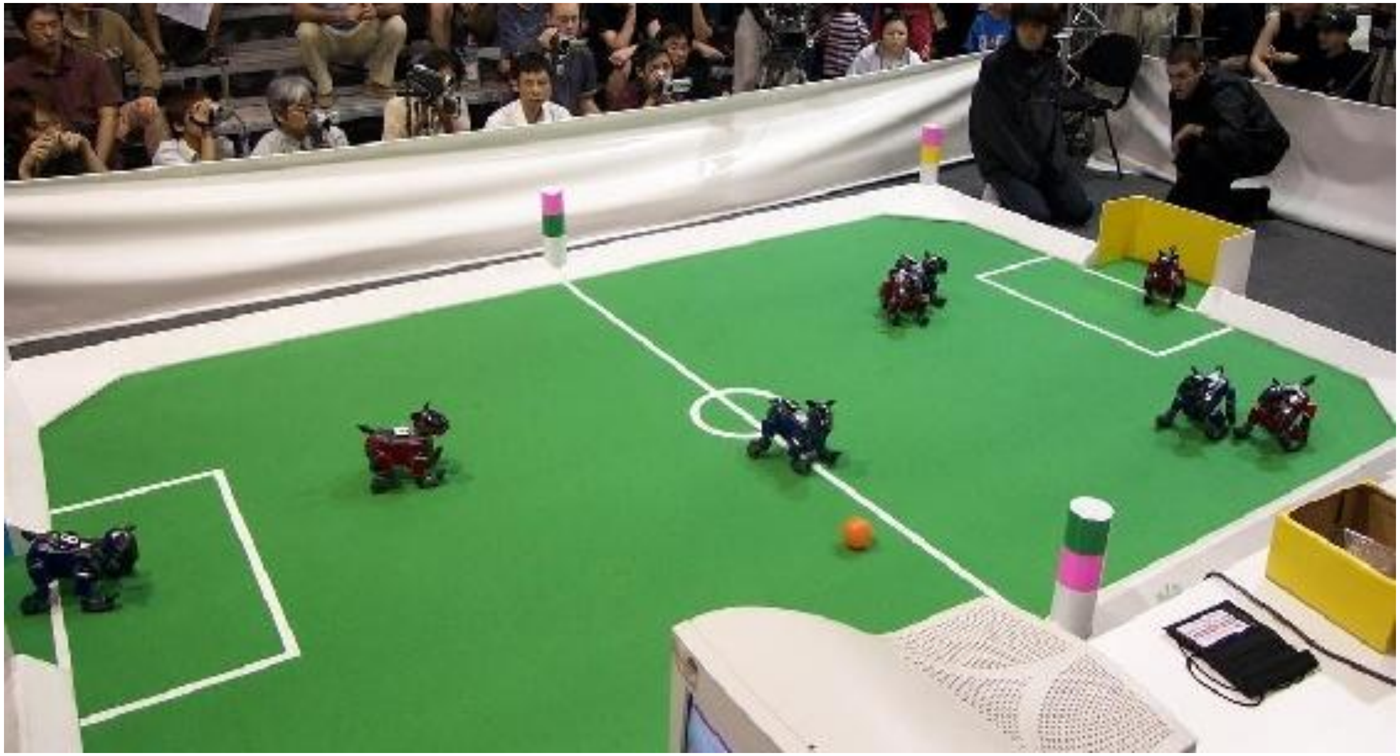
$$H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t} \qquad G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}$$

Localization

"Using sensory information to locate the robot in its environment is the most fundamental problem to providing a mobile robot with autonomous capabilities." [Cox '91]

- **Given**
  - Map of the environment.
  - Sequence of sensor measurements.
- **Wanted**
  - Estimate of the robot's position.
- **Problem classes**
  - Position tracking.
  - Global localization.
  - Kidnapped robot problem (recovery).

Landmark-based Localization



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

- Stochastic linearization through *unscented transform*.
- Extract *sigma-points* from Gaussian.
  - Mean and symmetric points along main axes of covariance.
  - N-dim Gaussian => 2N+1 sigma points.
- Two weights for each sigma point, one each to compute mean and covariance.
- Encode additional knowledge about underlying distribution.
- Project sigma points through *g()*.
- Compute mean and covariance of projected points.

