Filters

IAR Lecture 9

Barbara Webb

Keeping track

- Our aim is to keep track of the state of our robot, in particular of its location or pose relative to a map.
- Have multiple sources of information, about (intended) self motion and observed environment.
- All the information has some uncertainty, so we will maintain a probabilistic estimate.
- We want to 'filter' the current state and information to maintain the best possible estimate (c.f. sensor fusion).
- We can chose different ways of representing the probability distribution of our estimate.
- We'll start by using the first two moments, i.e. the mean and the variance, and assuming the normal distribution.

Representing the probability of our estimate as a normal distribution $bel(x_t) = N(x_t; \mu_t, \Sigma_t)$

We are trying to estimate the state of a robot $x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$

e.g. its pose $x = [x, y, \theta]'$

At time t we apply some control signal $u = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}$

e.g. set wheel speeds $u = [v_R, v_L]'$

Then we take a measurement $z = [z_1, \ldots, z_k]$

e.g. distance to three known landmarks $z = [dist_1, dist_2, dist_3]$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad u = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}$$

If we assume the system is linear with Gaussian noise…

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

A_t Matrix (nxn) that describes how the state evolves from t to $t\text{-}1$ without controls or noise. Could be identity matrix.

B_t Matrix (nxm) that describes how the control u_t changes the state from t to $t\text{-}1$. Could be a zero matrix.

ε_t Random variable representing the process noise, assumed to be independent and normally distributed with covariance R_t

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad z = [z_1, \ldots, z_k]$$

…and we assume the measurement is a linear function of the state, with Gaussian noise…

$$\boxed{z_t = C_t x_t + \delta_t}$$ ⇦ Measurement model

$\boxed{C_t}$ Matrix (kxn) that describes how to map the state x_t to an observation z_t. Could be identity matrix

$\boxed{\delta_t}$ Random variable representing the measurement noise, assumed to be independent and normally distributed with covariance Q_t

…and we assume our previous estimate is normally distributed…

$$bel(x_{t-1}) = N\left(\mu_{t-1}, \Sigma_{t-1}\right)$$

…then the optimal prediction is given by the **Kalman Filter**:

Prediction: apply the process model to predict the new mean and covariance

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

Measurement **Innovation**: apply the measurement model to predict the measurement, and calculate how it differs from the actual measurement

$$\nu_t = z_t - C_t \bar{\mu}_t$$

Correction: correct the prediction of the new mean and covariance according to the measurement innovation, weighted by the 'Kalman gain'

$$\mu_t = \bar{\mu}_t + K_t \nu_t \qquad \Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$$

…to obtain an optimal new estimate of the state distribution:

