DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF COPENHAGEN

Sequential Inference by Filtering: Particle filtering

Kim Steenstrup Pedersen

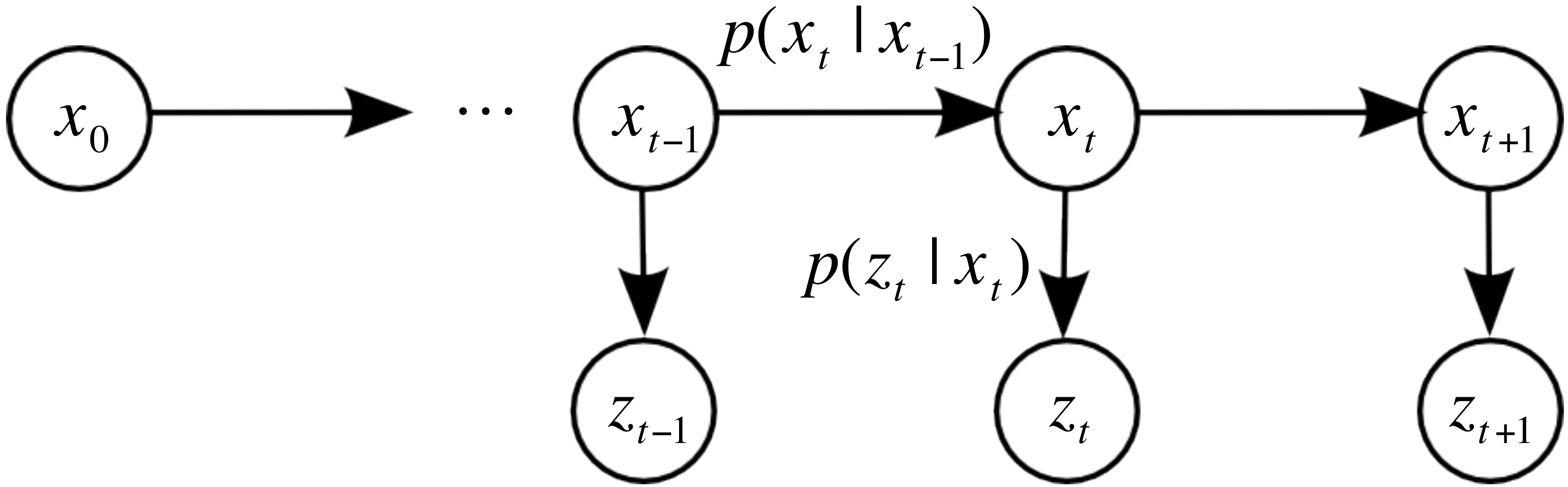
Plan for today

- The sequential inference / estimation problem solved with Bayesian filtering
- Kalman filtering
- Particle filtering - a sequential Monte Carlo method

- COMMENT: I will use human motion tracking as motivating example, but remember these are general methods.

Sequential estimation by Bayesian filtering

- Given observations $z_t \in R^m$ infer the hidden state $x_t \in R^n$ where *t* denotes discrete time.
- x_t and z_t are random variables (i.e. are uncertain / noisy).
- Lets assume that the states x_t form a 1st order Markov chain and the observations z_t are independent conditioned on x_t.



Sequential estimation by Bayesian filtering

- Let $x_{0:t} \equiv \{x_0, \ldots, x_t\}$ and $z_{1:t} \equiv \{z_1, \ldots, z_t\}$
- We want to recursively estimate either the:
  - Posterior distribution

    $$p(x_{0:t} \mid z_{1:t})$$

  - Filtering distribution (marginal of posterior)

    $$p(x_t \mid z_{1:t}) = \int p(x_{0:t} \mid z_{1:t}) dx_{0:t-1}$$

  - Expectation with respect to posterior

    $$E_{p(x_{0:t} \mid z_{1:t})}[h_t(x_{0:t})] = \int h_t(x_{0:t}) p(x_{0:t} \mid z_{1:t}) dx_{0:t}$$

  - Expectation with respect to filtering distribution, etc.

Sequential estimation by Bayesian filtering

Bayesian filtering are governed by two equations:

- Dynamical model:

$$x_{t+1} = f(x_t) + s_t$$ (Stochastic difference equation)

$$x_{t+1} \sim p(x_{t+1} \mid x_t)$$ (Distribution of the state)

- Observation model:

$$z_t = g(x_t) + v_t$$

$$z_t \sim p(z_t \mid x_t)$$ (Distribution of observations given state)

We also need to know the distribution on the initial state $p(x_0)$

Bayesian filtering: Recursive marginal

- Relation between posterior and filtering distribution:

$$p(x_t \mid z_{1:t}) = \int p(x_{0:t} \mid z_{1:t}) dx_{0:t-1}$$

General filtering steps:

- Prediction:

$$p(x_t \mid z_{1:t-1}) = \int \underbrace{p(x_t \mid x_{t-1}) p(x_{t-1} \mid z_{1:t-1})}_{=p(x_t,\, x_{t-1} \mid z_{1:t-1})} dx_{t-1}$$

- Correction (update):

$$p(x_t \mid z_{1:t}) = \frac{p(z_t \mid x_t) p(x_t \mid z_{1:t-1})}{\int p(z_t \mid x_t) p(x_t \mid z_{1:t-1}) dx_t}$$

Summary of Kalman filtering

The assumptions are:

- The dynamical and observation models are linear.
- Both the dynamical and observation noises are Gaussian distributed.

The Kalman filter tracks an estimate of a Gaussian filtering distribution, i.e. by tracking the mean and covariance.

Are these assumptions always valid?

Probably not – depends on the nature of the tracking problem.

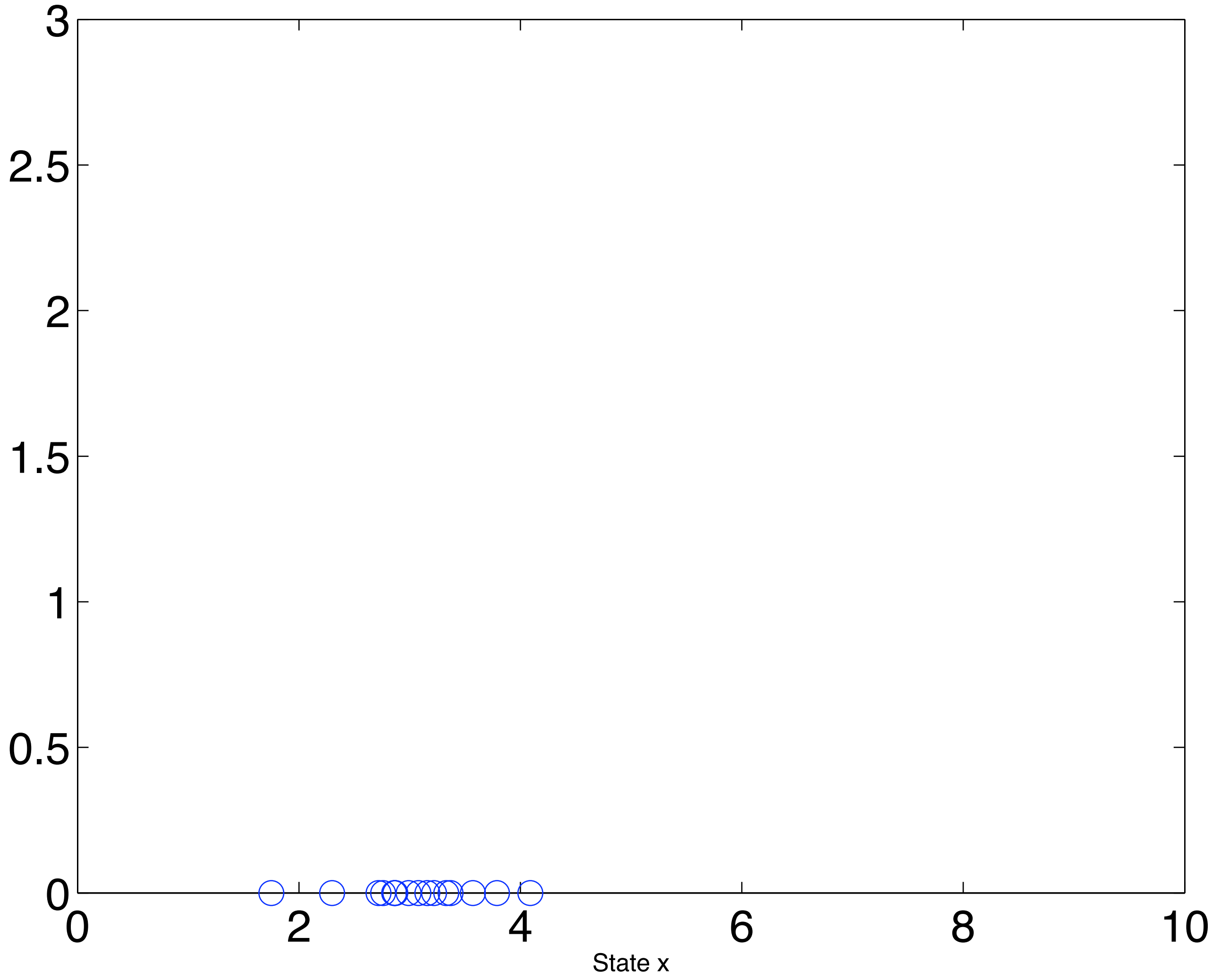
What to use instead of the Gaussian model?