Unscented Transform

Sigma points | Weights

$$\chi^{[0]} = \mu \quad w_m^{[0]} = \frac{\lambda}{n+\lambda}, \quad w_c^{[0]} = \frac{\lambda}{n+\lambda} + (1-\alpha^2+\beta)$$

$$\chi^{[i]} = \mu \pm \left(\sqrt{(n+\lambda)\Sigma}\right)_i \quad w_m^{[i]} = w_c^{[i]} = \frac{1}{2(n+\lambda)} \qquad \text{for } i = 1, \ldots, 2n$$

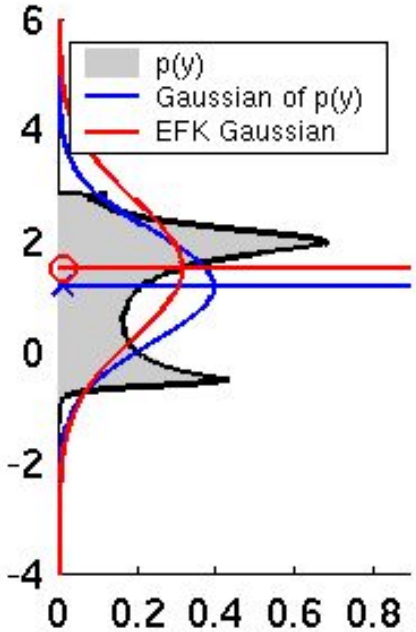
Pass sigma points through nonlinear function:

$$y^{[i]} = g(\chi^{[i]})$$

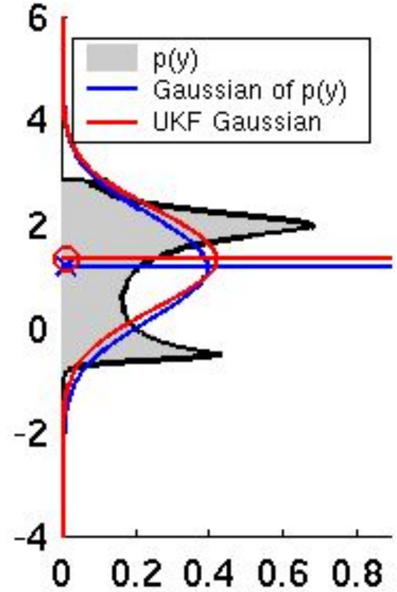
Recover mean and covariance:

$$\mu' = \sum_{i=0}^{2n} w_m^{[i]} y^{[i]}; \quad \Sigma' = \sum_{i=0}^{2n} w_c^{[i]} (y^{[i]} - \mu)(y^{[i]} - \mu)^T$$

