OPTIMAL FILTERING

LECTURE 7

- Non-linear filtering
  - Extended Kalman
  - Unscented Kalman
  - Particle filter
  - Rao-Blackwellized particle filter.

KTH Electrical Engineering

Reading instructions

Arulampalam, S., Maskell, S., Gordon, N. and Clapp, T. (2002) “A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking”. IEEE Trans. SP, 50(2):174–188.

F. Gustafsson, “Particle filter theory and practice with positioning applications,” IEEE A&E Syst. Mag. Part II: Tutorials, vol. 25, no. 7, pp. 53–82, 2010.

Julier, S. J., and Uhlmann, J. K. “Unscented Filtering and Nonlinear Estimation,” Proceedings of the IEEE 92, 3 (March 2004), 401–422.

F. Daum, “Nonlinear filters: Beyond the Kalman filter,” Aerospace and Electronic Sys. Magazine, vol. 20, no. 8, pp. 57–69, Aug. 2005.

KTH Electrical Engineering

Motivation

Wiener & Kalman

- Linear system models
- Only care about an estimate of x (or y).
- Limiting yourself to linear estimators and minimizing mean square error

  or
- Process/measurement noise is Gaussian

Non-linear filters

- Non-linear model

  or
- Non-Gaussian noise
- Possibly interested in full posterior density of x (or y).

KTH Electrical Engineering

Model

$$x_{k+1} = f(x_k) + w_k$$
$$y_k = h(x_k) + v_k$$

- w_k and v_k temporally white, arbitrary distributions
- Markov model!
- Can be generalized, see previous lecture.

KTH Electrical Engineering

Extended Kalman Filter (EKF)

- Approximate all distributions to be Gaussian
- Use first order Taylor approximation around current state estimate

$$f(x_k) \approx f(\hat{x}_{k|k}) + F_k(x_k - \hat{x}_{k|k})$$
$$h(x_k) \approx h(\hat{x}_{k|k-1}) + H_k(x_k - \hat{x}_{k|k-1})$$

KTH Electrical Engineering

- Linearized model

$$x_{k+1} \approx F_k x_k + \underbrace{f(\hat{x}_{k|k}) - F_k \hat{x}_{k|k}}_{\text{known virtual input}} + w_k$$
$$y_k \approx H_k x_k + \underbrace{h(\hat{x}_{k|k-1}) - H_k \hat{x}_{k|k-1}}_{\text{known virtual input}} + v_k$$

More on Extended Kalman

Variations of EKF

- Use second (higher) order Taylor
- Iteratively update the first order Taylor approximation several times within each Kalman filter iteration

KTH Electrical Engineering

Problems with EKF

- Best for "mild" non-linearities. Rough approximation otherwise.
- Need to calculate (or numerically approximate) derivatives in each iteration.

Unscented Kalman – Sigma Point Filters

- Approximate all distributions to be Gaussian
- Use derivative free approximation of first and second order moments
- Gaussian distribution represented by small number of *sigma points*.
- Each sigma point given by a value and a weight
- Work flow:
  - Mean & covariance $\Longrightarrow$ sigma points
  - Pass the sigma points through the non-linearity
  - Resulting new sigma points $\Longrightarrow$ mean & covariance
- Resulting filter similar to Kalman in structure

Mean/Covariance $\Longrightarrow$ Sigma Points

Given mean μ_x and covariance $\Sigma_x = SS^*$ (S is a "square root" of Σ_x):

$$x^{(0)} = \mu_x \qquad w^{(0)} = \text{user choice}$$

$$x^{(k)} = \mu_x + \sqrt{\frac{n_x}{1-w^{(0)}}}S_{:,k} \qquad w^{(k)} = \frac{1-w^{(0)}}{2n_x}$$

$$x^{(-k)} = \mu_x - \sqrt{\frac{n_x}{1-w^{(0)}}}S_{:,k} \qquad w^{(-k)} = \frac{1-w^{(0)}}{2n_x}$$

- n_x = dimensionality of x
- $2n_x + 1$ sigma points, $k = -n_x, \ldots, n_x$
- Note: $\sum_{k=-n_x}^{n_x} w^{(k)} = 1$
- Many other choices available in the literature

Sigma Points $\Longrightarrow$ Mean/Covariance

Goal: Mean and variance of $y = h(x)$.

Transformed sigma points: $y^{(k)} = h(x^{(k)})$.

"Estimated" mean and covariance:

$$\mu_y \approx \sum_{k=-n_x}^{n_x} w^{(k)} y^{(k)}$$

$$\Sigma_y \approx \sum_{k=-n_x}^{n_x} w^{(k)} \big(y^{(k)} - \mu_y\big)\big(y^{(k)} - \mu_y\big)^*$$

Extra correction term sometimes used for the covariance

Point Mass & Particle Filters

Main idea: Approximate a continuous distribution by a discrete distribution, sum of "particles".

Each particle represented by a value and a weight (probability).

Point mass filters: Values from a fixed regular grid.