What choices do we have for modelling the filtering distribution?

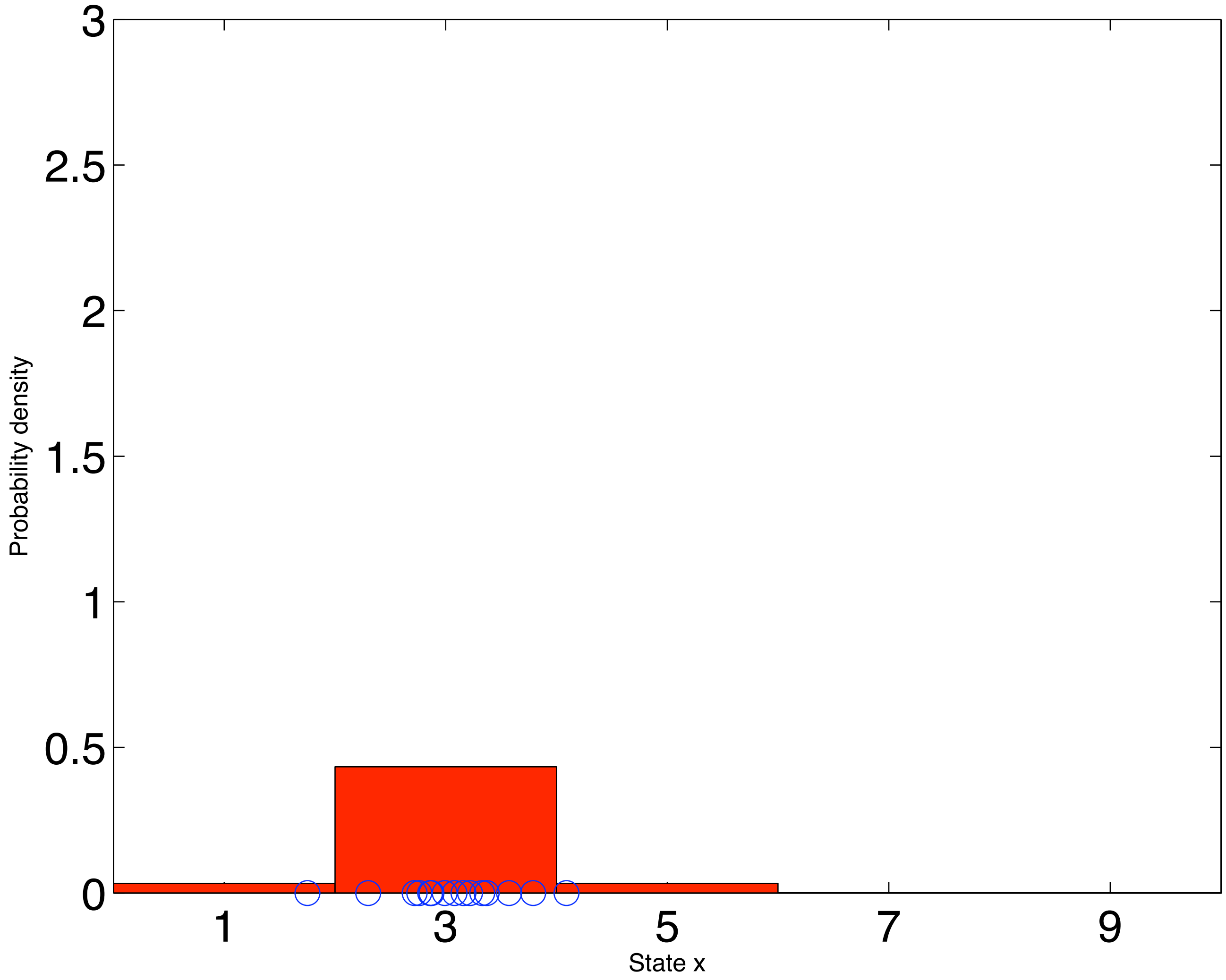
- **Parametric models:**
  - Find probability distributions with a “nice” functional form that allows us to construct a filtering algorithm.
    This is in general difficult!

- **Non-parametric models:**
  - Histogram density estimates
  - Kernel density estimates
  - Weighted samples / particles (Monte Carlo approach)

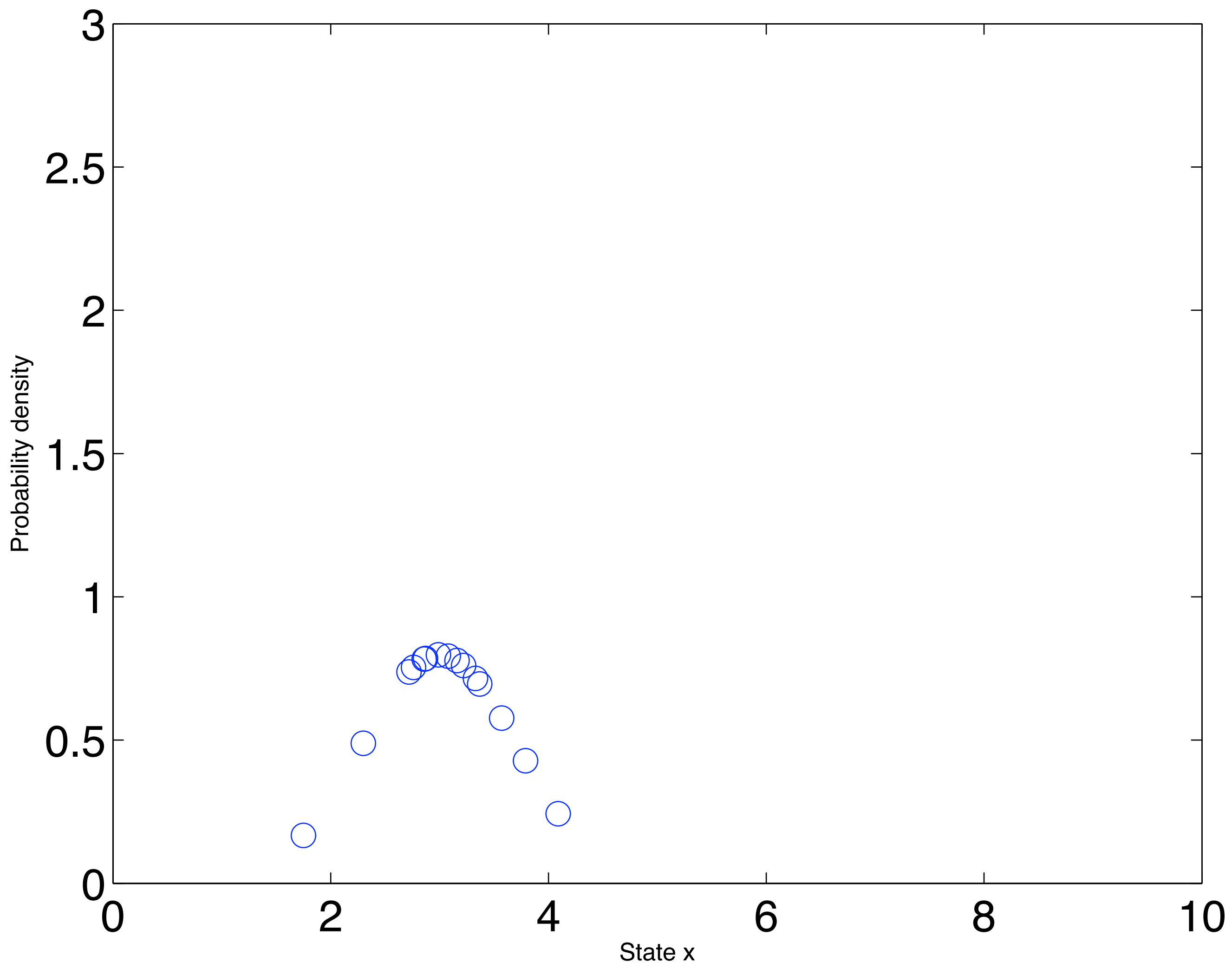
Monte Carlo Example: 1D estimate (particles)



Example: 1D estimate (histogram)



Example: 1D estimate (particles with weights)



