Nonlinear observers

design techniques, part 1b

Erik Frisk

Dept. of Electrical Engineering, Linköping University

December 20, 2018

Today's lecture and reading material

Today's lecture covers particle filters and recalls some basics from last lecture.

1. Particle filter [12,13]

Extended Kalman Filter

Measurement update

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - h(\hat{x}_{t|t-1}))$$
$$K_t = P_{t|t-1}H_t^T(H_tP_{t|t-1}H_t^T + R_t)^{-1}$$
$$P_{t|t} = P_{t|t-1} - K_tH_tP_{t|t-1}$$

$$H_t = \frac{\partial}{\partial x}h(x)|_{x=\hat{x}_{t|t-1}}$$

Time update

$$\hat{x}_{t+1|t} = f(\hat{x}_{t|t}, 0)$$
$$P_{t+1|t} = F_tP_{t|t}F_t^T + G_tQ_tG_t^T$$

$$F_t = \frac{\partial}{\partial x}f(x,w)|_{x=\hat{x}_{t|t},w=0}$$
$$G_t = \frac{\partial}{\partial w}f(x,w)|_{x=\hat{x}_{t|t},w=0}$$

UKF

Measurement update

From last lecture, we had

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + P^{xy}_{t|t-1}P^{-yy}_{t|t-1}(y_t - \hat{y}_t)$$
$$P_{t|t} = P_{t|t-1} - P^{xy}_{t|t-1}P^{-yy}_{t|t-1}P^{xyT}_{t|t-1}$$

where

$$\hat{y}_t = \sum_i w^{(i)}y_t^{(i)}, \quad y_t^{(i)} = h(x^{(i)}_{t|t-1}, e_t^{(i)})$$
$$P^{yy}_{t|t-1} = \sum_i w^{(i)}(y_t^{(i)} - \hat{y}_t)(y_t^{(i)} - \hat{y}_t)^T$$
$$P^{xy}_{t|t-1} = \sum_i w^{(i)}(x^{(i)}_{t|t-1} - \hat{x}_{t|t-1})(y_t^{(i)} - \hat{y}_t)^T$$

Time update is done in a corresponding way, see Julier/Uhlmann or G. Hendeby.

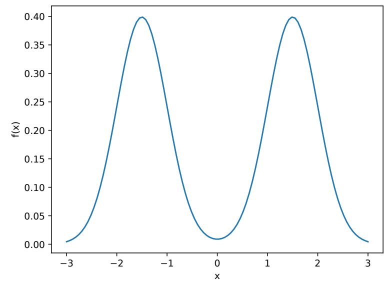
Outline

- *Nonlinear filtering*
- *Particle representation of distributions*
- *Time update*
- *Measurement update*
- *Resampling*
- *Summary of algorithm*
- *Exercises*

Particle Filter

Objectives

- approximative method for non-linear filtering
- A bayesian approach (note that I did not use Bayesian arguments for the EKF/UKF)
- simple basic principle, but requires caution for it to work well
- applicable to sytrongly non-linear systems, where EKF and linearizing megthods fail
- typical example, multimodal distributions, (coming from $y = x^2$).



Nonlinear bayesian filtering

With a basic model described by

$$\begin{aligned} x_{t+1} &= f(x_t, \epsilon_t) \\ y_t &= h(x_t) + e_t \end{aligned}$$

where ϵ_t and e_t are white independent sequences with known (arbitrary) distributions.
In Bayesian filtering, the estimate are represented by their distributions (i.e. x is a random variable)

$$p(x_t|\mathcal{Y}^t), \quad \mathcal{Y}^t = (y_1, \dots, y_t)$$

Then the estimate could, e.g., be computed as

$$\hat{x}_{t|t} = \mathbb{E}(x_t|\mathcal{Y}^t) = \int x_t p(x_t|\mathcal{Y}^t)\, dx_t$$

or even the maximum likelihood estimate.