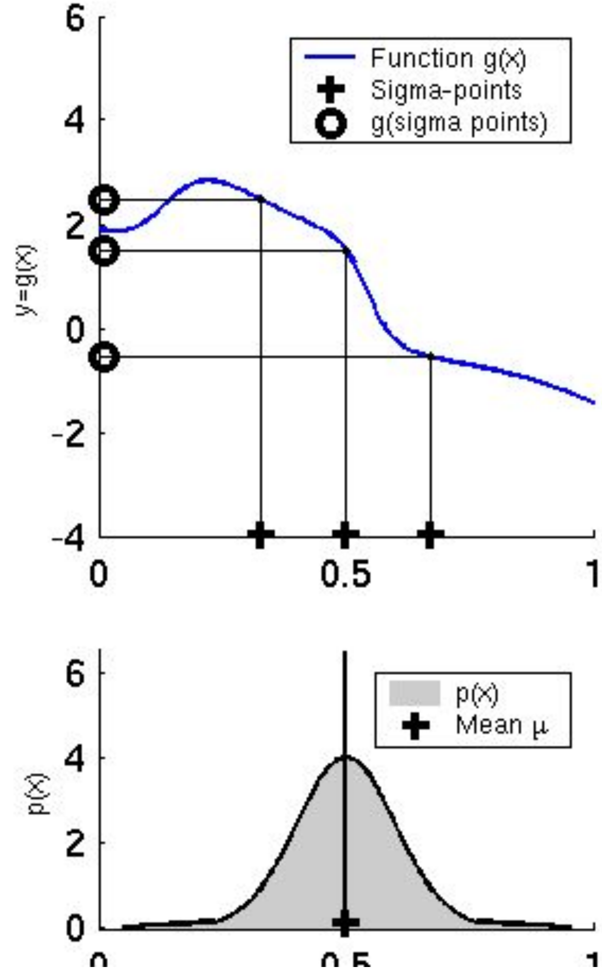
Linearization via Unscented Transform



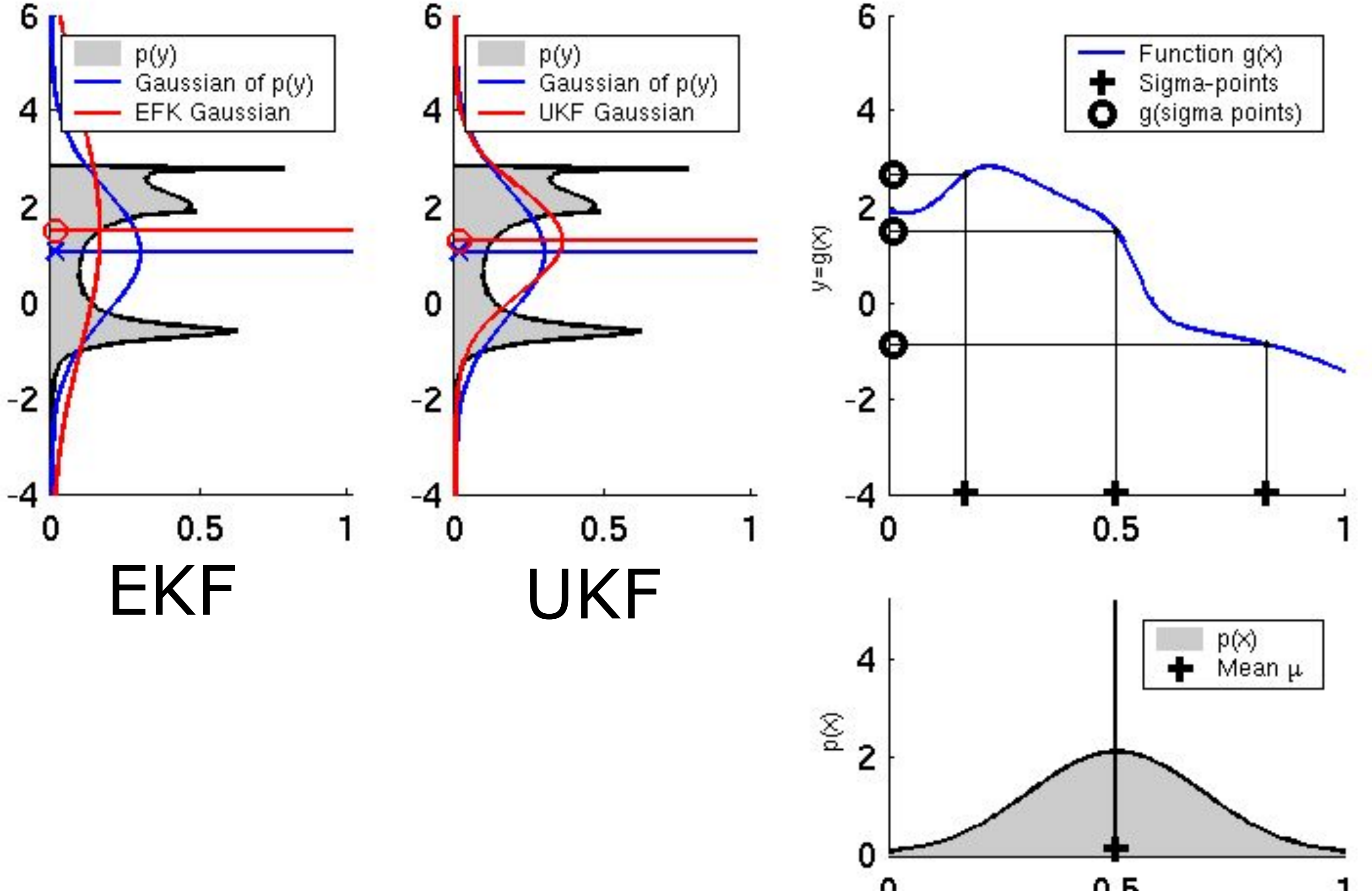
