

Master's thesis tutorial: part III

for the Autonomous Compliant Research group

Tinne De Laet, Wilm Decré, Diederik Verscheure

Katholieke Universiteit Leuven,
Department of Mechanical Engineering,
PMA Division

30 oktober 2006

Outline

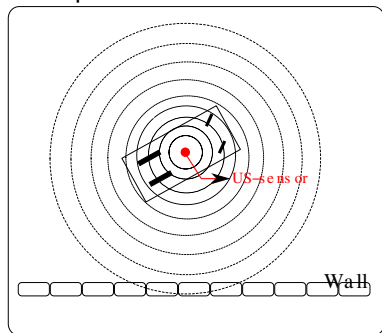
Outline

- 1 General
- 2 Basic concepts in probability
- 3 Recursive state estimation
- 4 Gaussian filters (Statistics-based methods)
- 5 Nonparametric methods (Sample-based filters)
- 6 Bayesian networks
- 7 BFL
- 8 On-line links
- 9 Further reading

Probabilistic state estimation

- Estimating state from sensor data
- State often not fully observable
- Sensor data corrupted by noise

Example:



Outline

- 1 General
- 2 Basic concepts in probability**
- 3 Recursive state estimation
- 4 Gaussian filters (Statistics-based methods)
- 5 Nonparametric methods (Sample-based filters)
- 6 Bayesian networks
- 7 BFL
- 8 On-line links
- 9 Further reading

Random variables and probability

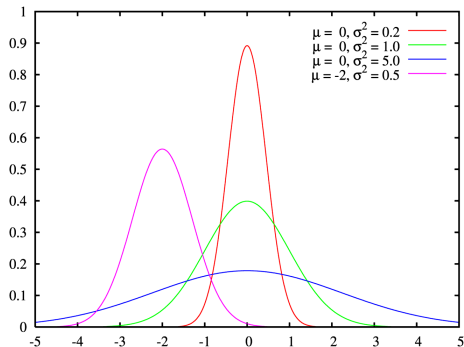
- Random variable X with value x
- Discrete case:
 - Probability: $p(X = x) = p(x)$
 - $\sum_x p(x) = 1$
- Continuous case:
 - Probability density function (PDF): $p(x)$
 - probability that $X \in (x, x + \delta x)$ equals $p(x)\delta x$ for $\delta x \rightarrow 0$
 - $\int_x p(x)dx = 1$

Remark: Also for vector variables \mathbf{X} and \mathbf{x} .

Gaussian

- Example: Gaussian with mean μ and variance σ^2

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



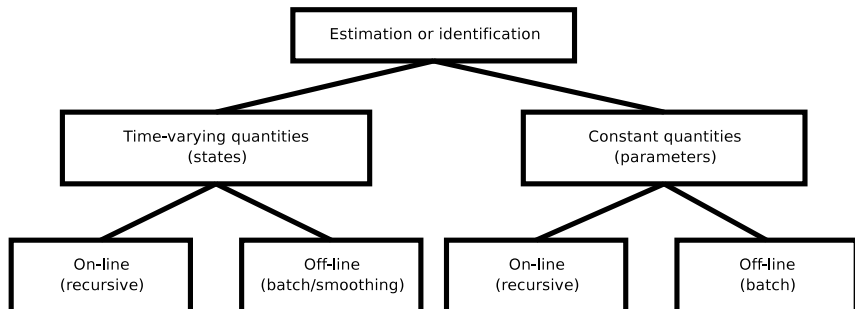
Probability distributions

- Joint distribution: $p(x, y) = p(X = x, Y = y)$
- Conditional probability: $p(x|y)$
- Two fundamental rules of probability:
 - Sum rule (*theorem of total probability, marginalization*):
 - Discrete case: $p(x) = \sum_y p(x|y)p(y) = \sum_y p(x, y)$
 - Continuous case: $p(x) = \int_y p(x|y) p(y) dy$
 - Product rule: $p(x, y) = p(x|y)p(y) = p(y|x)p(x)$
- Independence: $p(x, y) = p(x) p(y)$
- Conditional independence: $p(x, y|z) = p(x|z) p(y|z)$
- Bayes rule: $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$