$$bel(x_t) = N\left(\mu_t, \Sigma_t\right)$$

Kalman filter

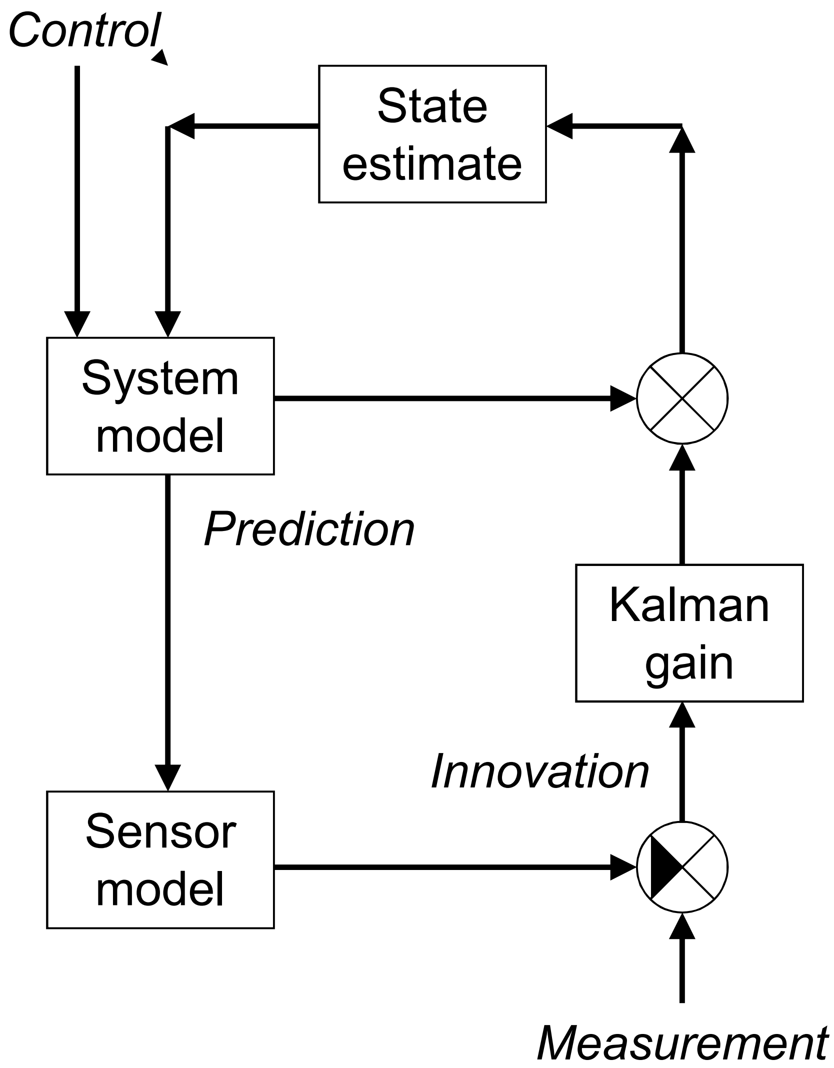
Kalman Gain K_t: calculate an appropriate weighting for the innovation, based on the predicted covariance and the measurement noise

$$K_t = \frac{\bar{\Sigma}_t C_t^T}{C_t \bar{\Sigma}_t C_t^T + Q_t}$$

Transforms innovation (difference of measured and predicted error) into state through inverse of sensor model

Weights the contribution of the innovation relative to the variance of the measurement