Ideal measurement and time update

Measurement update

$$(\hat{x}_{t|t-1}, P_{t|t-1}) \to (\hat{x}_{t|t}, P_{t|t}) \quad \sim \quad p(x_t|\mathcal{Y}^{t-1}) \to p(x_t|\mathcal{Y}^t)$$

The measurement-update is given by a direct application of Bayes' rule

$$p(x_t|\mathcal{Y}^t) = \cdots = \frac{p(y_t|x_t)}{p(y_t|\mathcal{Y}^{t-1})} p(x_t|\mathcal{Y}^{t-1})$$

Time update

$$(\hat{x}_{t|t}, P_{t|t}) \to (\hat{x}_{t+1|t}, P_{t+1|t}) \quad \sim \quad p(x_t|\mathcal{Y}^t) \to p(x_{t+1}|\mathcal{Y}^t)$$

$$p(x_{t+1}|\mathcal{Y}^t) = \int p(x_{t+1}|x_t, \mathcal{Y}^t) p(x_t|\mathcal{Y}^t)\, dx_t = \int p(x_{t+1}|x_t) p(x_t|\mathcal{Y}^t)\, dx_t$$

Basic message

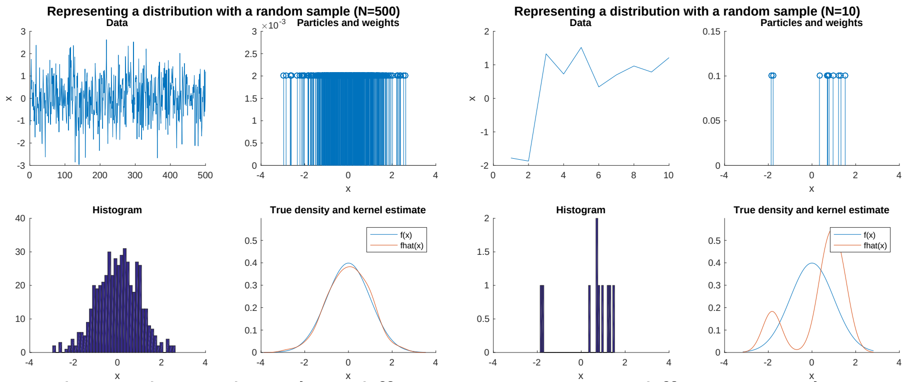
This quantities can not be computed exactly except for special cases (linear gaussian case gives Kalman Filter)

Outline

- *Nonlinear filtering*
- *Particle representation of distributions*
- *Time update*
- *Measurement update*
- *Resampling*
- *Summary of algorithm*
- *Exercises*

Basic tool: Representing a distribution with samples

500 samples/particles | 10 samples/particles



Weight can be used to place different importance to different particles (more later)

Particle representation of distribution

Basic tool: approximate a probability density with a number of weighted samples from the distribution.
With samples and associated weights

$$x^{(i)} \sim p(x), \quad i = 1, \ldots, N$$
$$w^{(i)} \quad \text{(possibly } 1/N\text{)}$$

then

$$p(x) \approx \sum_i \delta(x - x^{(i)}) w^{(i)}$$

With such an approximation, it is straightforward to estimate any statistics. For example

$$\hat{\mu}_x = \sum_i x^{(i)} w^{(i)}$$
$$\hat{P}_x = \sum_i w^{(i)} (x^{(i)} - \hat{\mu}_x)(x^{(i)} - \hat{\mu}_x)^T$$

Particle approximation of $p(x)$

As long as you can draw samples from a distribution, you can represent it (but $\mathbb{R}^n$ can be a big space)

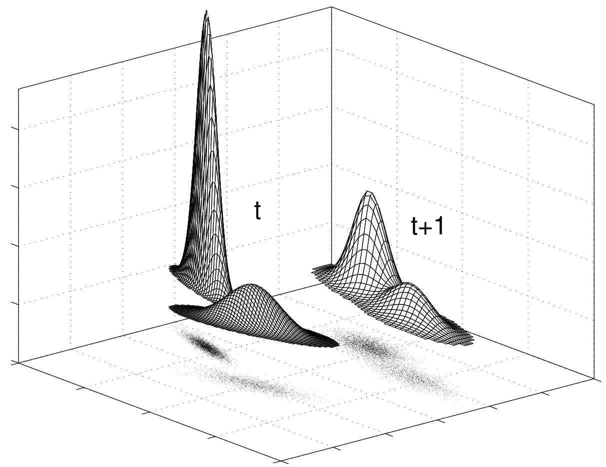


Figure borrowed from G. Hendeby

Outline

- Nonlinear filtering
- Particle representation of distributions
- **Time update**
- Measurement update
- Resampling
- Summary of algorithm
- Exercises