Particle filters: Values drawn at random (stochastic grid)

Importance Sampling

- Standard Monte Carlo:

$$\mathrm{E}[g(x)] = \int g(x)p(x)\,dx \approx \frac{1}{N}\sum_{n=1}^{N} g(x^{(n)})$$

where $x^{(n)}$ i.i.d. drawn from distribution $p(x)$.

- Importance Sampling: Introduce suitable *proposal density* $q(x)$:

$$\mathrm{E}[g(x)] = \int g(x)p(x)\,dx = \int g(x)\underbrace{\frac{p(x)}{q(x)}}_{\text{weight}}\underbrace{q(x)}_{\substack{\text{proposal}\\ \text{dens.}}}\,dx \approx \sum_{n=1}^{N} g(x^{(n)})w^{(n)}$$

where $x^{(n)}$ drawn from the proposal density $q(x)$ and $w^{(n)} \propto \frac{p(x^{(n)})}{q(x^{(n)})}$.

Importance Sampling Cont.

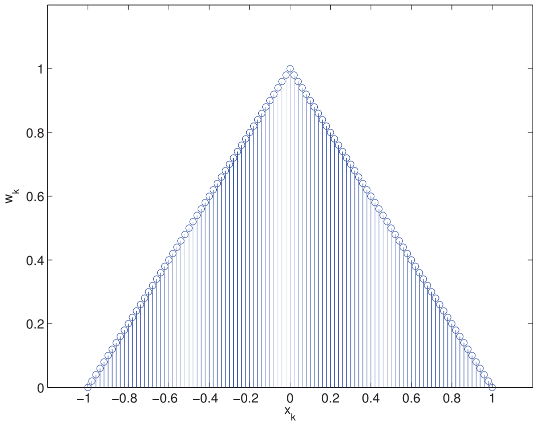
Equivalently $p(x)$ is approximated by

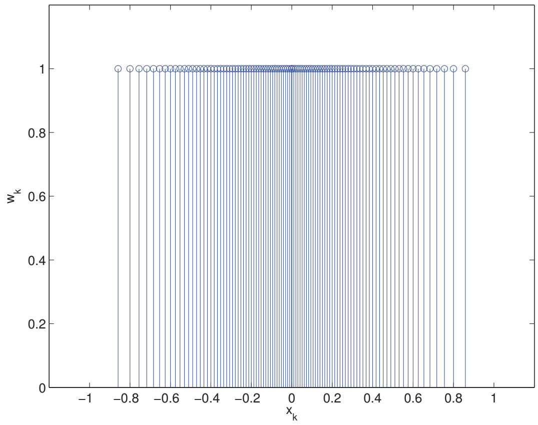
$$p(x) \approx \sum_{n=1}^{N} w^{(n)}\delta(x - x^{(n)})$$

$\{x^{(n)}, w^{(n)}\}$ called **particle distribution**.

Sometimes $p(x)$ or $q(x)$ contain an unknown scaling $\Longrightarrow$ weights $w^{(n)}$ are scaled. Easy solution is to normalize so $\sum w^{(n)} = 1$.

Illustration Importance Sampling





Case 1: Uniformly distributed particles

Case 2: Equally weighted particles

Note: Both represent the same distribution !

Resampling

Given particle distribution $\{x^{(n)}, w^{(n)}\}$, draw new particle values with replacement from the set $\{x^{(n)}\}$ with $P(x = x^{(n)}) = w^{(n)} \Longrightarrow$ i.i.d. samples from the density

$$p(x) = \sum_{n=1}^{N} w^{(n)} \delta(x - x^{(n)})$$

$\Longrightarrow$ all new weights $= 1/N$

Algorithms: “Ripley’s method”, “Stratified sampling”, “Systematic sampling”, . . .

Note: Many repeated values in the resampled particle distribution.

When needed: If weights before resampling are too different: $\hat{N}_{\text{eff}} = 1/\sum (w^{(n)})^2 \ll N$

Particle Filter

Initialization: Generate N particles $x_1^{(n)} \sim p_{x_0}$, $w_{1|0}^{(n)} = 1/N$.

Measurement update: $w_{k|k}^{(n)} \propto w_{k|k-1}^{(n)} p(y|x_k^{(n)})$

Optionally resample

Time update:

Generate prediction values from proposal distribution $x_{k+1}^{(n)} \sim q(x_{k+1}|x_k^{(n)}, y_{k+1})$

Update weights $w_{k+1|k}^{(n)} = w_{k|k}^{(n)} \frac{p(x_{k+1}^{(n)}|x_k^{(n)})}{q(x_{k+1}|x_k^{(n)}, y_{k+1})}$

Particle Filter

Problem: Most weights may go to zero "sample depletion"

Solutions:

- Large number of particles
- Resampling
- Good choice of proposal distribution
- Dithering
- ...

Curse of dimensionality Too many particles needed to cover interesting area in higher dimensions. Works best for state vectors of dimension $\leq 2, 3, 4$.

Rao-Blackwellized particle filter

(Marginalized particle filter)

- If state update is linear in some state variables & Gaussian process noise.
- Use particle representation for non-linear state variables
- Use Kalman for linear state variables
- Non-linear part acts as extra "measurements" given predicted non-linear state variables
- Can handle large state dimensions if only few are non-linear.