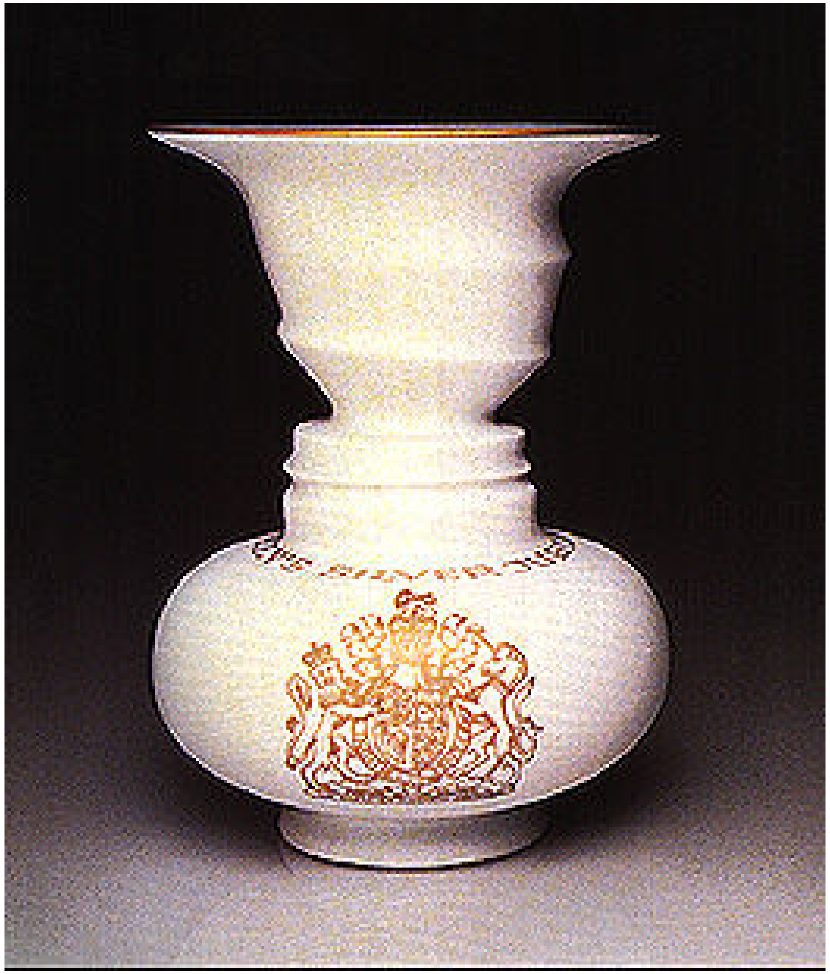
Kalman filter



- Can be derived from the Bayes filter (see extra handout online); this depends on the fact that linear transformation of a Gaussian distribution is also Gaussian
- Under assumptions of linearity and gaussian independent noise, this is the *optimal* estimator for the state
- Has many applications

Kalman filter

- Applied to human cognition by Grush (2004)
- *“The idea is that in addition to simply engaging with the body and environment, the brain constructs neural circuits that act as models of the body and environment. During overt sensorimotor engagement, these models are driven by efference copies in parallel with the body and environment, in order to provide expectations of the sensory feedback, and to enhance and process sensory information.”*
- E.g., what we see depends on what we expect to see.



Limitations of the Kalman filter

- What if the system and measurements are non-linear?
- This is almost always the case in robot applications.
- Some possible solutions:
  - take linear approximation around the current estimate to the non-linear functions (Extended Kalman filter)
  - represent distributions by random samples (e.g. Particle filter)

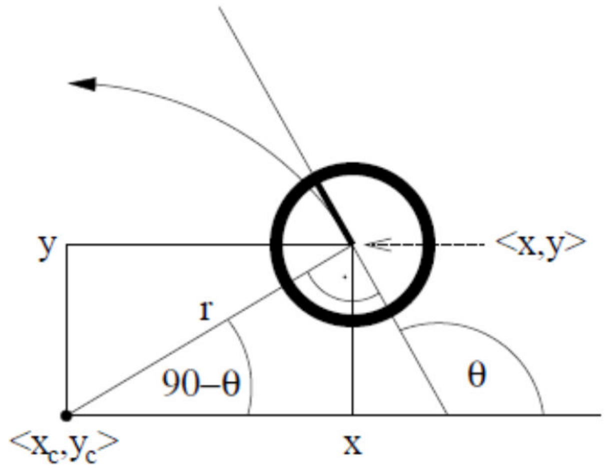
Extending the Kalman filter

- Most realistic robotic problems involve nonlinear functions

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

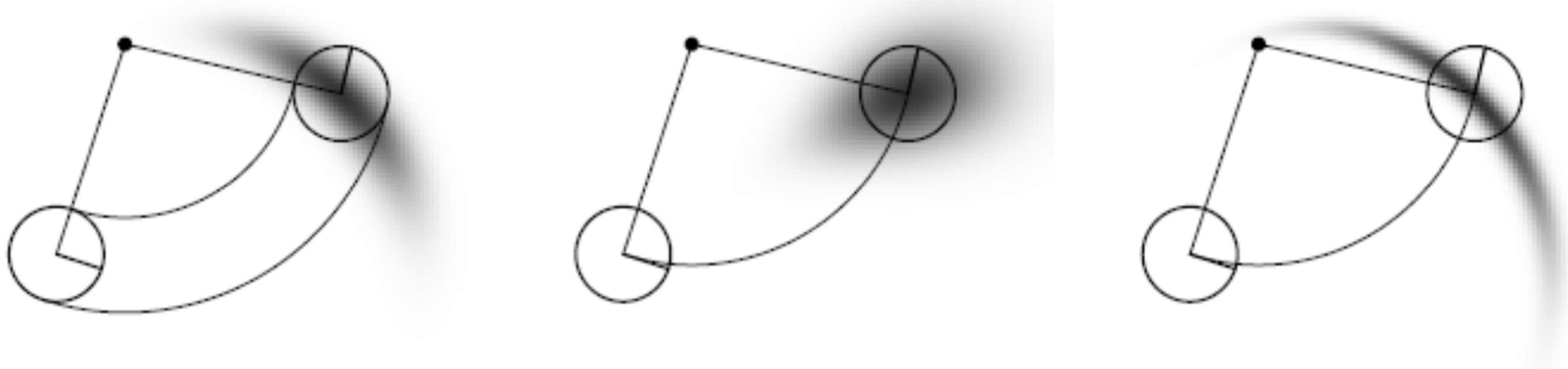
$$z_t = C_t x_t + \delta_t \quad \Longrightarrow \quad z_t = h(x_t)$$