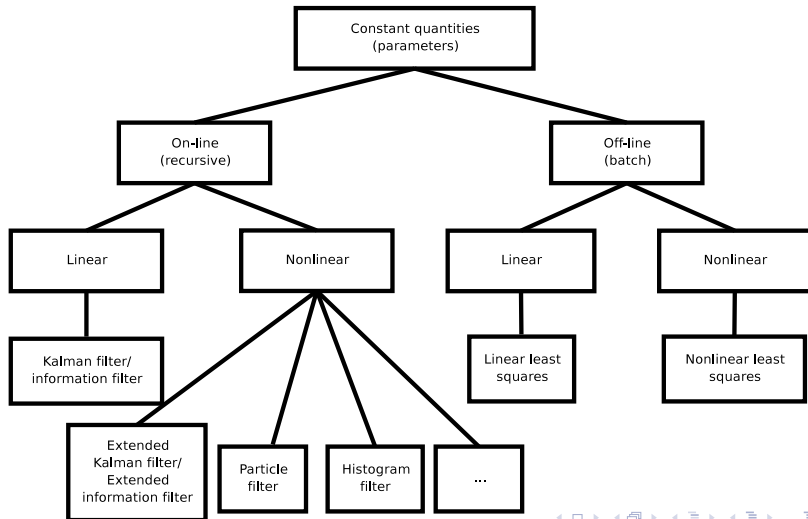
Terminology in estimation

- State x and data or measurements y
- *Prior probability distribution*: $p(x)$
- What we want to know is the *posterior probability distribution*: $p(x|y)$
 \Rightarrow Use of Bayes rule! $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$
- *Expectation* or *Expected value* of a random variable X :
 - Discrete case: $E[X] = \sum_x xp(x)$
 - Continuous case: $E[X] = \int_x xp(x) dx$
- *Variance* of a random variable X :
 $var[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2$
- *Covariance matrix* of two vector variables \mathbf{X} and \mathbf{Y} :
 $cov[\mathbf{X}, \mathbf{Y}] = E[(\mathbf{X} - E[\mathbf{X}])(\mathbf{Y} - E[\mathbf{Y}])^T]$
- Special case: $cov[\mathbf{X}, \mathbf{X}] = cov[\mathbf{X}]$

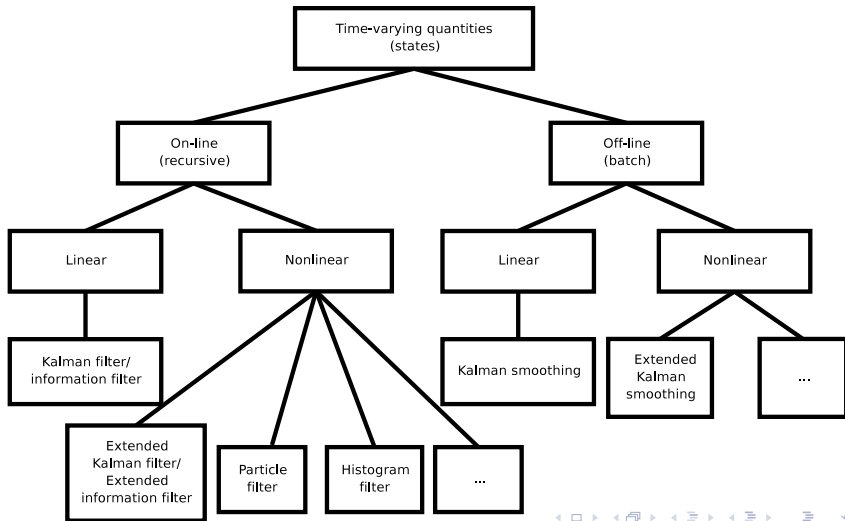
Estimation and identification



Parameter identification



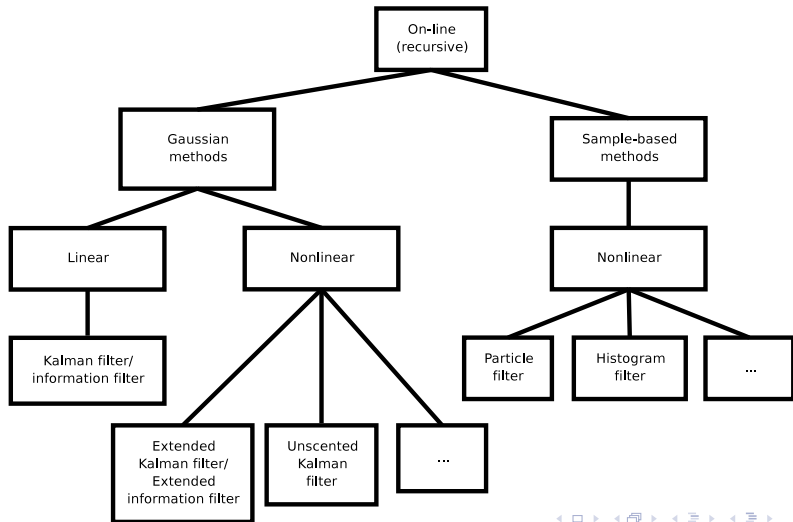
State estimation



The problem of on-line estimation

On-line estimation is generally less robust than off-line estimation, due to the fact that “statistics”, i.e. mean and covariance for the (Extended/Unscented) Kalman filter, or “samples”, i.e. particles for the Particle filter, are used to “summarize” the information gathered at a certain time step. Often, summary statistics or samples do not fully describe the gathered knowledge. Hence, information is thrown away at every time-step, which is not recovered afterwards.