Particle filter: The intuitive explanation

- Approximate $p(x_t \mid z_{1:t})$ with a set of *N* particles:

  $$x_t^{(i)} \quad , i = 1, \ldots, N$$

  – With weights $w_t^{(i)} \quad , \sum_{i=1}^{N} w_t^{(i)} = 1$

- Dynamical and observation models given by

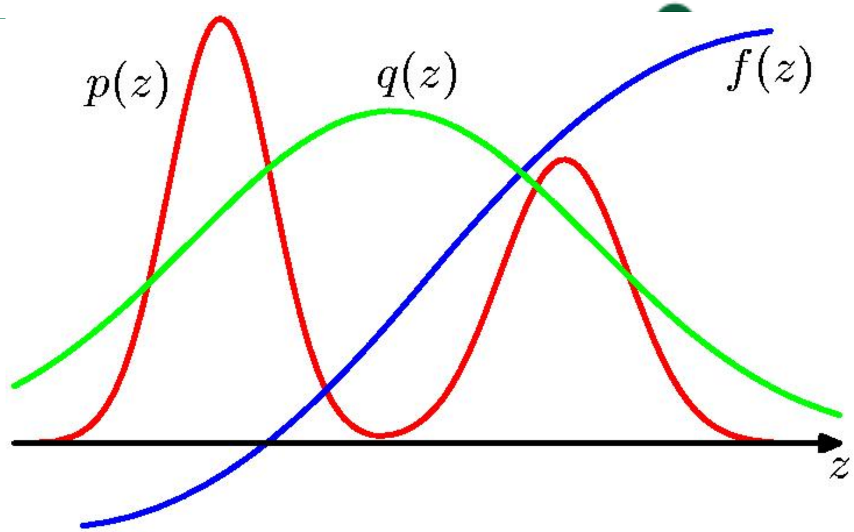
  $$x_{t+1} = f(x_t) + s_t \sim p(x_{t+1} \mid x_t)$$

  $$z_t = g(x_t) + v_t \sim p(z_t \mid x_t)$$

But we need a method for generating samples

Importance Sampling
Approximate expectation $E[f]$

- Sample i.i.d. from $q(\mathbf{z})$
  $(\mathbf{z}^{(1)},\ldots,\mathbf{z}^{(L)})$
- Use samples to approximate $E[f]$ by



$$E[f] \approx \sum_{l=1}^{L} \omega_l f(\mathbf{z}^{(l)})$$

Renormalized Importance weights

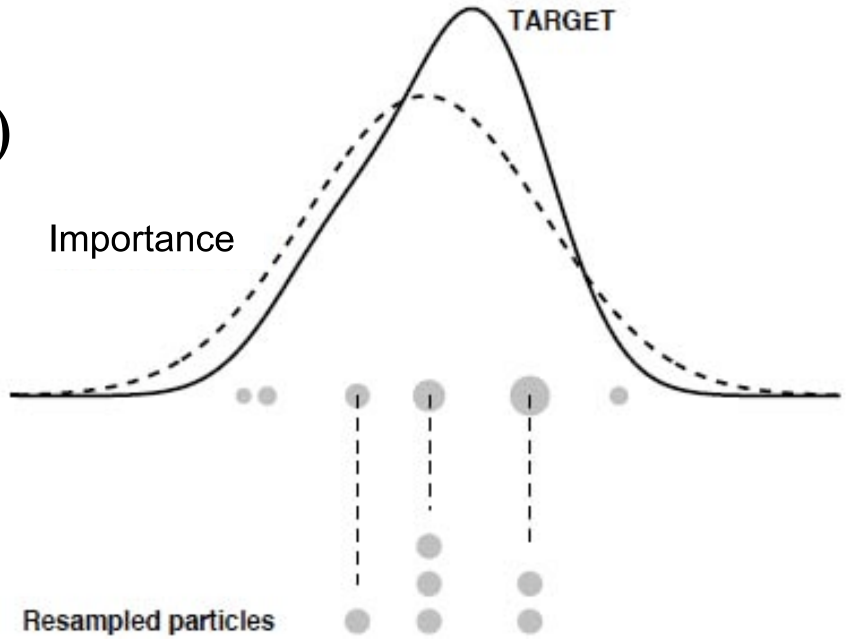
$$\omega_l = \frac{\tilde{p}(\mathbf{z}^{(l)}) / \tilde{q}(\mathbf{z}^{(l)})}{\sum_m \tilde{p}(\mathbf{z}^{(m)}) / \tilde{q}(\mathbf{z}^{(m)})}$$

$$p(\mathbf{z}) = 1/Z_p \, \tilde{p}(\mathbf{z})$$

$$q(\mathbf{z}) = 1/Z_q \, \tilde{q}(\mathbf{z})$$

Sampling-Importance-Resampling (SIR)
A two stage approach

- Sampling: Sample i.i.d. samples $\mathcal{M} = (\mathbf{z}^{(1)},\ldots,\mathbf{z}^{(M)})$ from $q(\mathbf{z})$
- Importance: Compute importance weights
- Resampling: Sample with replacement from $\mathcal{M}$ based on weights $\omega^{(l)}$ $(\mathbf{z}^{(1)},\ldots,\mathbf{z}^{(N)})$
  Usually $M \geq N$



Sequential SIR
Considering sequences and the posterior

- Consider each particle as a sequence of states

$$x_{0:t}^{(i)} = x_0^{(i)}, x_1^{(i)}, \ldots, x_t^{(i)}$$

- Importance weights for sequences:

$$\tilde{w}_t^{(i)} = \frac{p(x_{0:t}^{(i)} \mid z_{1:t})}{q(x_{0:t}^{(i)})}$$

and normalized $w_t^{(i)} = \tilde{w}_t^{(i)} \Big/ \sum_{i=1}^{M} \tilde{w}_t^{(i)}$

Recall: Recursive posterior

- Let $x_{0:t} \equiv \{x_0, \ldots, x_t\}$ and $z_{1:t} \equiv \{z_1, \ldots, z_t\}$
- Bayes' theorem gives the posterior

$$p(x_{0:t} \mid z_{1:t}) = \frac{p(z_{1:t} \mid x_{0:t})\, p(x_{0:t})}{\int p(z_{1:t} \mid x_{0:t})\, p(x_{0:t})\, dx_{0:t}}$$

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

- Recursive estimate of posterior

$$p(x_{0:t+1} \mid z_{1:t+1}) = p(x_{0:t} \mid z_{1:t}) \frac{p(z_{t+1} \mid x_{t+1})\, p(x_{t+1} \mid x_t)}{p(z_{t+1})}$$

Sequential SIR – the proposal distribution

- If we choose the following proposal distribution

$$q(x_{0:t}) = \text{transition} \times \text{posterior(t-1)} = p(x_t \mid x_{t-1})p(x_{0:t-1} \mid z_{1:t-1})$$

- The importance weights simplifies to

$$\tilde{w}_t^{(i)} = \frac{p(x_{0:t}^{(i)} \mid z_{1:t})}{q(x_{0:t}^{(i)})} = \frac{1}{p(z_t)} \frac{p(z_t \mid x_t^{(i)})p(x_t^{(i)} \mid x_{t-1}^{(i)})p(x_{0:t-1}^{(i)} \mid z_{1:t-1})}{p(x_t^{(i)} \mid x_{t-1}^{(i)})p(x_{0:t-1}^{(i)} \mid z_{1:t-1})}$$

$$= p(z_t \mid x_t^{(i)}) \quad (p(z_t) \text{ does not matter after renormalization})$$

- Renormalized $w_t^{(i)} = \tilde{w}_t^{(i)} \Big/ \sum_{i=1}^{M} \tilde{w}_t^{(i)}$

Sequential SIR – the distribution after resampling

- After resampling the particle distribution approximates:

$$x_{0:t}^{(i)} \sim \tilde{w}_t^{(i)} p(x_t^{(i)} \mid x_{t-1}^{(i)}) p(x_{0:t-1}^{(i)} \mid z_{1:t-1}) = p(x_{0:t}^{(i)} \mid z_{1:t})$$

- From sequences to states at time *t*:

$$x_{0:t}^{(i)} = x_0^{(i)}, x_1^{(i)}, \ldots, x_t^{(i)}$$

The states at time *t* represented by the particles are distributed as the filter distribution:

$$x_t^{(i)} \sim p(x_t^{(i)} \mid z_{1:t})$$

Particle filter: The intuitive explanation

- Approximate $p(x_t \mid z_{1:t})$ with a set of *N* particles:

  $$x_t^{(i)} \quad , i = 1, \ldots, N$$

  – With weights $w_t^{(i)} \quad , \sum_{i=1}^{N} w_t^{(i)} = 1$

- Dynamical and observation models given by

  $$x_{t+1} = f(x_t) + s_t \sim p(x_{t+1} \mid x_t)$$

  $$z_t = g(x_t) + v_t \sim p(z_t \mid x_t)$$

Particle filter: The intuitive explanation

1. **Prediction:**
   – Sample according to transition probability
   or equivalently:
   1. Move particles according to the dynamical model
   2. Diffuse by adding prediction noise

2. **Correction:** Compute new weights using observation model

3. **Resampling:** Sample particles independently identically (i.i.d.) from state distribution given by the weights and return to step 1