E.g. 'simple' motion mode



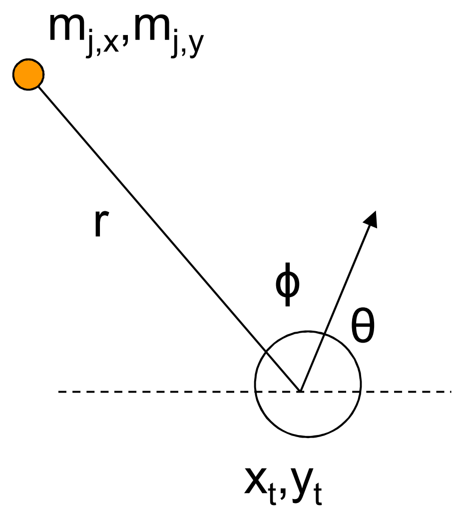
$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t}\cos\theta + \frac{v_t}{\omega_t}\cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix} + N(0, R_t)$$

Where the control action is a translational velocity v_t and a rotational velocity ω_t; these also have an associated error



E.g. 'Simple' sensor model: assume have identifiable landmark j, with 'signature' s_j at position $m_{j,x}$, $m_{j,y}$

$$\begin{pmatrix} r_t^i \\ \phi_t^i \\ s_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x}-x)^2+(m_{j,y}-y)^2} \\ a\tan 2(m_{j,y}-y, m_{j,x}-x)-\theta \\ s_j \end{pmatrix} + \begin{pmatrix} \varepsilon_{\alpha_r^2} \\ \varepsilon_{\alpha_\phi^2} \\ \varepsilon_{\alpha_s^2} \end{pmatrix}$$