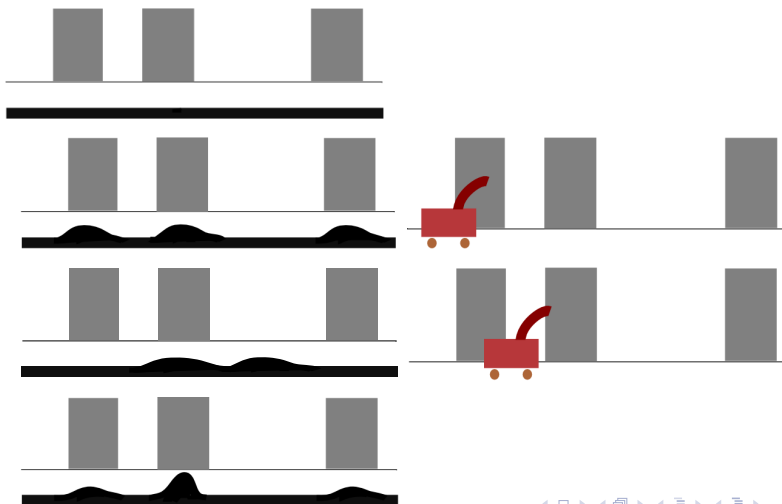
On-line state estimation



Outline

- 1 General
- 2 Basic concepts in probability
- 3 Recursive state estimation**
- 4 Gaussian filters (Statistics-based methods)
- 5 Nonparametric methods (Sample-based filters)
- 6 Bayesian networks
- 7 BFL
- 8 On-line links
- 9 Further reading

Recursive state estimation

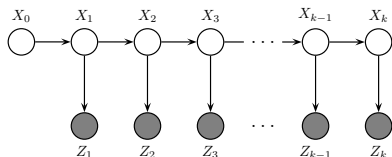


Recursive state estimation

- State x , measurements y , control u
- Subscript t denotes time-instance
- Goal for recursive estimation = posterior pdf

$$p(x_t | x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t | x_{t-1}, z_t, u_t) \text{ (Markov condition)}$$
- Two steps:
 - *state transition probability*: $p(x_t | x_{t-1}, u_t) \Rightarrow$ PREDICTION
 - *measurement probability*: $p(z_t | x_t) \Rightarrow$ CORRECTION

\Rightarrow *dynamic Bayesian network (DBN)*



Belief

- The belief reflects the robot's internal knowledge about the state of the environment *belief*: $bel(x_t) = p(x_t | z_{1:t}, u_{1:t})$
- The belief just before incorporating the latest measurement z_t , the prediction, is denoted as: $\bar{bel}(x_t) = p(x_t | z_{1:t-1}, u_{1:t})$

A general Bayes Filter Algorithm

Algorithm `Bayes_filter`($bel(x_{t-1}, u_t, z_t)$)

for all x_t do

$$\bar{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1} \Rightarrow \text{prediction}$$

$$bel(x_t) = \eta p(z_t | x_t) \bar{bel}(x_t) \Rightarrow \text{correction}$$

endfor

return $bel(x_t)$

Remark: Initial belief $bel(x_0)$ needed in first timestep

Outline

- 1 General
- 2 Basic concepts in probability
- 3 Recursive state estimation
- 4 Gaussian filters (Statistics-based methods)**
 - Kalman filter
 - Extended Kalman filter
 - Iterated extended Kalman filter
 - Unscented Kalman filter
 - Information filter
 - Extended information filter
 - Nonminimal state Kalman filter
- 5 Nonparametric methods (Sample-based filters)
- 6 Bayesian networks
- 7 BFL
- 8 On-line links
- 9 Further reading

Gaussian filters

- Earliest tractable implementations of the Bayes filter for continuous space.
- Most popular, despite shortcomings.

Basic idea

- Beliefs are represented by multivariate normal distributions.
⇒ Unimodal
- Characterized by two sets of parameters (*moments parametrization*): mean (μ) and covariance (Σ).
- Other parametrization possible (*canonical parametrization*) → see information filter

⇒ Poor match for any global estimation problems in which many distinct hypotheses exist!

Gaussian filters

Different type of Gaussian filters:

- Kalman filter (KF)
- Extended Kalman filter (EKF)
- Iterated Extended Kalman filter (IEKF)
- Unscented Kalman filter (UKF)
- Information filter

Assumptions

Assumptions

- Linear Gaussian system:
 - Linear state transition: $x_t = A_t x_{t-1} + B_t u_t + \epsilon_t$
 - Additive Gaussian noise ϵ_t
- Linear Gaussian measurement:
 - Linear measurement model: $z_t = H_t x_t + \delta_t$
 - Additive Gaussian noise δ_t
- Initial belief $bel(x_0)$ is Gaussian.

Remark

Comparable with least-squares solution with stochastic inspired weights.

Algorithm Kalman filter

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

$$\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \quad \rightarrow \text{PREDICTION}$$
$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$
$$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t (z_t - H_t \bar{\mu}_t) \rightarrow \text{CORRECTION}$$
$$\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$$

return μ_t, Σ_t

General

In practice rarely linear process and measurement model!
⇒ EKF relaxes the linearity assumption.