Example: 1D position tracking

- Dynamical model:
  $x_t = x_{t-1} + 2 + w_t \quad , \quad w_t \sim N(w_t \mid 0, 0.5^2)$
- Observation model:
  $z_t = x_t + v_t \quad , \quad v_t \sim N(v_t \mid 0, 0.5^2)$

$$p(x_t \mid x_{t-1}) = N(x_t \mid x_{t-1} + 2, 0.5^2) \quad , \quad p(z_t \mid x_t) = N(z_t \mid x_t, 0.5^2)$$

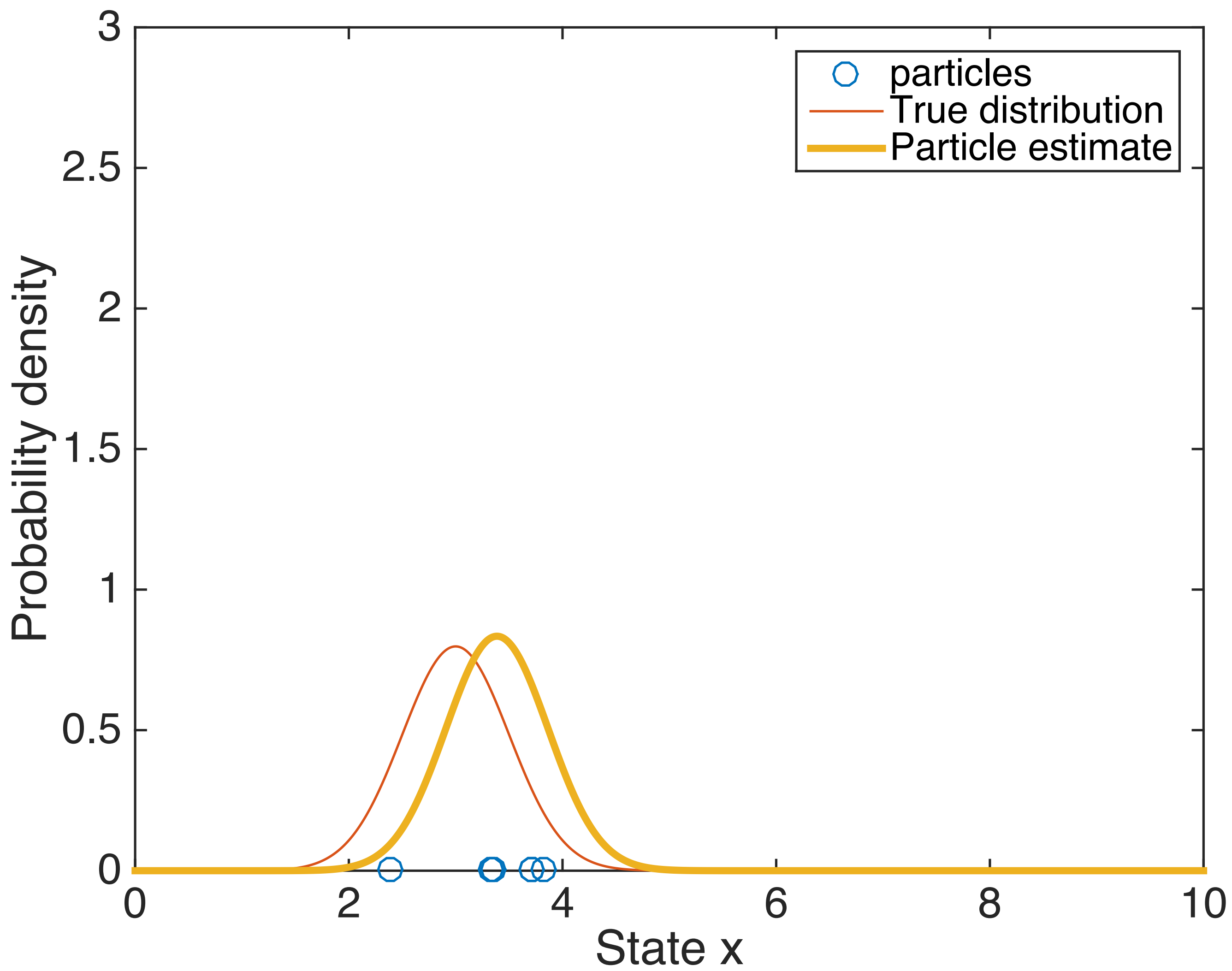
- Current state mean and uncertainty (variance):

  $x_{t-1} = 3 \quad , \quad P_{t-1} = 0.5^2$

  $p(x_{t-1} \mid z_{1:t-1}) = N(x_{t-1} \mid 3, 0.5^2)$

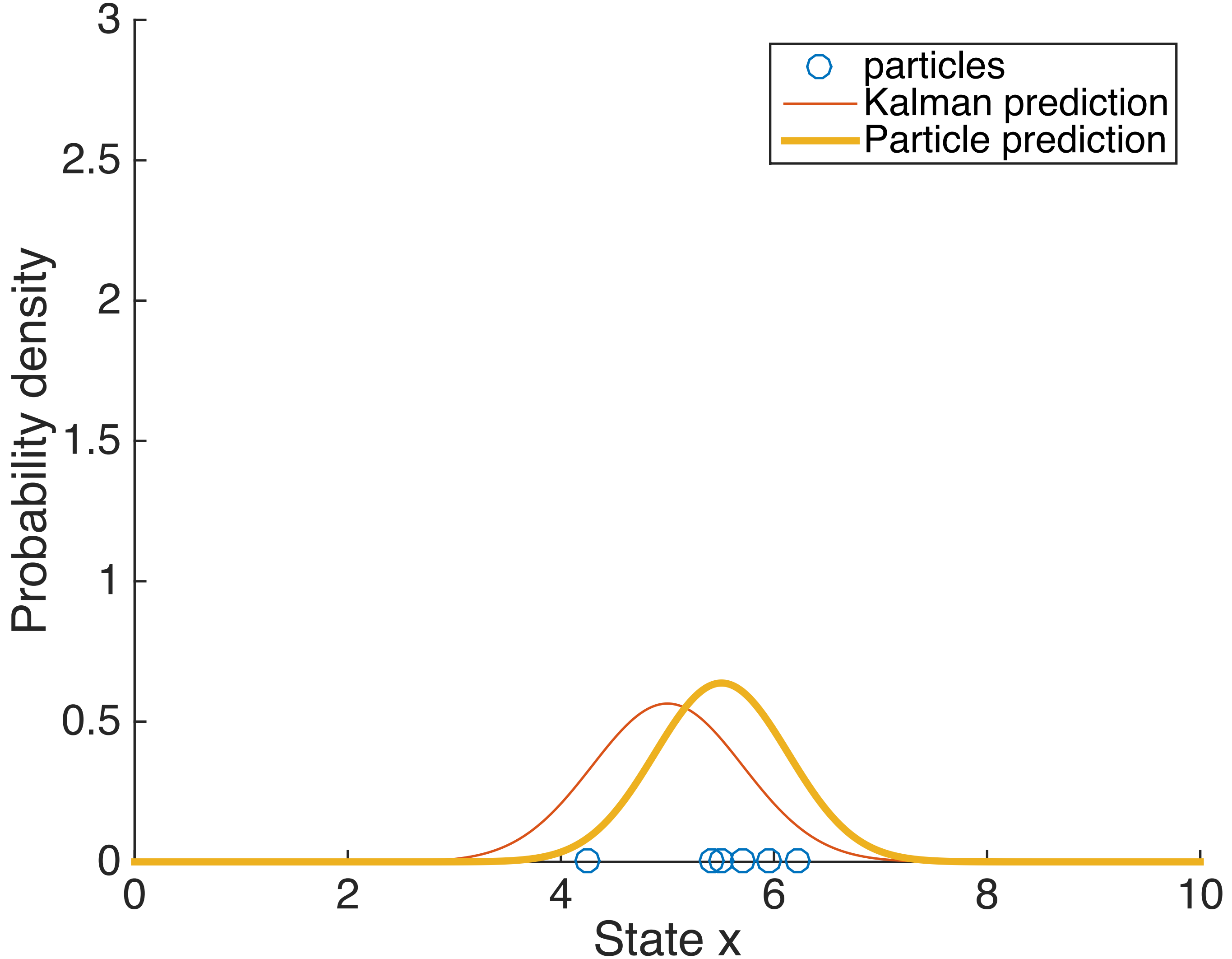
- Lets make a prediction of the new state!
  We can use the dynamical model.