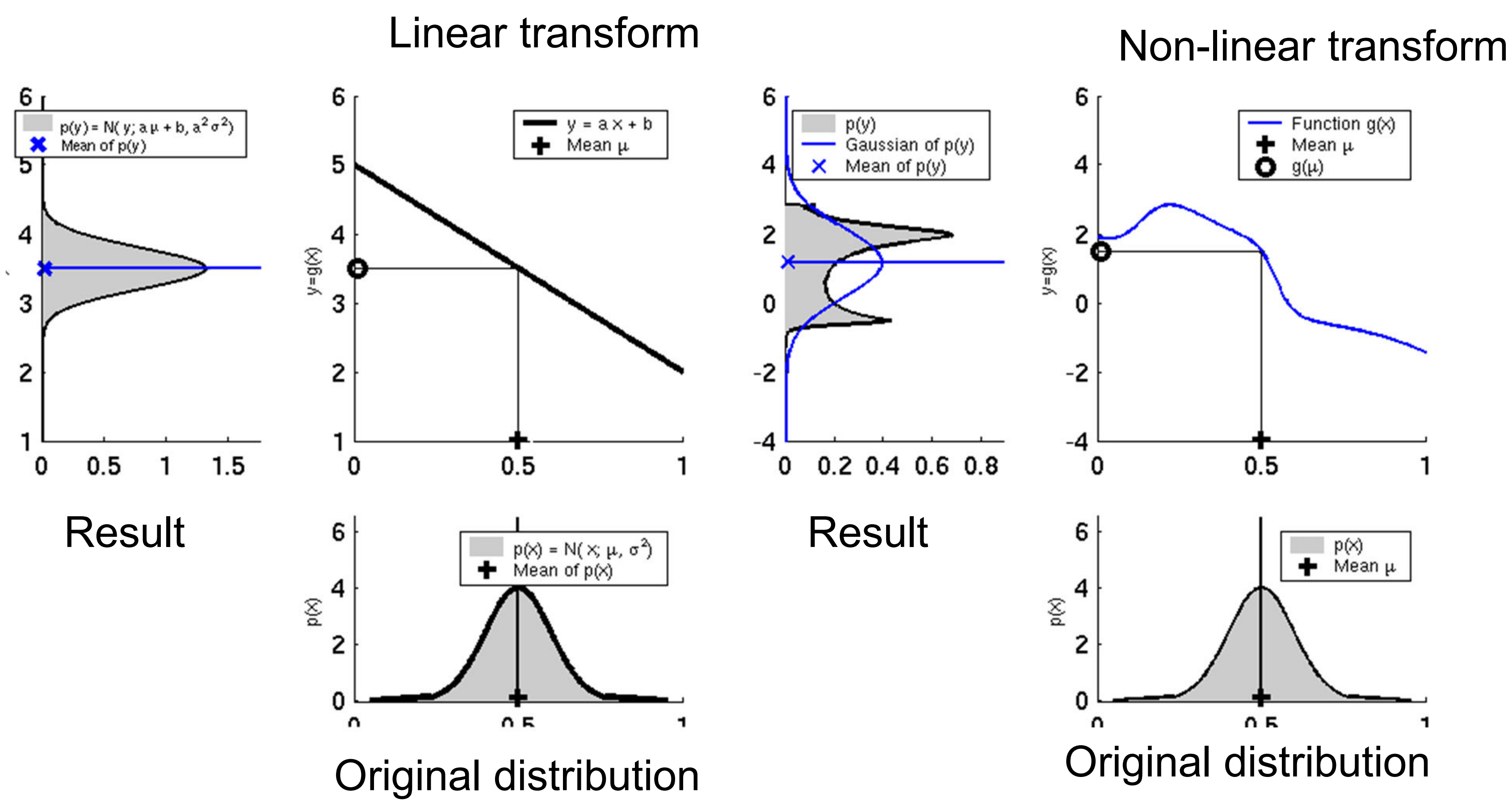


- Usually assume feature independence:

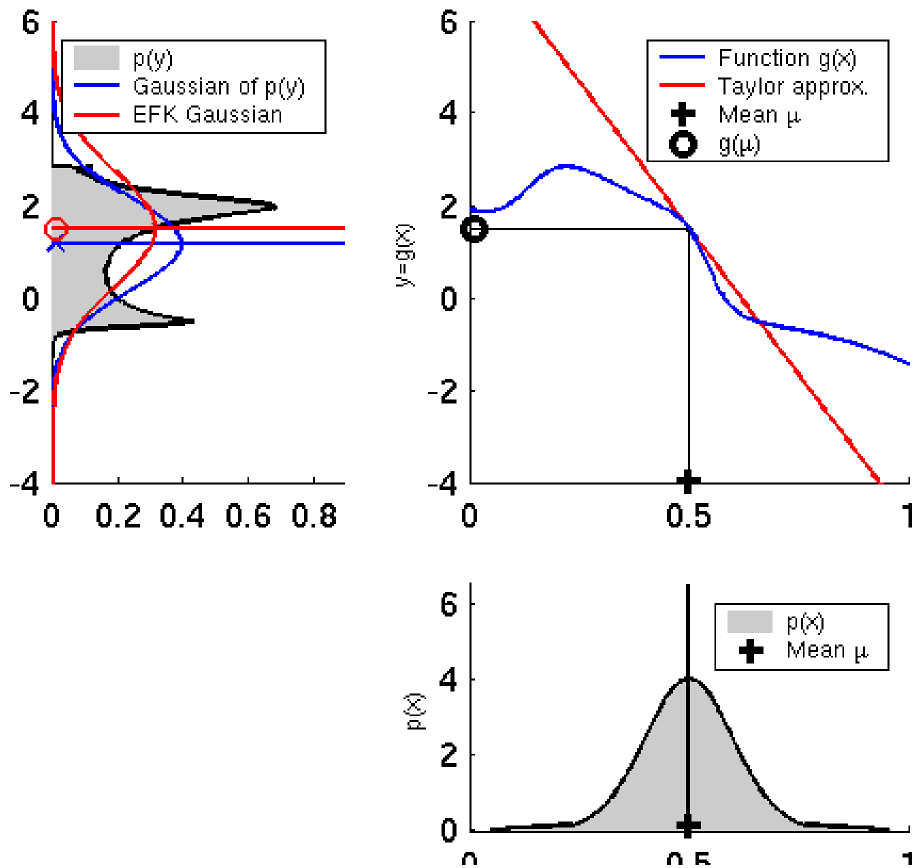
$$p(z_t \mid x_t, m) = \prod_i p(z_t^i \mid x_t, m)$$