Assumptions

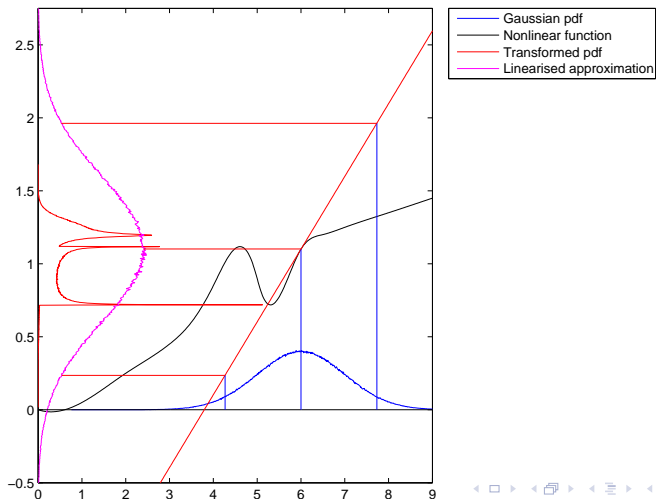
- Nonlinear Gaussian system:
 - Nonlinear state transition: $x_t = g(u_t, x_{t-1}) + \epsilon_t$
 - Additive Gaussian noise ϵ_t
- Nonlinear Gaussian measurement:
 - Nonlinear measurement model: $z_t = h(x_t) + \delta_t$
 - Additive Gaussian noise δ_t
- Initial belief $bel(x_0)$ is Gaussian.

Result: true belief no longer Gaussian.

Gaussian approximation

The extended Kalman filter calculates a *Gaussian approximation* to the true belief.

Linearization effect



Linearizations

EKF uses (first order) *Taylor approximation*. Linearization in the most likely state which is for Gaussians the mean of the posterior.

- Linearization of system model:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + \underbrace{g'(u_t, \mu_{t-1})}_{A_t} (x_{t-1} - \mu_{t-1}),$$

with $g'(u_t, x_{t-1}) = \frac{\partial g(u_t, x_{t-1})}{\partial x_{t-1}}$

- Linearization of measurement model:

$$h(x_t) \approx h(\bar{\mu}_t) + \underbrace{h'(\bar{\mu}_t)}_{H_t} (x_t - \bar{\mu}_t),$$

with $h'(x_t) = \frac{\partial h(x_t)}{\partial x_t}$

Algorithm extended Kalman filter

Algorithm extended Kalman filter ($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)

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

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

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

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

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

return μ_t, Σ_t

Very similar to the Kalman filter algorithm!

- Linear predictions in the KF are replaced by their nonlinear generalizations in EKF.
- EKF uses Jacobians instead of the linear system matrices in the case of KF.

Advantages and limitations

Advantages

- Simplicity and computational efficiency (unimodal presentation).
- If the nonlinear functions are approximately linear at the mean of the estimate and the covariance is small, the EKF performs well.

Limitation

The approximation of state transitions and measurements using linear Taylor expansion can be insufficient. The goodness of approximation depends on two main factors:

- the degree of uncertainty, and
- the degree of nonlinearity of the functions.

General

IEKF tries to do better than the EKF by linearization of the measurement model around the updated state estimate. This is achieved by iteration:

- First linearize around the predicted state estimate ($\bar{\mu}_t$) and do measurement update.
- Linearize the measurement model around the newly obtained estimate (μ_t^1) (where 1 stand for the first iteration).
- Iterate this process.

Algorithm Iterated extended Kalman filter

Algorithm iterated_extended_Kalman_filter ($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)

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

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

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

$$\mu_t^1 = \bar{\mu}_t + K_t^1 (z_t - h(\bar{\mu}_t))$$

$$\Sigma_t^1 = (I - K_t^1 h_t) \bar{\Sigma}_t$$

for $i = 1 : n$

$$H_t^i = \frac{\partial h(x_t)}{\partial \mu_t^{i-1}}$$

$$\eta^i = h(\mu_t^{i-1}) + H_t^i (\bar{\mu} - \mu_t^{i-1})$$

$$K_t = \bar{\Sigma}_t (H_t^i)^T \left((H_t^i) \bar{\Sigma}_t (H_t^i)^T \right)^{-1}$$

$$\mu_t^i = \bar{\mu}_t + K_t^i (\eta^i)$$

end

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

$$\mu_t = \mu_t^i$$

return μ_t, Σ_t

Advantages and limitations

Advantages

- Simplicity and computational efficiency (unimodal presentation).
- Outperforms the EKF in case of certain nonlinear measurement models.
- The IEKF is the best way to handle nonlinear measurement models that fully observe the part of the state that makes the measurement model non-linear.

Limitation

- Computationally more involved than extended Kalman filter.
- Uni-modal representation.