Partikelfilter - tidsuppdatering

As before, we want to update the state estimate distribution from t to $t+1$, i.e.,

$$p(x_{t|t}) \Rightarrow p(x_{t+1|t})$$

and since we have a particle representation,

$$(x_{t|t}^{(i)}, w_{t|t}^{(i)}) \Rightarrow (x_{t+1|t}^{(i)}, w_{t+1|t}^{(i)}), \quad i = 1, \dots, N$$

under the dynamic equation

$$x_{t+1} = f(x_t, \epsilon_t), \quad \epsilon_t \sim p_\epsilon(\epsilon_t)$$

where $f(\cdot)$ is a non-linear function.

The process noise ϵ_t is distributed as $p_\epsilon(\epsilon_t)$. It is assumed that we can draw samples from this distribution. This could be a gaussian, but there is no restriction really.

Partikelfilter - tidsuppdatering

A simple, suboptimal, way is to draw N random samples from $p_\epsilon(\epsilon_t)$

$$\epsilon_t^{(i)}, \quad i = 1, \dots, N$$

and insert into the dynamic equation

$$x_{t+1|t}^{(i)} = f(x_{t|t}^{(i)}, \epsilon_t^{(i)}) \quad i = 1, \dots, N$$

With this sinple strategy, there is no reason to change the weights/importance, of each particle and thus

$$w_{t+1|t}^{(i)} = w_{t|t}^{(i)}$$

representing the distribution $p(x_{t+1}|\mathcal{Y}^t)$. The estimate of the mean aposteriori is then

$$\hat{x}_{t+1|t} = \sum_{i=1}^{N} w_{t+1|t}^{(i)} x_{t+1|t}^{(i)}$$

Outline

- Nonlinear filtering
- Particle representation of distributions
- Time update
- **Measurement update**
- Resampling
- Summary of algorithm
- Exercises

Measurement update

As before, we want to update the state estimate distribution from $t|t-1$ to $t|t$ using the information in a measurement y_t, i.e.,

$$p(x_{t|t-1}) \overset{y_t}{\Rightarrow} p(x_{t|t})$$

and since we have a particle representation,

$$(x_{t|t-1}^{(i)}, w_{t|t-1}^{(i)}) \overset{y_t}{\Rightarrow} (x_{t|t}^{(i)}, w_{t|t}^{(i)}), \quad i = 1, \dots, N$$

under the measurement equation

$$y_t = h(x_t) + e_t \quad e_t \sim p_e(e_t)$$

where $h(\cdot)$ is a non-linear function.

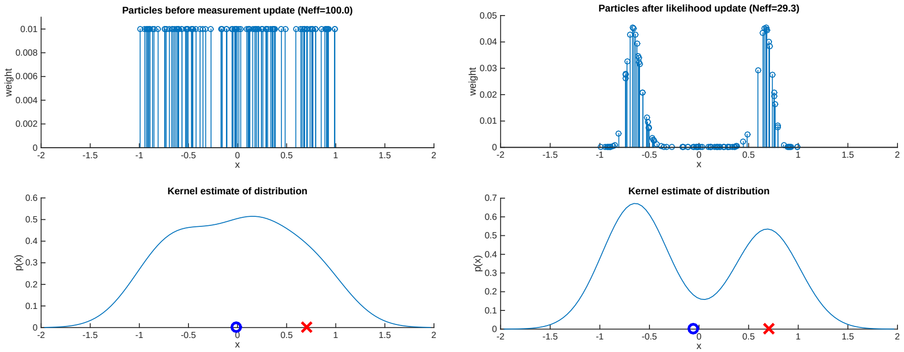
The measurement noise e_t is distributed as $p_e(e_t)$. It is assumed that we can evaluate the density function. This could be a gaussian, but there is no restriction really.