1D Example: Particles compared to Kalman



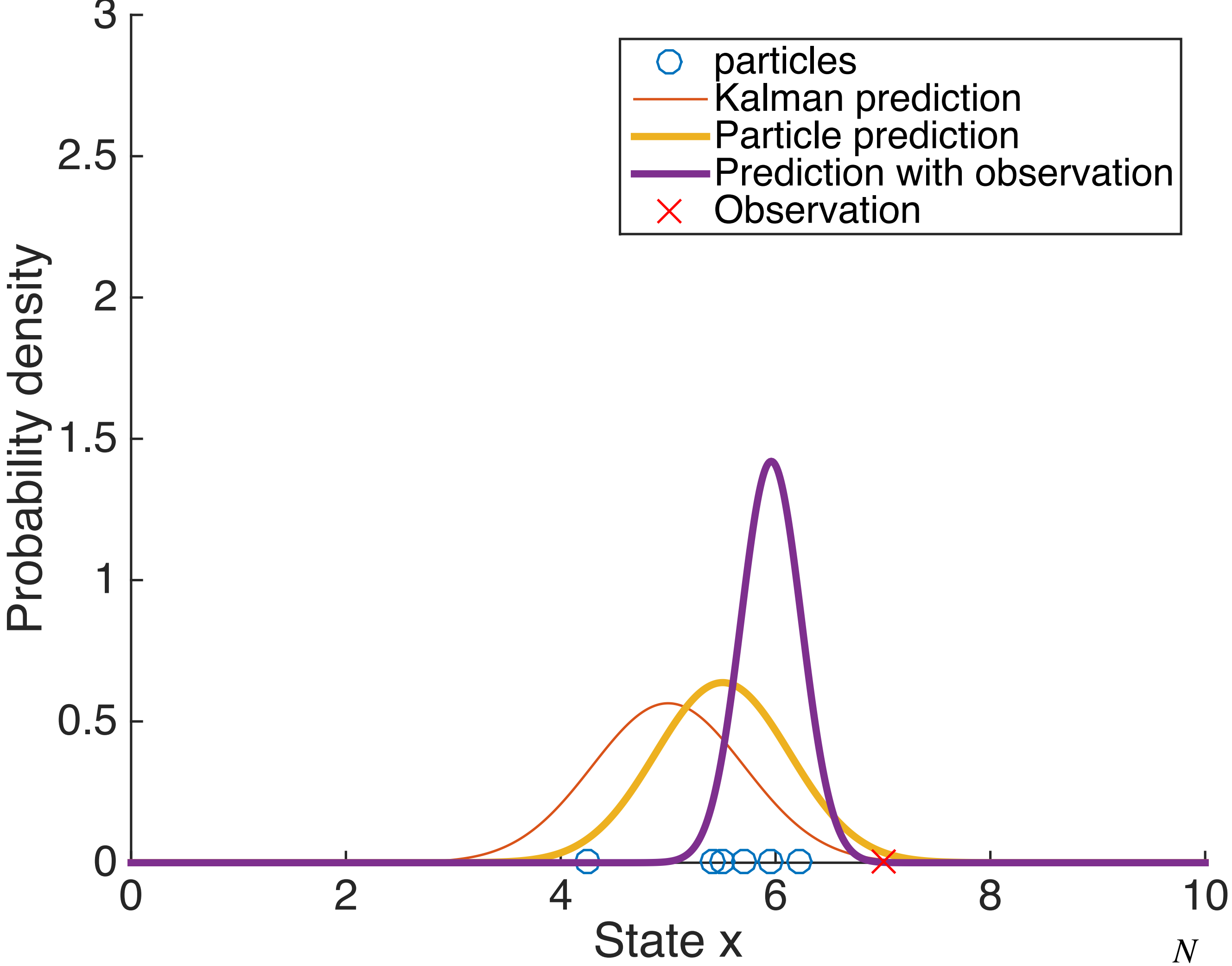
$$Initial: x_{t-1} \sim N(x_{t-1} \mid 3, 0.5^2)$$

1D Example: Particles compared to Kalman Prediction: Move and diffuse



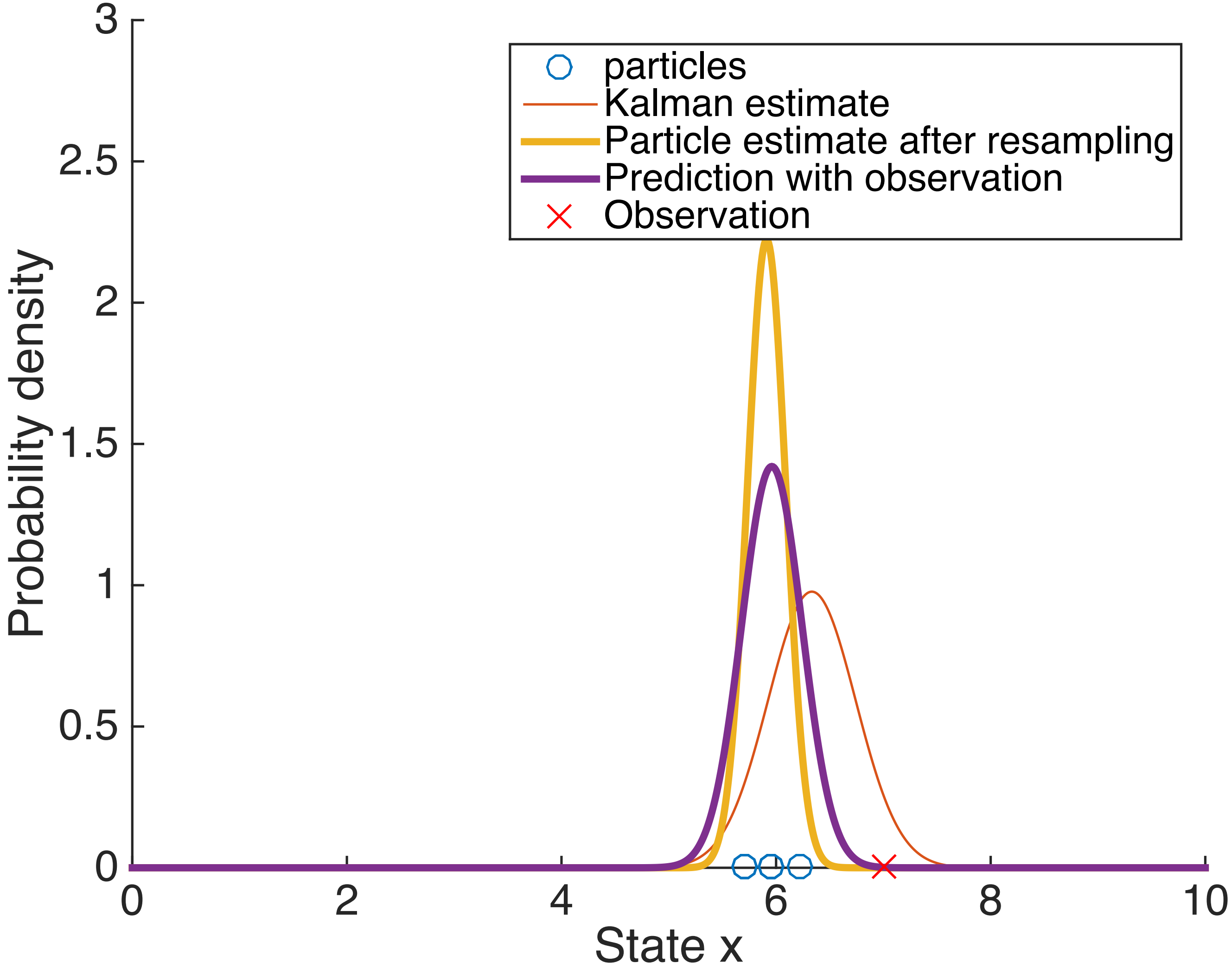
Kalman prediction $\hat{x}_t \sim N(\hat{x}_t \mid 5, 0.5^2 + 0.5^2) = N(\hat{x}_t \mid 5, 0.7^2)$

1D Example: Particles compared to Kalman
Correction: New weights based on observation



$$p(z_t \mid x_t) = N(z_t \mid x_t, 0.5^2) \text{ and } w_t^{(i)} = p(z_t \mid x_t^{(i)}) / \sum_{i=1}^{N} w_t^{(i)}$$

1D Example: Particles compared to Kalman
After resampling the particles



Why these steps?

- What does the sampling do?
  - This makes the particle go towards the mode of the distribution
- What happens if we don’t diffuse particles?
  - Otherwise they degenerate into the most likely particle
- Why do we need the weights?
  - The weights and particles represent our estimate of the distribution
  - We could do without them using Monte Carlo sampling, but then we need much more particles to track the distribution and especially the modes of the distribution.

Why these steps?

- Why do we need the resampling?
  - If we do not do resampling, most weights will be close to zero leading to the particles being redundant.
  - Resampling guarantees that all particles are effectively in use.

- Number of particles matter
  - Too few particles leads to all particles being close to modes of the distribution leading to a poor representation
  - Many particles is costly in computation time

The (bootstrap) particle filter algorithm

1. Initialization at t=0
   - For $i = 1,\ldots,N$ sample $x_0^{(i)} \sim p(x_0^{(i)})$ and $\mathrm{t} = 1$
2. Importance sampling step
   - For $i = 1,\ldots,N$ sample $\tilde{x}_t^{(i)} \sim p(x_t \mid x_{t-1}^{(i)})$
   - For $i = 1,\ldots,N$ evaluate $\tilde{w}_t^{(i)} = p(z_t \mid \tilde{x}_t^{(i)})$
   - Renormalize the importance weights
3. Selection step
   - Resample with replacement N particles from set $\{\tilde{x}_t^{(i)}; i=1,\ldots,N\}$ according to weights
   - Set $\mathrm{t} \leftarrow t+1$ and go to step 2

(From the book "Sequential Monte Carlo Methods in Practice", Doucet, de Freitas, Gordon)

What do we have after each step?