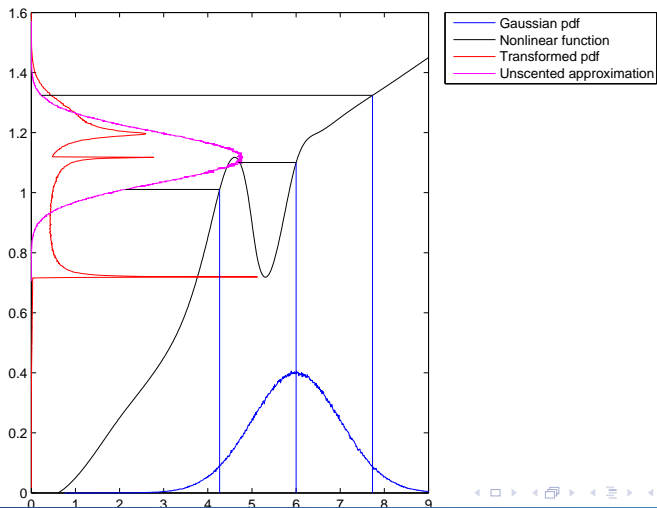
General

EKF is only one way to linearize the transformation of a Gaussian.

Unscented Kalman filter

UKF performs a stochastic linearization through the use of a weighted statistical linear regression process.

Illustration linearization



Procedure

Procedure

- Extract *sigma-points* from the Gaussian.
 - These points are located at the mean and symmetrically along the main axes of the covariance (two per dimension).
- Two weights associated with each sigma point (one for calculating mean, and one for covariance)
- Pass sigma-points through process model (g).
- The parameters (μ and Σ) are extracted from the mapped sigma points.

Algorithm unscented Kalman filter

Algorithm unscented Kalman filter ($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)

$$\chi_{t-1} = (\mu_{t-1} \quad \mu_{t-1} + \gamma\sqrt{\Sigma_{t-1}} \quad \mu_{t-1} - \gamma\sqrt{\Sigma_{t-1}})$$

$$\bar{\chi}_t^* = \mathbf{g}(u_t, \chi_{t-1})$$

$$\bar{\mu}_t = \sum_{i=0}^{2n} w_m^i \bar{\chi}_t^{*i}$$

$$\bar{\Sigma}_t = \sum_{i=0}^{2n} w_c^i (\bar{\chi}_t^{*i} - \bar{\mu}_t) (\bar{\chi}_t^{*i} - \bar{\mu}_t)^T + R_t$$

$$\bar{\chi}_t = (\bar{\mu}_t \quad \bar{\mu}_t + \gamma\sqrt{\bar{\Sigma}_t} \quad \bar{\mu}_t - \gamma\sqrt{\bar{\Sigma}_t})$$

$$\bar{Z}_t = h(\bar{\chi}_t)$$

$$\hat{z}_t = \sum_{i=0}^{2n} w_m^i \bar{Z}_t^i$$

$$S_t = \sum_{i=0}^{2n} w_c^i (\bar{Z}_t^i - \hat{z}_t) (\bar{Z}_t^i - \hat{z}_t)^T + Q_t$$

$$\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} w_c^i (\bar{\chi}_t^i - \bar{\mu}_t) (\bar{Z}_t^i - \hat{z}_t)^T$$

$$K_t = \bar{\Sigma}_t^{x,z} S_t^{-1}$$

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

$$\Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^T$$

return μ_t, Σ_t

Advantages and limitations

Advantages

- UKF more accurate than the first order Taylor series expansion by the EKF.
- The UKF performs better than EKF and IEKF for the process update (doesn't use only local information)
- No need to calculate derivatives of the functions (interesting when discontinuous, ...) → *Derivative free filter*

Limitation

- Slightly slower than extended Kalman filter.

Some remarks

Remark

Resemblance to the sample based representation used in particle filters (see next section).

- *Key difference:* sigma points are determined deterministically, while particle filters draw samples randomly.
- Therefore the UKF is more efficient than PF in the case the underlying distribution is approximately Gaussian.
- However if the belief is highly non-Gaussian the UKF's performance is low.

General

- Dual of the Kalman filter.
- Represents belief by Gaussian but in the *canonical parametrization*: information matrix and information vector.
- Same assumptions as Kalman filter
- Different update equations
→ what is computationally complex in one parametrization happens to be simple in the other (and vice versa.)

Canonical parametrization

- Information matrix (or precision matrix): $\Omega = \Sigma^{-1}$.
- Information vector: $\xi = \Sigma^{-1}\mu$

Algorithm information filter

Algorithm information filter ($\xi_{t-1}, \Omega_{t-1}, u_t, z_t$)

$$\bar{\Omega}_t = (A_t \Omega_{t-1}^{-1} A_t^T + R_t)^{-1}$$

$$\bar{\xi}_t = \bar{\Omega}_t (A_t \Omega_{t-1}^{-1} \xi_{t-1} + B_t u_t) \quad \rightarrow \text{PREDICTION}$$

$$\Omega_t = H_t^T Q_t^{-1} H_t + \bar{\Omega}_t$$

$$\xi_t = H_t^T Q_t^{-1} z_t + \bar{\xi}_t \quad \rightarrow \text{CORRECTION}$$

return ξ_t, Ω_t

- Computationally most involved step is prediction.
- In IF: measurement updates are additive. Even more efficient if measurements carry only information about a subset of all state variables at the time.
- In KF: process updates are additive. Even more efficient if only a subset of variables is affected by a control, or if variables evolve independently of each other.

General

Extends the IF to the nonlinear case (similar to EKF).