Measurement update - illustration

Consider the measurement equation

$$y = x^2 + e, \quad e \sim \mathcal{N}(0, \sigma^2)$$

where the true $x = \sqrt{0.5}$.
Assume we know that the true value lies in $[-1, 1]$, let the particles be drawn from $\mathcal{U}(-1, 1)$. After the measurement update we get:



Measurement update

We have an particle approximation for $p(x_t|\mathcal{Y}^{t-1})$ and the measurement equation

$$y_t = h(x_t) + e_t, \quad e_t \sim p_e(e_t)$$

and a new measurement y_t is obtained.

> ***Measurement update***
>
> Update weights/importance of each particle according to how likely the new output y_t is for each particle.

$$\begin{aligned} p(x_t|\mathcal{Y}^t) = p(x_t|y_t, \mathcal{Y}^{t-1}) &= \frac{p(y_t|x_t, \mathcal{Y}^{t-1})\, p(x_t|\mathcal{Y}^{t-1})}{p(y_t|\mathcal{Y}^{t-1})} \\ &= \frac{p(y_t|x_t)\, p(x_t|\mathcal{Y}^{t-1})}{p(y_t|\mathcal{Y}^{t-1})} \propto p(y_t|x_t)\, p(x_t|\mathcal{Y}^{t-1}) \end{aligned}$$

Measurement update

Then, for each particle we have

$$p(x_{t|t}^{(i)}|\mathcal{Y}^t) \propto p(y_t|x_{t|t-1}^{(i)})\, p(x_{t|t-1}^{(i)}|\mathcal{Y}^{t-1}), \quad i = 1, \dots, N$$

and with the measurement equation

$$y_t = h(x_t) + e_t, \quad e_t \sim p_e(e_t)$$

we have

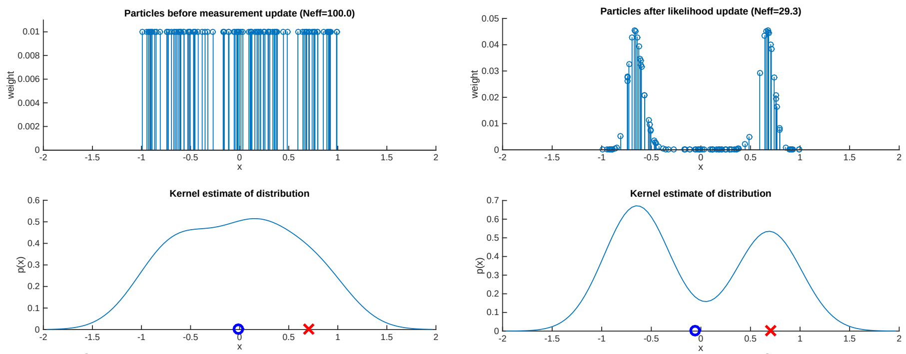
$$p(y_t|x_{t|t-1}^{(i)}) = p_e(y_t - h(x_{t|t-1}^{(i)})), \quad p(x_t|\mathcal{Y}^{t-1}) = w_{t|t-1}$$

Thus, the weights are updated according to $w_{t|t}^{(i)} \propto p(x_t^{(i)}|\mathcal{Y}^t)$, i.e.,

$$w_{t|t}^{(i)} \propto p(y_t|x_t^{(i)}) w_{t|t-1}^{(i)}$$
$$\sum_{i=1}^{N} w_{t|t}^{(i)} = 1$$

It doesn't work!

With this we have defined both a time update and measurement update. Only one problem: It doesn't work



Lots of particles with small weight, i.e., unlikely. The number of particles are reduced at each measurement update until there are "none" left.

A solution

Resampling!