- Step 2: A set of weighted particles approximating the filtering distribution.

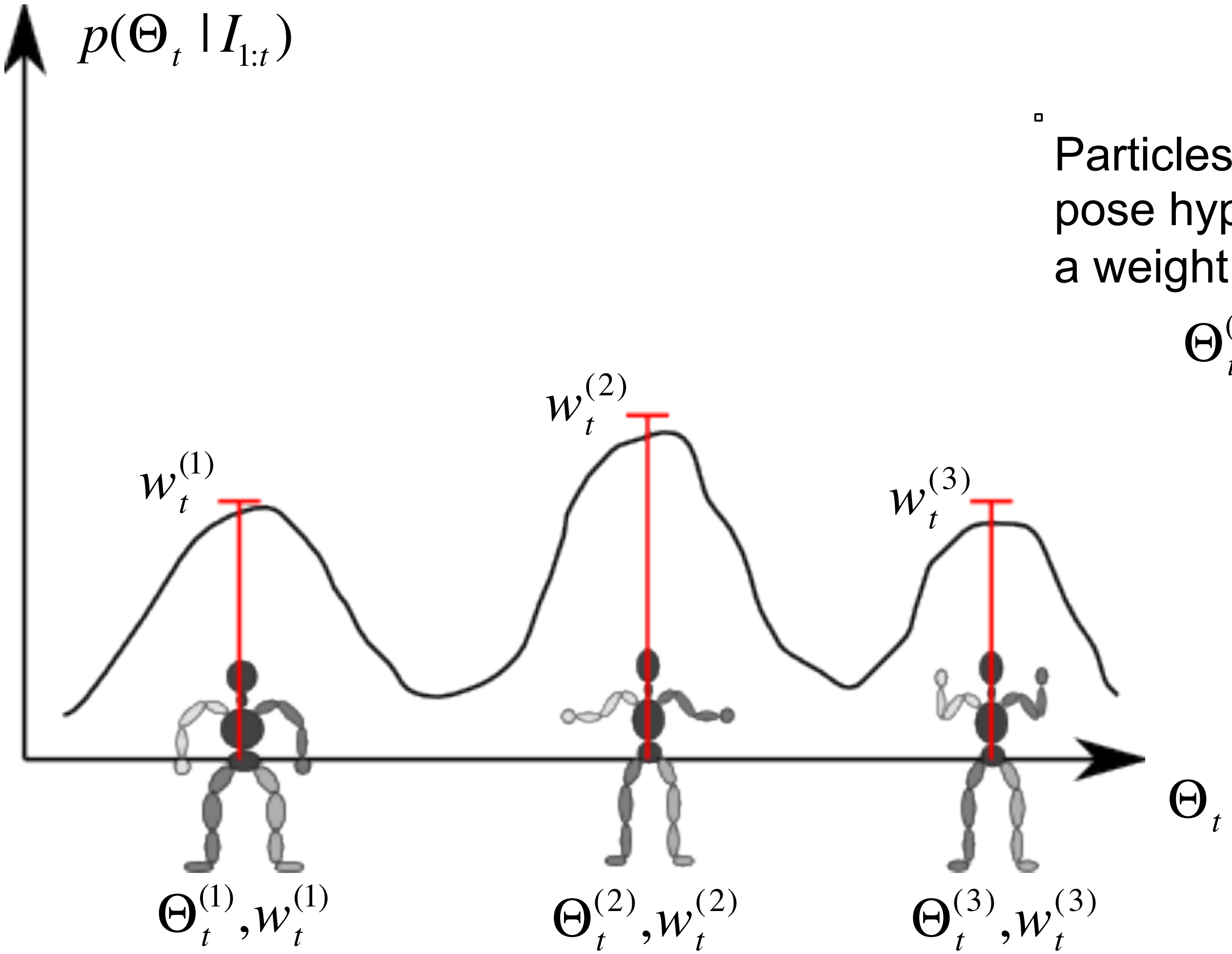
$$\tilde{w}_t^{(i)} = p(z_t \mid \tilde{x}_t^{(i)})$$

- Step 3: A set of uniformly weighted particles approximating the filtering distribution.

$$w_t^{(i)} = \frac{1}{N}$$

- We do not need the past weight in the weight update.

Back to tracking:
Particle filtering as an approximate tracker



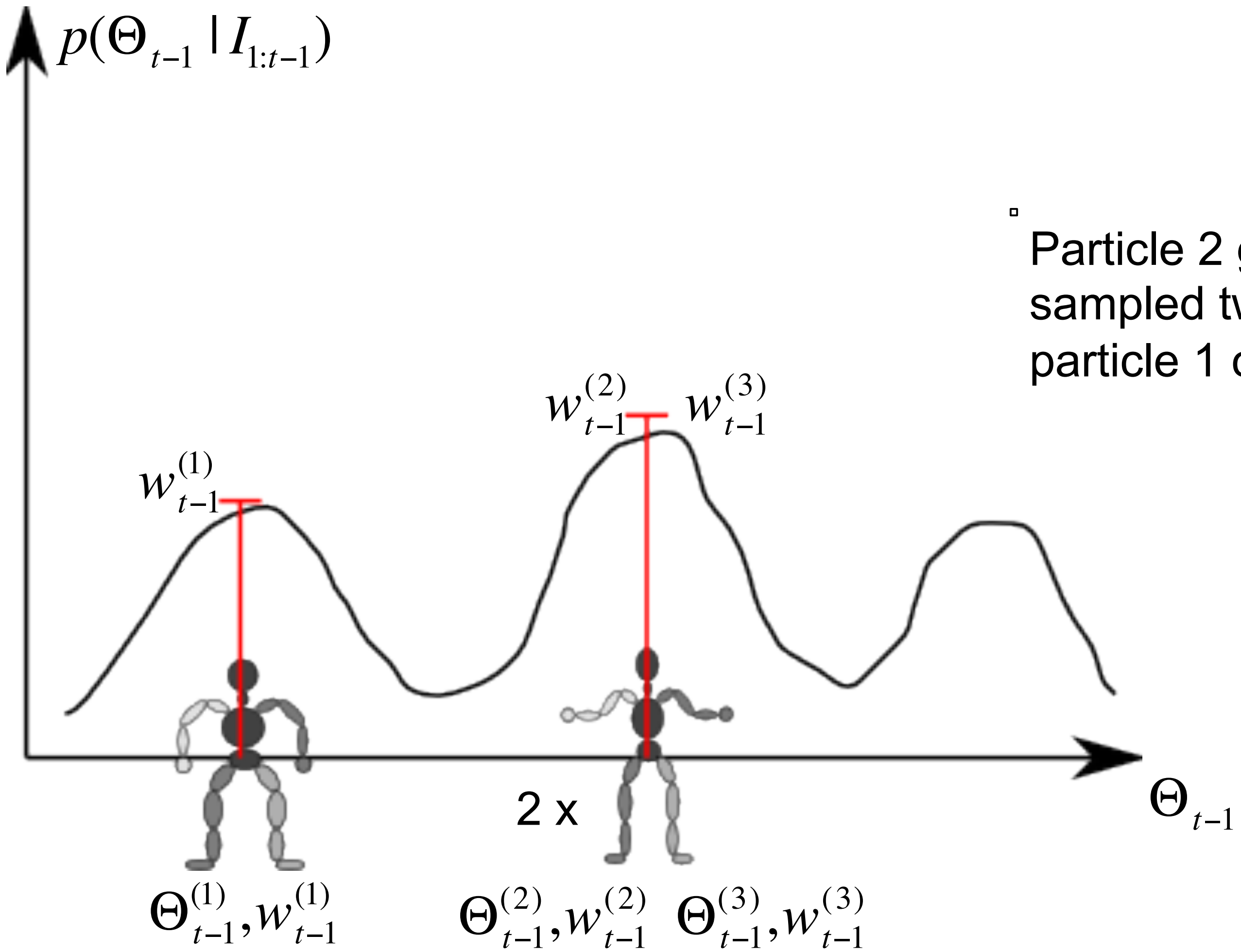
- Particles consist of a pose hypothesis and a weight:

  $$\Theta_t^{(i)}, w_t^{(i)}$$

The more particles, the more precise an approximation of $p(\Theta_t \mid I_{1:t})$