Algorithm extended_information_filter ($\xi_{t-1}, \Omega_{t-1}, u_t, z_t$)

$$\mu_{t-1} = \Omega_{t-1}^{-1} \xi_{t-1}$$

$$\bar{\Omega}_t = (A_t \Omega_{t-1}^{-1} A_t^T + R_t)^{-1}$$

$$\bar{\xi}_t = \bar{\Omega}_t g(u_t, \mu_{t-1}) \rightarrow \text{PREDICTION}$$

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

$$\Omega_t = \bar{\Omega}_t + H_t^T Q_t^{-1} H_t$$

$$\xi_t = \bar{\xi}_t + H_t^T Q_t^{-1} (z_t - h(\bar{\mu}_t) - H_t \bar{\mu}_t) \rightarrow \text{CORRECTION}$$

return ξ_t, Ω_t

Advantages and limitations

Advantages

- Easy to represent global uncertainty: $\Omega = \mathbf{0}$
- Tends to be numerically more stable than the Kalman filter in a lot of applications.
- Allows to integrate information without immediately resolving it into probabilities (interesting in case of large estimation problems). This can be done by adding new information locally to the system (an extension is necessary).
- Natural fit for multi-robot problems. (adding information (commutative))

Limitation

- The need to recover a state estimate in the update step is an important disadvantage. (inversion of information matrix).
- However, information matrix often exhibits sparse structure (they can be thought of as sparse graphs: *Markov Random Fields*).

Nonminimal state Kalman filter

- Transform the original state into a higher dimensional space where the measurement equations are linear
- It avoids the accumulation of linearization errors (EKF, IEKF, IF, EIF).
- The transformation is however not always possible.

Outline

- 1 General
- 2 Basic concepts in probability
- 3 Recursive state estimation
- 4 Gaussian filters (Statistics-based methods)
- 5 Nonparametric methods (Sample-based filters)**
 - Histogram filter
 - Particle filter
- 6 Bayesian networks
- 7 BFL
- 8 On-line links
- 9 Further reading

Sample-based filters

- Do not rely on a fixed functional form of the posterior (e.g. Gaussians).
- Approximation of the posteriors by a finite number of values (discretization of belief)
- Choice of the values:
 - *Histogram filters*: Decompose the state space into finitely many regions and represent the cumulative posterior for each region by a single probability value.
 - *Particle filters*: Represent the posteriors by finitely many samples.

Advantages and limitations

Advantages

- No assumptions on the posterior density, well-suited to represent complex multimodal beliefs.

Limitations

- High computational cost. → *Resource-adaptive algorithms*

Histogram filter(Grid-based methods)

Decomposes the state space into finitely many regions and represent the cumulative posterior for each region by a single probability value.

- *Discrete Bayes filters*: finite spaces.
- *Histogram filters*: continuous spaces.

Discrete Bayes filter

- Random variable X_t can take finitely many values.

Algorithm discrete_Bayes_filter ($\{p_{k,t-1}\}, u_t, z_t$)

for all k do

$$\bar{p}_{k,t} = \sum_i p(X_t = x_k | u_t, X_{t-1} = x_i) p_{i,t-1} \rightarrow \text{PREDICTION}$$

$$p_{k,t} = \eta p(z_t | X_t = x_k) \bar{p}_{k,t} \rightarrow \text{CORRECTION}$$

endfor

return $\{p_{k,t}\}$

Histogram filter

- Approximate inference tool for continuous state spaces.
- The continuous space is decomposed into finitely many (K) *bins* or *regions*:

$$\text{dom}(X_t) = x_{1,t} \cup x_{2,t} \cup \dots \cup x_{K,t}.$$

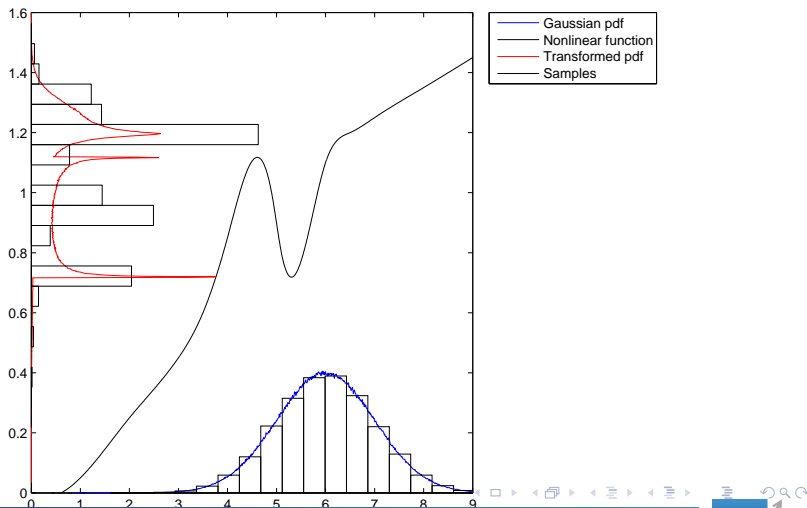
Trade off between accuracy and computational burden.