- Effect is to allow incremental update, feature by feature
- See Thrun et al 2005 pp.204-210 for details of resulting EKF filters

The non-linear transform of a gaussian is not a gaussian



The Extended Kalman filter works by linearising the function around the current estimate



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:

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

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

EKF Algorithm

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

5. Correction:
6. $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$ $\longleftarrow$ $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
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

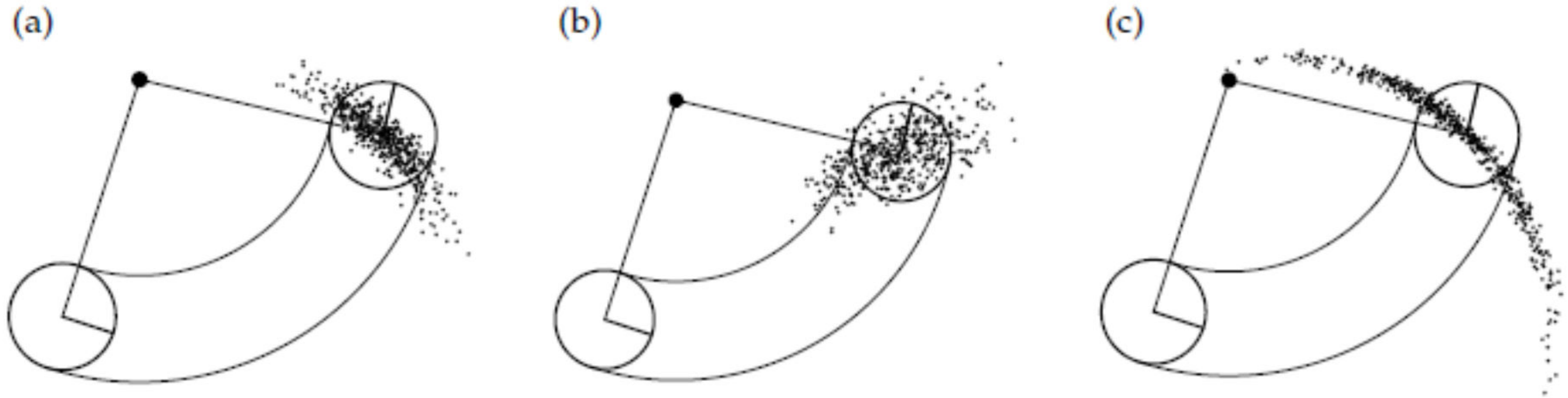
$$H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t} \qquad G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}$$

Landmark-based Localization

The Kalman Filter and EKF are just special case solutions of the Bayes filter:

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

An alternative non-parametric representation of the probability distribution *Bel*(x_t) : particles



http://www.cs.washington.edu/ai/Mobile_Robotics/

Particle filters

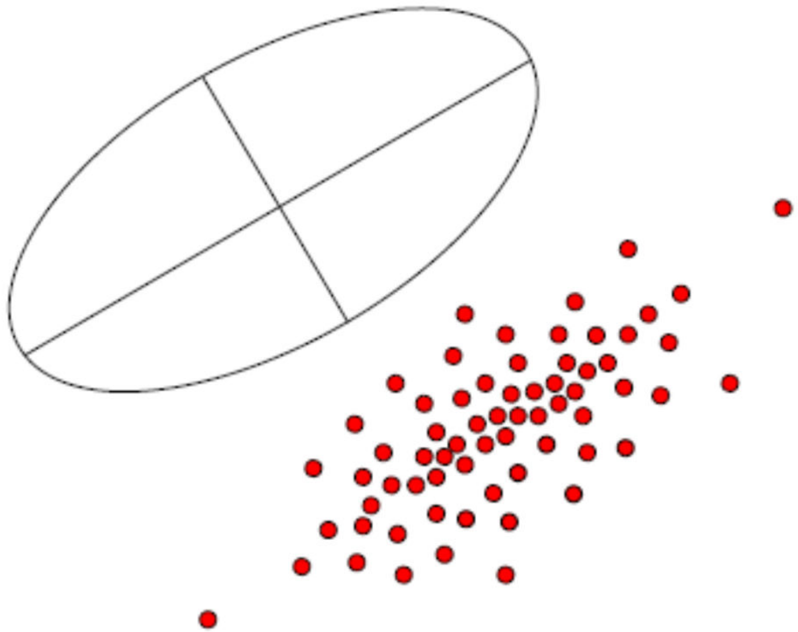
- Represent probability distribution as a set of discrete particles which occupy the state space