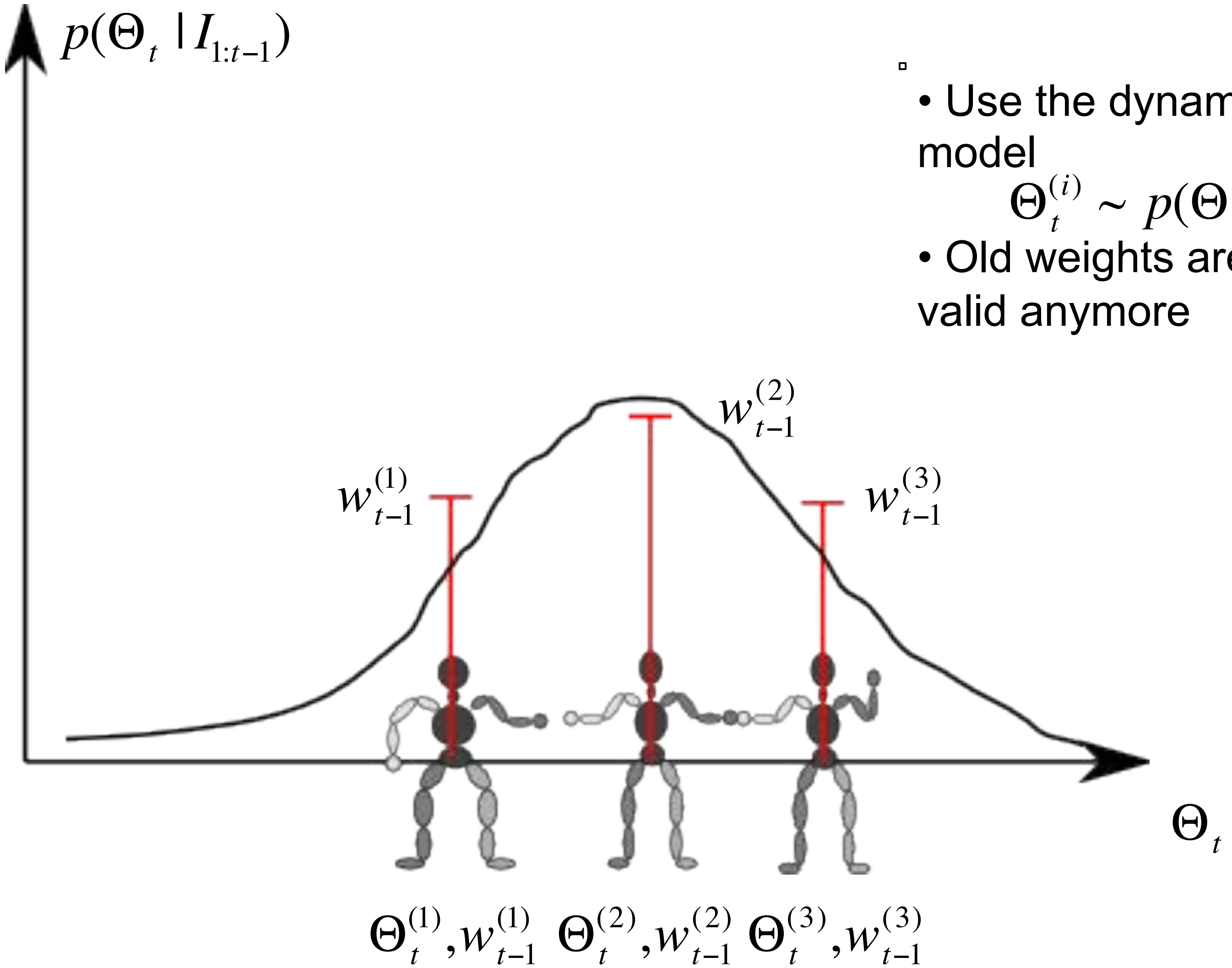
Particle filtering: Resampling
Choose particles randomly based on weights

Randomly pick particles such that we always have *N* particles.



Particle 2 gets sampled twice and particle 1 once.

Particle filtering: Prediction
Make an educated guess of the next pose

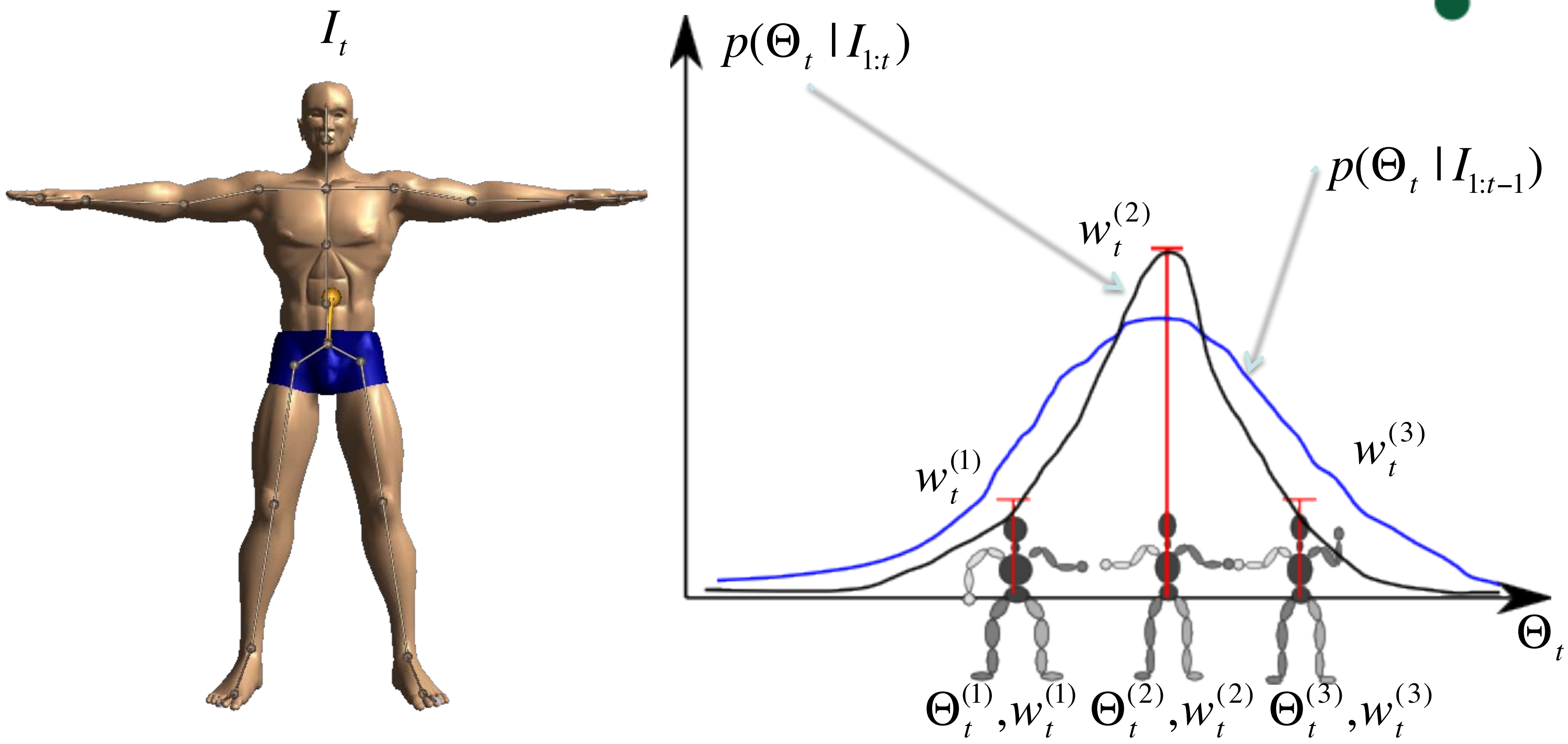


- Use the dynamical model

$$\Theta_t^{(i)} \sim p(\Theta_t \mid \Theta_{t-1})$$

- Old weights are not valid anymore

Partikel filtering: Correction
Correct our prediction with what we see



- New particle weight: $w_t^{(i)} = p(I_t \mid \Theta_t^{(i)})$

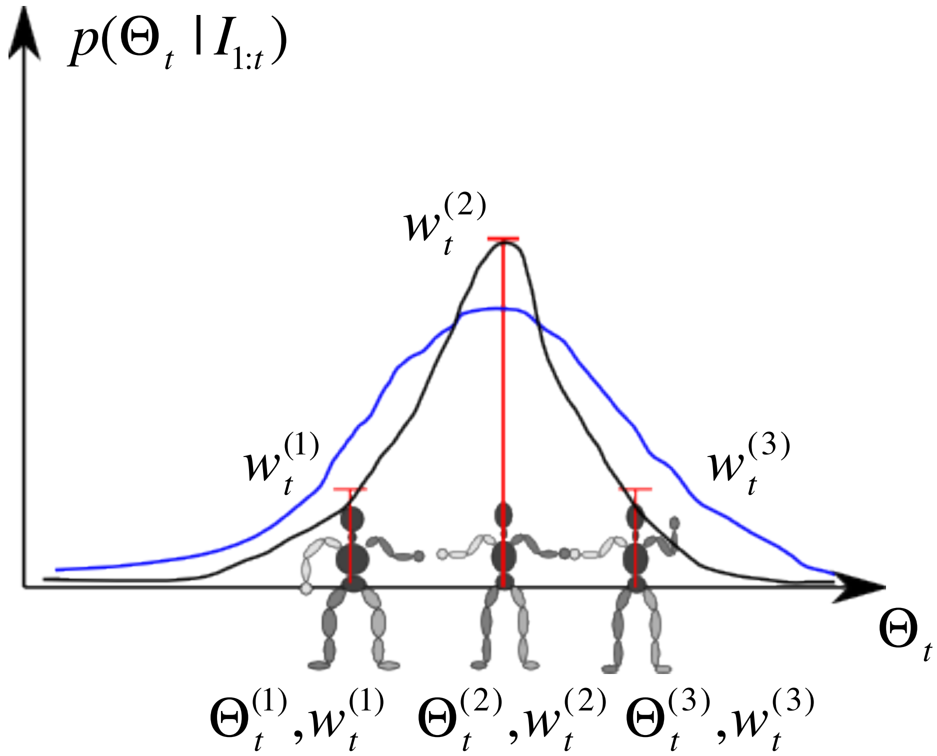
Tracking using particle filtering

How do we get our estimate of the pose?