Outline

- Nonlinear filtering
- Particle representation of distributions
- Time update
- Measurement update
- **Resampling**
- Summary of algorithm
- Exercises

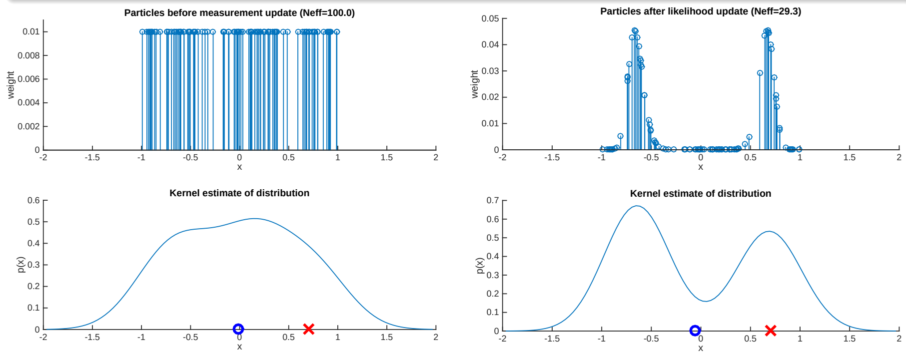
Resampling

Problem

Particles loses weight, $w^{(i)} \to 0$, number of effective particles can be defined as

$$N_{\text{eff}} = \frac{1}{\sum_i (w^{(i)})^2}$$

If all particles have equal weight $1/N$, then $N_{\text{eff}} = N$.



Resampling

Solution: Place the particles where they are needed which corresponds to where they are likely, i.e., where $p(x^{(i)})$ is high which is where the weight $w^{(i)}$ is high. This is referred to as resampling.

There are many methods for this, a simple (used in the original paper) is called SIR (sampling importance resampling)

Draw N new particles from the discrete distribution

$$p(x = x^{(i)}) = w^{(i)}$$

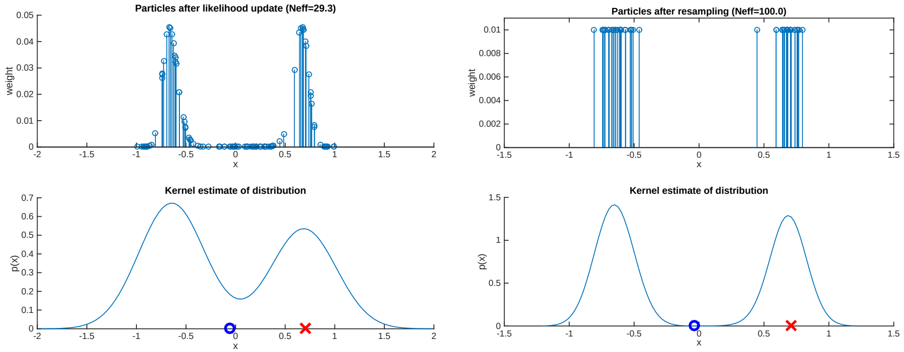
and let all particles have equal weight $w^{(i)} = 1/N$.

Resampling

Revisit the example with

$$y = x^2 + e$$

Before resampling ... after resampling



Outline

- Nonlinear filtering
- Particle representation of distributions
- Time update
- Measurement update
- Resampling
- Summary of algorithm
- Exercises

Particle filter - summary oif simplest version

1. Initialise: Draw particles $x_0^{(i)}$ and weights $w_{0|-1}^{(i)}$ according to initial guess.
2. Measurement update:
$$w_{t|t}^{(i)} \propto p(y_t|x_t^{(i)})w_{t|t-1}^{(i)}, \quad \sum_{i=1}^{N} w_{t|t}^{(i)} = 1$$
where
$$p(y_t|x_t^{(i)}) = p_e(y_t - h(x_{t|t-1}^{(i)}))$$
3. Compute
$$N_{\text{eff}} = \frac{1}{\sum_i (w^{(i)})^2}$$
and resample if $N_{\text{eff}} <$ threshold.
4. Time update:
$$\begin{aligned} w_{t+1|t}^{(i)} &= w_{t|t}^{(i)} \\ x_{t+1}^{(i)} &= f(x_t^{(i)}, \epsilon_t^{(i)}) \end{aligned}$$
5. Go to step 2