- The posterior becomes a piecewise constant PDF, which assigns a uniform probability to each state x_t within each region $x_{k,t}$:

$$p(x_t) = \frac{p_{k,t}}{|x_{k,t}|},$$

with $|x_{k,t}|$ the volume of the region $x_{k,t}$.

Illustration



Histogram filter

Decomposition techniques:

- *Static*: fixed decomposition, chosen in advance, irrespective of the shape of the posterior which is being approximated.
 - Easier to implement
 - Possibly wasteful with regards to computational resources.
- *Dynamic*: adapt the decomposition to the specific shape of the posterior distribution. The less likely a region, the coarser the decomposition.
 - More difficult to implement
 - Ability to make better use of computational resources.

An similar effect is obtained by *selective updating*

General (Sequential Monte Carlo methods)

- Nonparametric implementation of the Bayes filter.
- Approximation of the posterior by a finite number of values.
- These values are randomly drawn from the posterior distribution → *samples*:

$$X_t := x_t^1, x_t^2, \dots, x_t^M,$$

with M the number of particles (often large, e.g. $M = 1000$ for each dimension)

- The likelihood for a state hypothesis x_t to be included in the particle set X_t would ideally be proportional to its posterior belief:

$$x_t^i \sim p(x_t | z_{1:t}, u_{1:t})$$

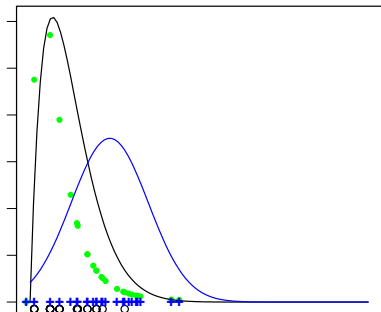
General

- The likelihood for a state hypothesis x_t to be included in the particle set X_t would ideally be proportional to its posterior belief:

$$x_t^i \sim p(x_t | z_{1:t}, u_{1:t})$$

- This posterior belief is however unknown, since this is what we want to calculate.
- Therefore we have to sample from an approximate distribution
 \Rightarrow *Importance sampling*

Importance sampling



Algorithm importance sampling

Require: $M \gg N$

for $j = 1$ to M do

 Sample $\tilde{x}_j \sim q(x)$

$$w_j = \frac{p(\tilde{x}_j)}{q(\tilde{x}_j)}$$

endfor

for $i = 1$ to N do

 Sample $x_i \sim (\tilde{x}_j, w_j) \quad 1 < j < M$

endfor

Particle filter

- Particle filter recursively constructs the particle set X_t from the set X_{t-1}

Problem

A problem concerning particle filtering is the *degeneracy problem*, *particle deprivation/depletion* or *sample impoverishment*, after a few iterations all but one particles will have negligible weight. To reduce this effect

- Good choice of importance density function,
- Use of resampling.

Particle filter

Algorithm Particle_filter (X_{t-1}, u_t, z_t)

$\bar{X}_t = X_t = \text{empty}$

for $m = 1$ to M do

 sample $x_t^m \sim p(x_t | u_t, x_{t-1}^m)$

$w_t^m = p(z_t | x_t^m)$

$\bar{X}_t = \bar{X}_t + \langle x_t^m, w_t^m \rangle$

endfor

for $m = 1$ to M do

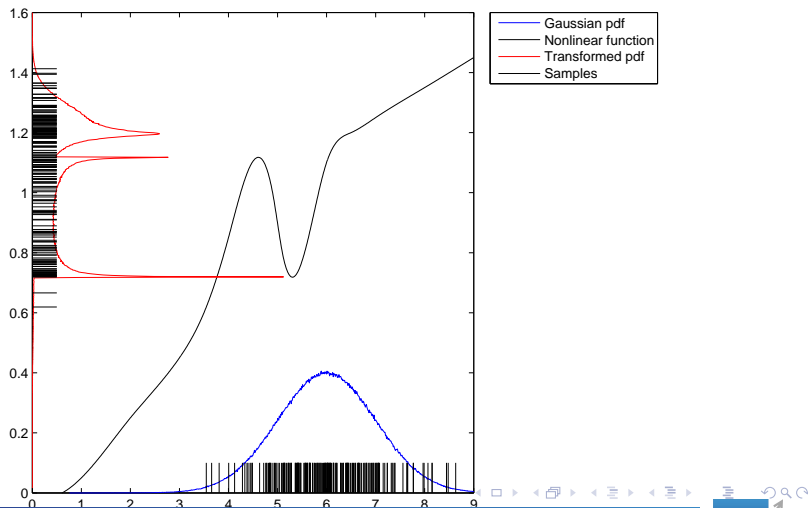
 draw i with probability $\propto w_t^i$

 add x_t^i to X_t

endfor

return X_t

Particle filter



Particle filter

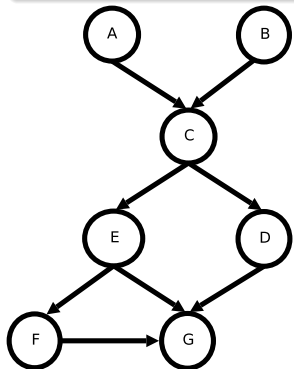
- A lot of different variants on the particle filtering (how particle deprivation is handled, variable number of particles, ...).
- How many samples should be used?