EKF



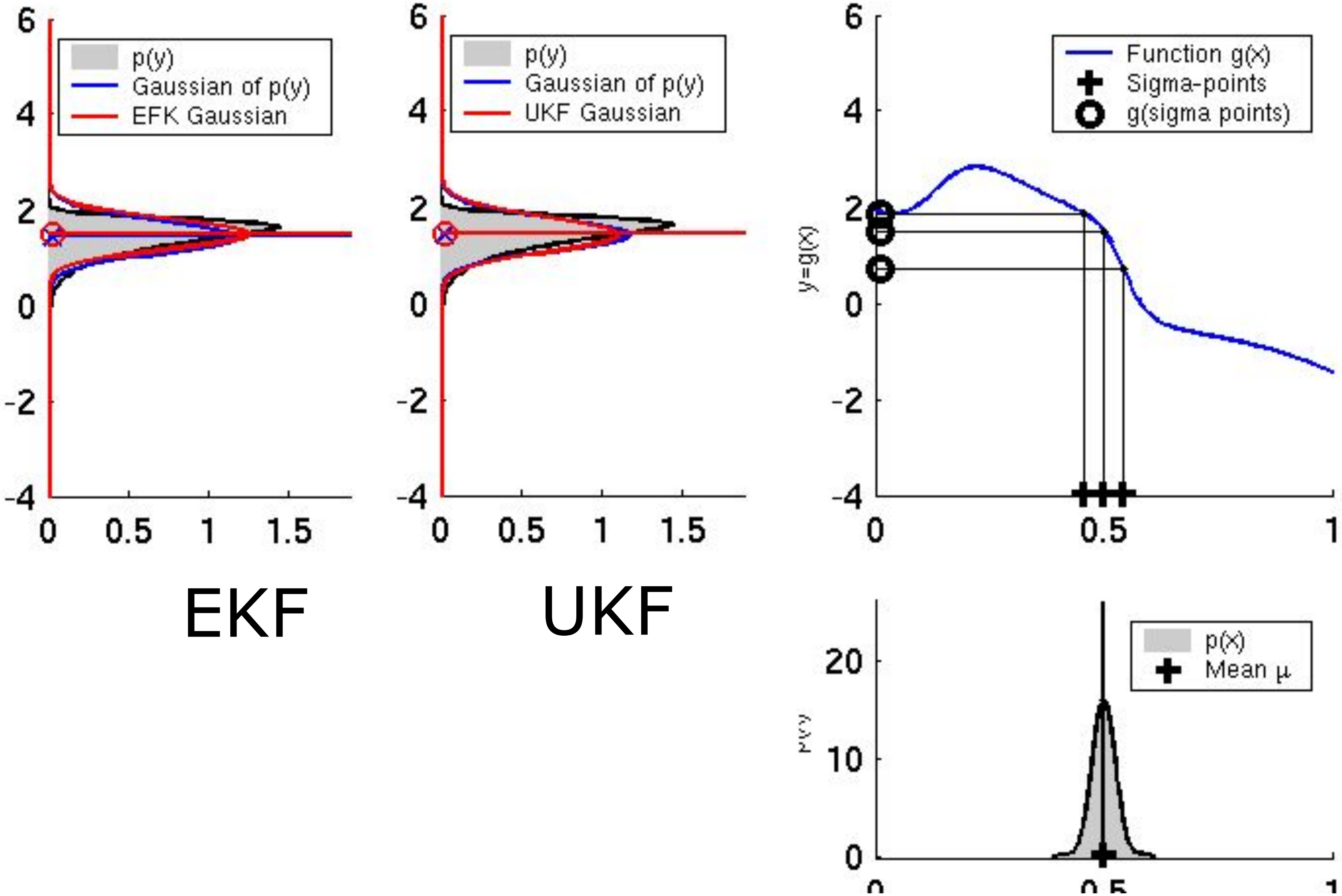
UKF



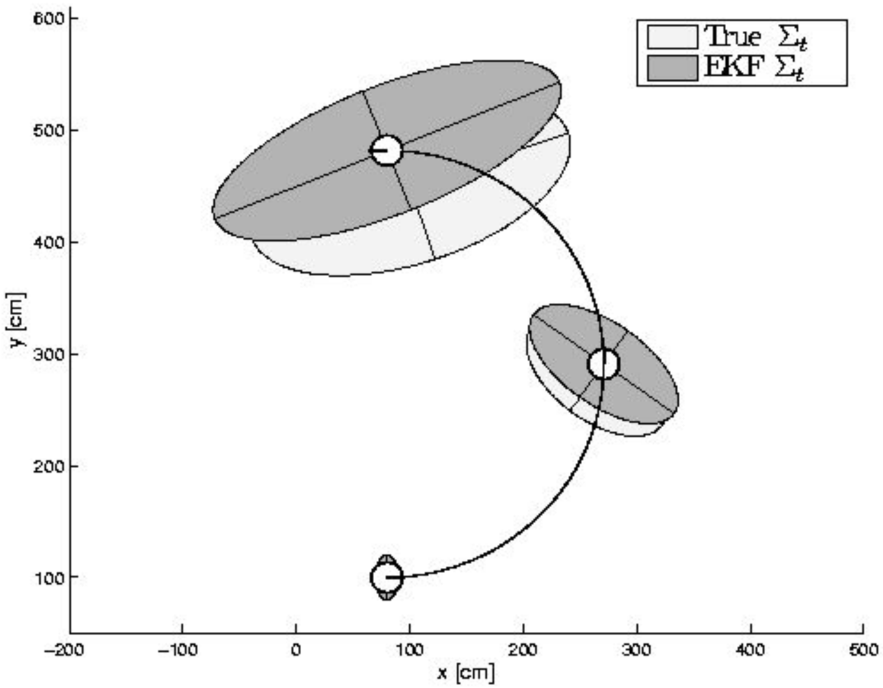