Particle =

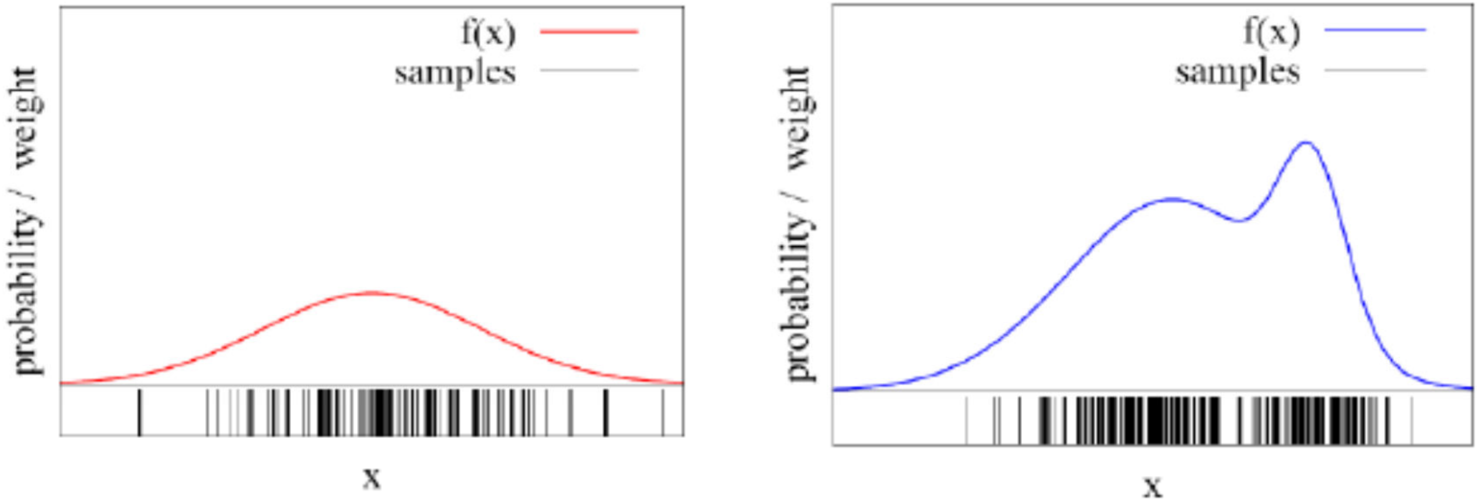
state hypothesis

Distribution =

set of state hypotheses



Particle sets can be used to approximate functions



The more particles fall into an interval, the higher the probability of that interval