Particle filter - some comments

- There are much more to be said, the algorithm here is the basic version, SIR (Sampling Importance Resampling)
- Read Gordon et.al. which is the original reference. Easy to read and short.
- Computationally heavy, number of particles important.
- Any non-linearity, any distribution
- You have to generate samples from process noise distribution.
- Scales badly in high dimensions ($\mathbb{R}^n$ can be a big space)
- Utilize linear sub-structures and use Kalman filters for those (Rao-Blackwellised particle filter)

Outline

- *Nonlinear filtering*
- *Particle representation of distributions*
- *Time update*
- *Measurement update*
- *Resampling*
- *Summary of algorithm*
- *Exercises*

Particle representation

Exercise Le3b.1:

a) Generate $N = 500$ particles to represent the distribution $p(x) \sim \mathcal{N}(0,1)$, assign the weight $1/N$ to all particles. Plot the particles and a kernel density estimate of the distribution.

Hint: Matlab commands `normrnd`, `stem`, and `ksdensity` are useful.

b) Use the particles from the a-exercise to compute new particles representing the distribution $p(y)$ where

$$y = \sin(2\pi x^2)\sqrt{|x|}$$

Plot the particles and a corresponding kernel density estimate.

Measurement update

Exercise Le3b.2: To illustrate the measurement update step in the particle filter, assume a measurement equation

$$y = x^2 + e, \quad e \sim \mathcal{N}(0, \sigma^2)$$

with $\sigma = 0.1$ and where the a-priori knowledge of x is $\mathcal{U}[0,1]$-distributed, i.e., uniformly distributed between 0 and 1. Generate $N = 100$ particles representing $p(x)$, assume the value $y = 0.7$ is measured.

a) Compute particles and weights after the measurement update step 2. Plot particles and a kernel density estimate of the distribution.

b) Compute the number of effective particles N_{eff}

c) Resample and again plot particles and a kernel density estimate of the distribution.

d) Compute the particle filter estimate of $\hat{x} = \mathbb{E}(x|y)$.

Hint: Matlab commands `rand` and `normpdf` are useful.

Resample function

```
function ind = resample(w)
  % Perform stochastic resampling.
  % Algorithm from Ripley (1988)
  % Implementation by G. Hendeby

  N = numel(w);
  qs = cumsum(w);
  u = fliplr(cumprod(rand(1,N).^(1./(N:-1:1))));
  i = 1;
  ind = zeros(1, N);
  for p = 1:N
    while qs(i) < u(p)
      i = i + 1;
    end
    ind(p) = i;
  end
end
```

Implement SIS

Exercise Le3b.3:
Implement the particle filter as described in

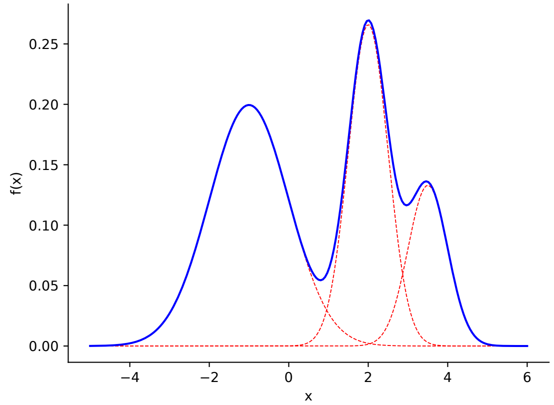
> *Gordon, Neil J., David J. Salmond, and Adrian FM Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation." IEE Proceedings F-radar and signal processing. Vol. 140. No. 2. IET, 1993.*

and reproduce Example 4.1. Regenerate Figures 1, 2, and 3 (excluding the confidence intervals).

Kernel Density Estimate (KDE)

A non-parametric way to estimate the probability density function from a set of samples $x_1, \ldots, x_n$. The density estimate is expressed as a sum of kernels

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i)$$



Nonlinear observers

design techniques, part 1b

Erik Frisk

Dept. of Electrical Engineering, Linköping University

December 20, 2018

LINKÖPING UNIVERSITY