UKF Sigma-Point Estimate (2)



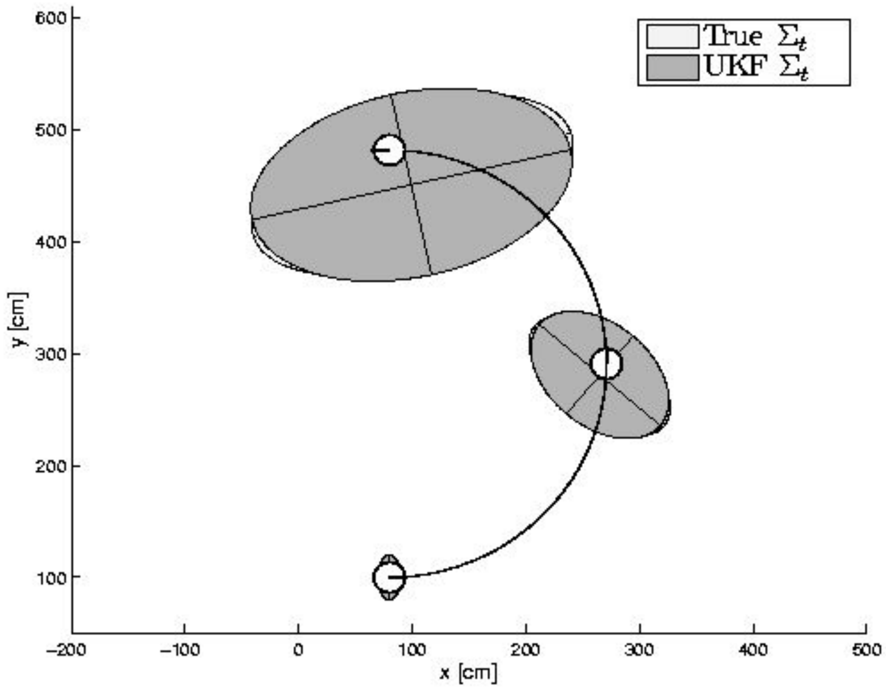
UKF Sigma-Point Estimate (3)