Outline

- 1 General
- 2 Basic concepts in probability
- 3 Recursive state estimation
- 4 Gaussian filters (Statistics-based methods)
- 5 Nonparametric methods (Sample-based filters)
- 6 Bayesian networks**
- 7 BFL
- 8 On-line links
- 9 Further reading

General

Bayesian networks are graphical structures for representing the probabilistic relationships among a large number of variables and for doing probabilistic inference with those variables.



Definition

Bayesian network

A Bayesian network consists of the following:

- A set of *variables* and a set of *directed edges* between variables.
- Each variable has a finite set of mutually exclusive states.
- The variables together with the directed edges form a *directed acyclic graph* (DAG).
- To each variable A with parents B_1, \dots, B_n , there is attached the potential table $P(A|B_1, \dots, B_n)$

Outline

- 1 General
- 2 Basic concepts in probability
- 3 Recursive state estimation
- 4 Gaussian filters (Statistics-based methods)
- 5 Nonparametric methods (Sample-based filters)
- 6 Bayesian networks
- 7 BFL**
- 8 On-line links
- 9 Further reading

Bayesian Filtering Library (BFL)

- Open source project (C++), started by Klaas Gadeyne
- State estimation software framework/library BFL: support for different filters (in particular particle filters and Kalman filters, but also e.g. grid based methods) and easily extensible towards other Bayesian methods.

Bayesian Filtering Library (BFL)

What is BFL?

- *Bayesian*: fully Bayesian software framework. Different Bayesian algorithms with maximum of code reuse. Easy comparison of performance of different algorithms.
- *Open*: Potential for maximum reuse of code and study algorithms
- *Independent*: BFL is decoupled possible from one particular numerical/stochastic library. Furthermore BFL is independent of a particular application. This means both its interface and implementation are decoupled from particular sensors, assumptions, algorithms, ... that are specific to a certain application.

Bayesian Filtering Library (BFL)

Getting support - the BFL community

There are different ways to get some help/support:

- A BFL-tutorial: <http://people.mech.kuleuven.be/~tdelaet/tutorialBFL.pdf>.
- The website:
<http://people.mech.kuleuven.be/~kgadeyne/bfl.html>
(also source code).
- Klaas Gadeyne's PhD thesis (see website).
- The mailing list (see website).

An example: mobile robot tracking

- Mobile robot deadreckoning
- Mobile robot with measurements

Outline

- 1 General
- 2 Basic concepts in probability
- 3 Recursive state estimation
- 4 Gaussian filters (Statistics-based methods)
- 5 Nonparametric methods (Sample-based filters)
- 6 Bayesian networks
- 7 BFL
- 8 On-line links**
- 9 Further reading

On line links

Estimation links:

- Wikipedia: http://en.wikipedia.org/wiki/Recursive_Bayesian_estimation,
http://en.wikipedia.org/wiki/Kalman_filter and
http://en.wikipedia.org/wiki/Particle_filter.
- BFL (Bayesian Filtering Library): <http://people.mech.kuleuven.be/~kgadeyne/bfl.html>.
- BNT (Bayes Net Toolbox):
<http://bnt.sourceforge.net/>.
- Sequential Monte Carlo Methods homepage:
<http://www-sigproc.eng.cam.ac.uk/smc/index.html>

Outline

- 1 General
- 2 Basic concepts in probability
- 3 Recursive state estimation
- 4 Gaussian filters (Statistics-based methods)
- 5 Nonparametric methods (Sample-based filters)
- 6 Bayesian networks
- 7 BFL
- 8 On-line links
- 9 Further reading

Further reading

Kalman filtering:

- Kalman filters: a tutorial (<http://people.mech.kuleuven.be/~tdelaet/journalA99.pdf>)
- Nonminimal state Kalman Filter: doctorate Tine Lefebvre, Contact modelling, parameter identification and task planning for autonomous compliant motion using elementary contacts, Dept. Mechanical Engineering KUL.

Particle filtering:

- A Particle Filter Tutorial for Mobile Robot Localization, I.M. Rekleitis (<http://www.cim.mcgill.ca/~yiannis/particletutorial.pdf>)
- Sequential Monte Carlo Methods in Practice, A. Doucet et al., Springer, 2001

Further reading

Bayesian networks:

- Bayesian Networks and Decision Graphs, F.V. Jensen, Springer, 2001
- Learning Bayesian Networks, R.E. Neapolitan, Prentice Hall, 2004
- Bayesian Nets and Causality, J. Williamson, Oxford University Press, 2005

Presentation and article version

- This presentation is available online: <http://people.mech.kuleuven.be/~tdelaet/estimation/part3.pdf>.
- An article version of the presentation including extra comments and explanations is available online: <http://people.mech.kuleuven.be/~tdelaet/estimation/article3.pdf>.

Bibliography