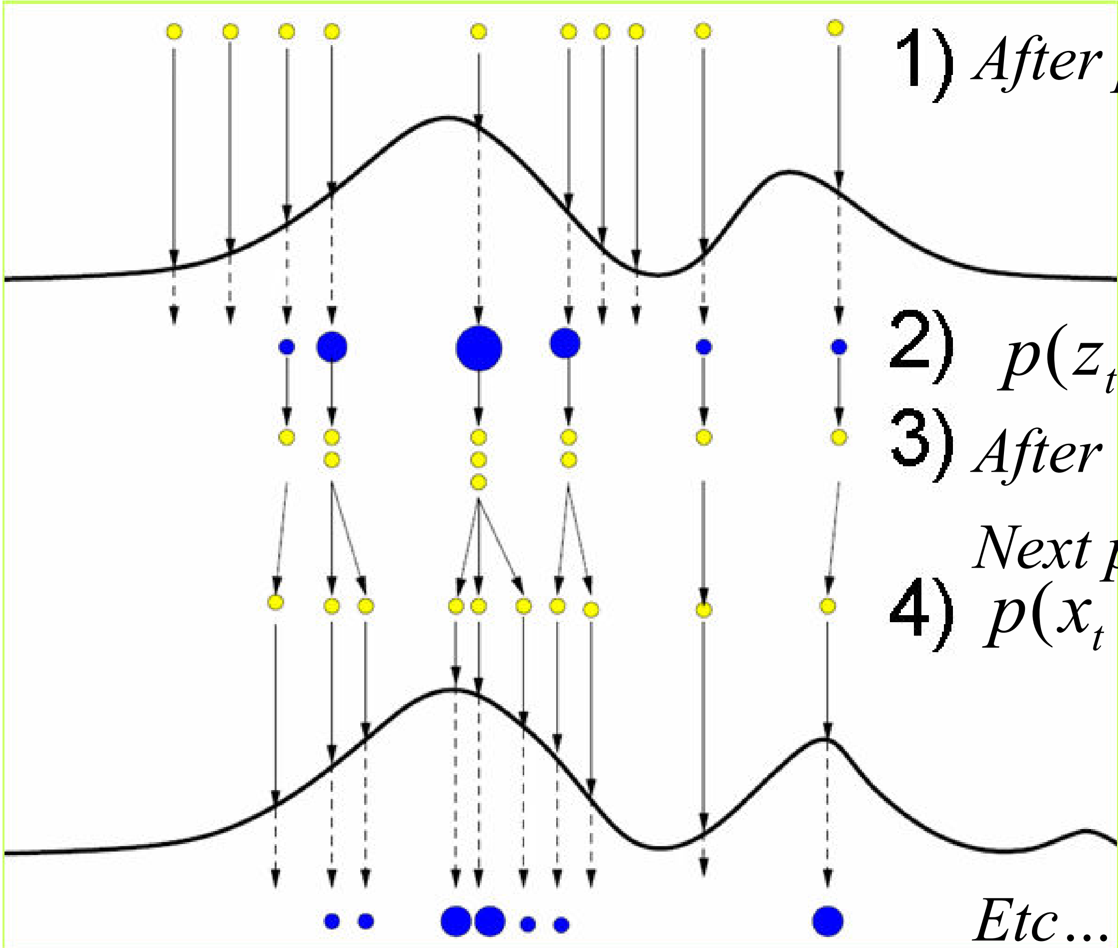
From Particle Set to Particle Filter:

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

- Initialize with M particles to represent *Bel(x_0)*
- Update cycle:
  – Prediction: for each particle, generate a new particle x_t^i drawn from $p(x_t \mid u_t, x_{t-1}^i)$
  – Correction: draw particles from this set (with replacement) with probability proportional to:

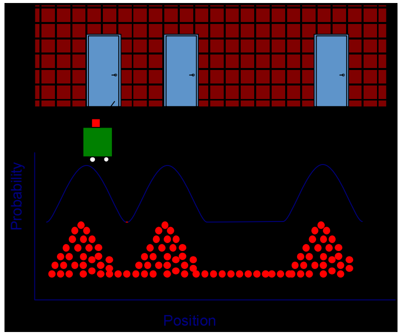
$$p(z_t \mid x_t^i)$$

Particle Filter Update Cycle: Visually



Particle Filter – Advantages/Disadvantages

- Can represent almost any probabilistic model, e.g. multi-modal distributions



- Relative easy to implement
- Can increase accuracy with computational resource

Number of particles grows exponentially with the dimensionality of the state space

1D – n particles

2D – n^2 particles

mD – n^m particles

References:

Sebastian Thrun, Wolfram Burgard and Dieter Fox, “Probabilistic Robotics”, MIT Press, Cambridge MA, 2005

Roland Siegwart & Illah Nourbakshsh “Introduction to Autonomous Robotics” MIT Press, Cambridge MA, 2011