Prediction Quality



EKF

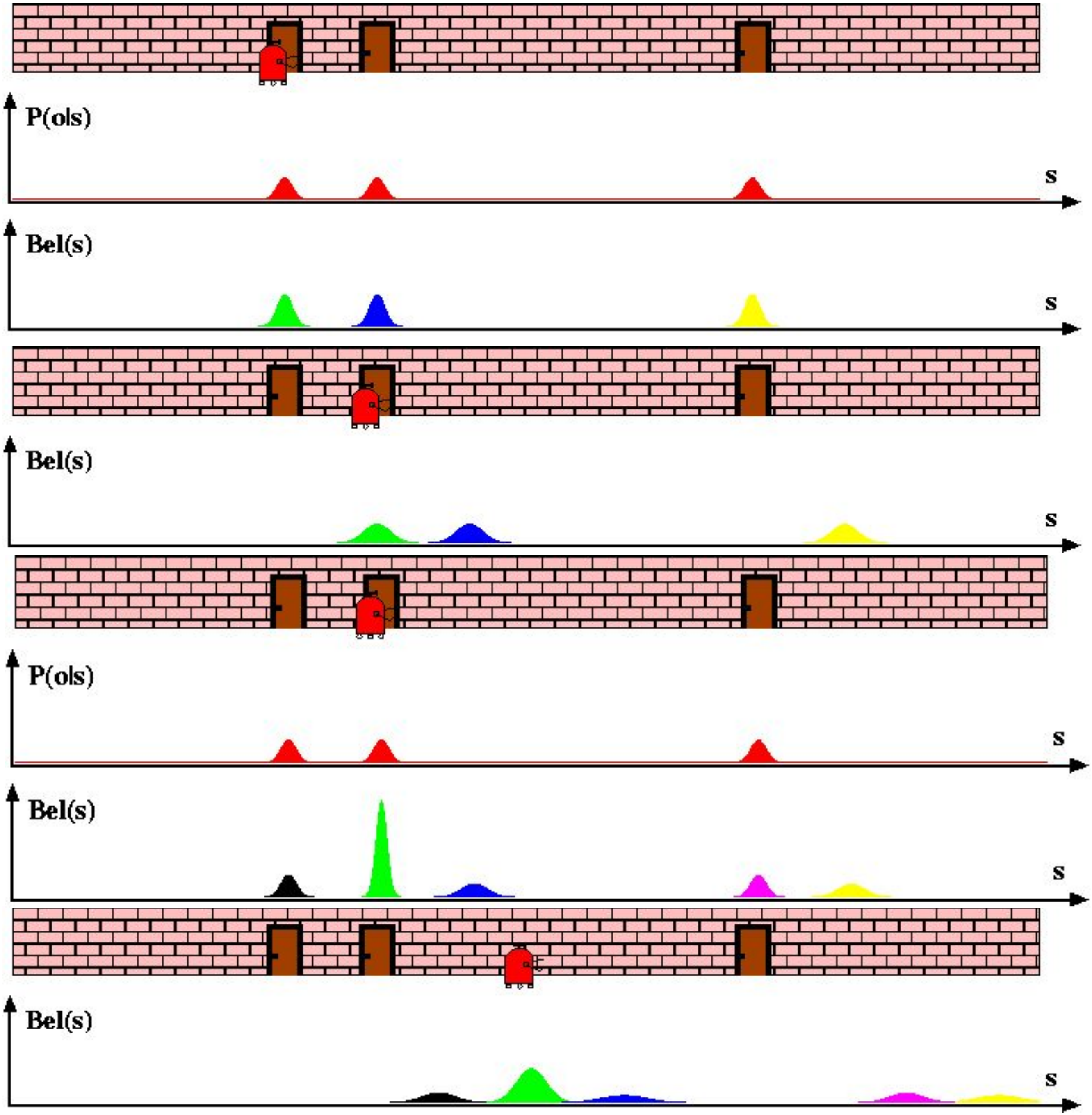


UKF

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!

Hypothesis Tracking



Localization With MHT

- How to represent belief for multiple hypotheses?
- Each hypothesis is tracked by a Kalman filter.
- **Additional problems**:
  - Data association: Which observation corresponds to which hypothesis?
  - Hypothesis management: When to add / delete hypotheses?
- Lot of work on target tracking, motion correspondence etc.

Summary

- Gaussian filters.
- Kalman filter: linearity assumption.
  - Robot systems non-linear.
  - Works well in practice.
- Extended Kalman filters: linearization.
  - Tangent at the mean.
- Unscented Kalman filters: better linearization.
  - Sigma control points.

- Information filter: dual of KF, uses canonical parameterization *(Section 3.5, PR).*