1. Find MAP particle $\Theta_t^{(i)}, w_t^{(i)}$
   - Find the mode
2. Compute the weighted average

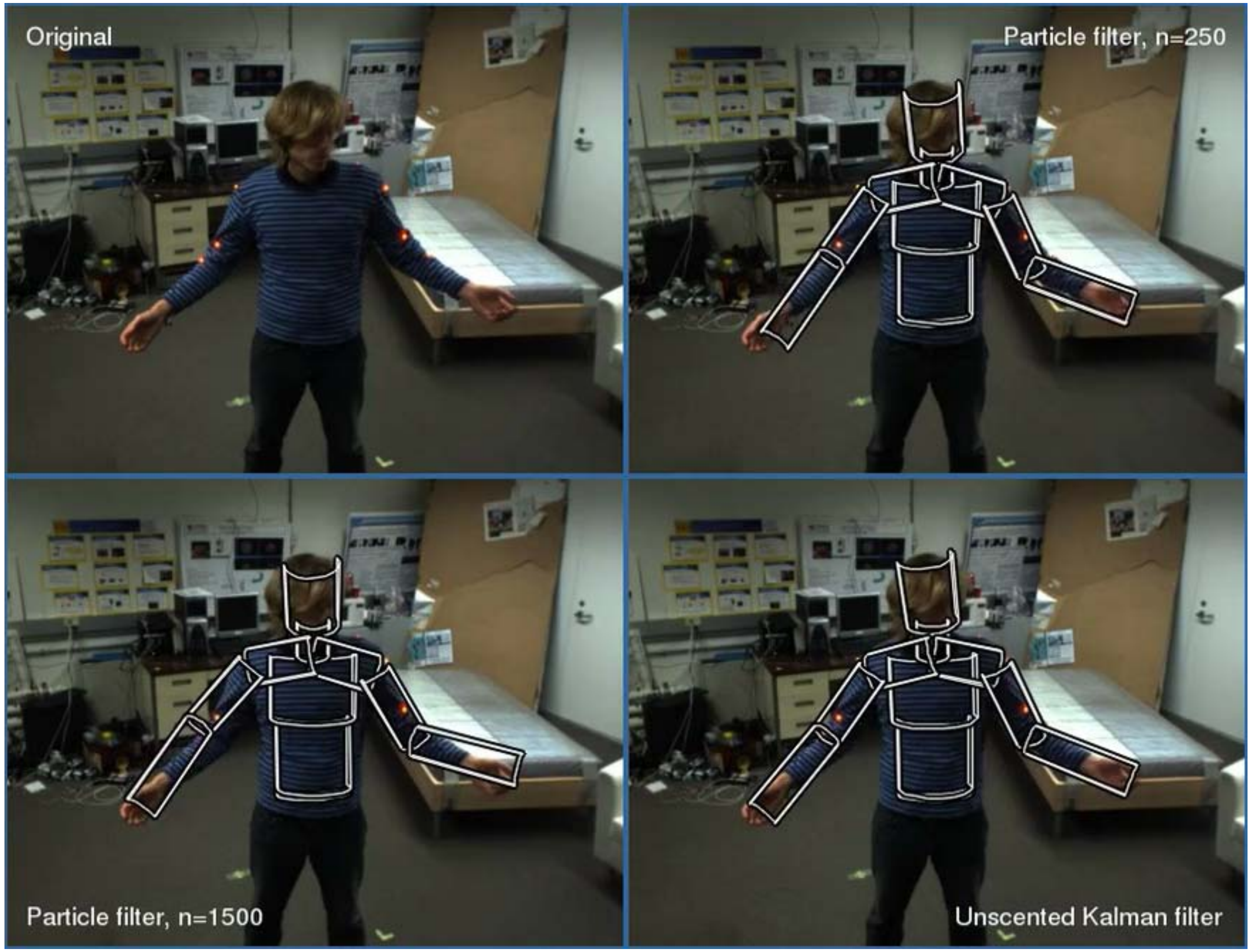
$$\hat{\Theta}_t = \sum_{i=1}^{N} w_t^{(i)} \Theta_t^{(i)}$$

3. And there are other possibilities …



Particle filter based solution: Using in the order of 100 particles. Transition probability is Gaussian

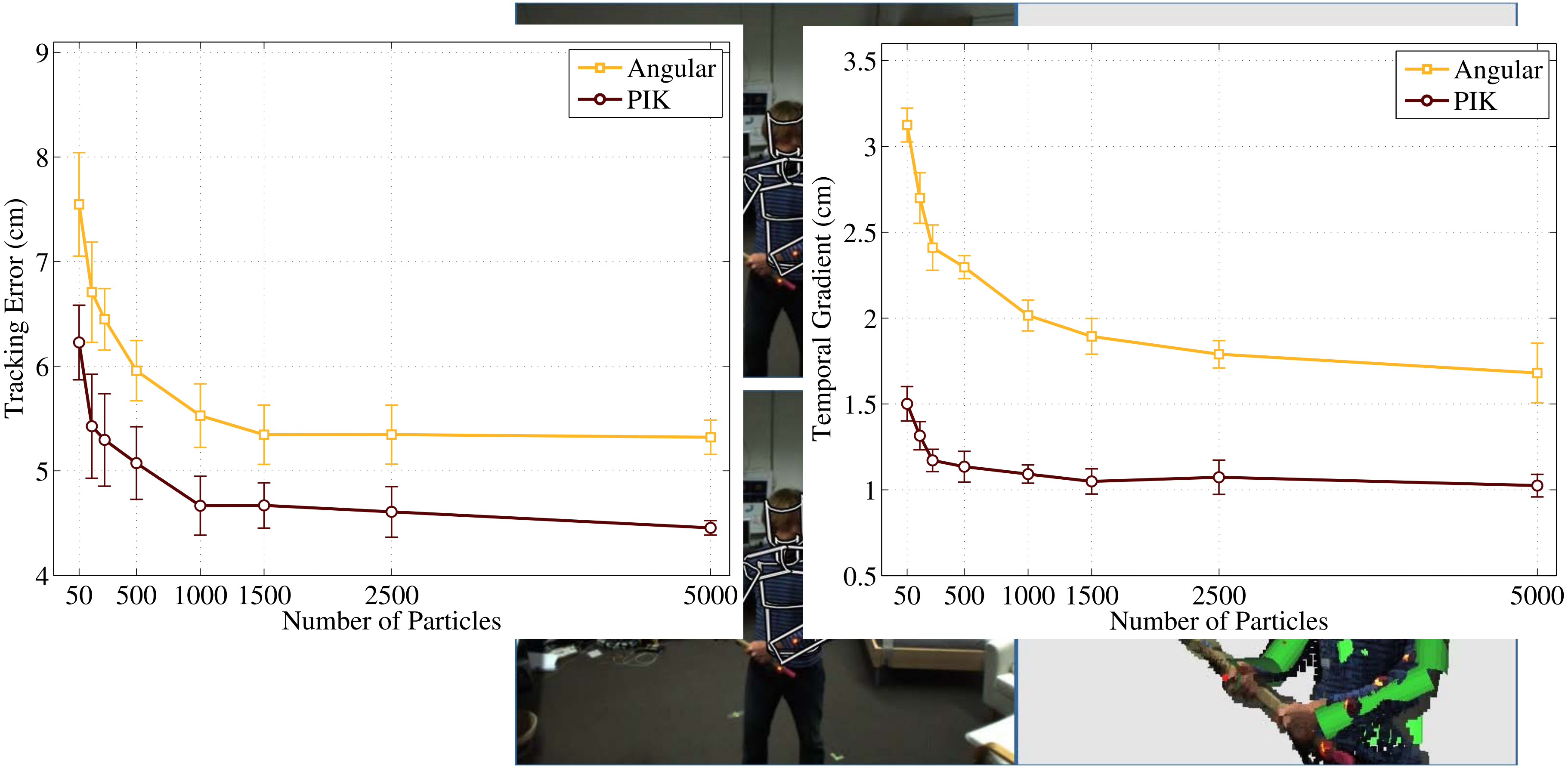
Particle filter versus unscented Kalman filter: Number of particles matter



Motion priors

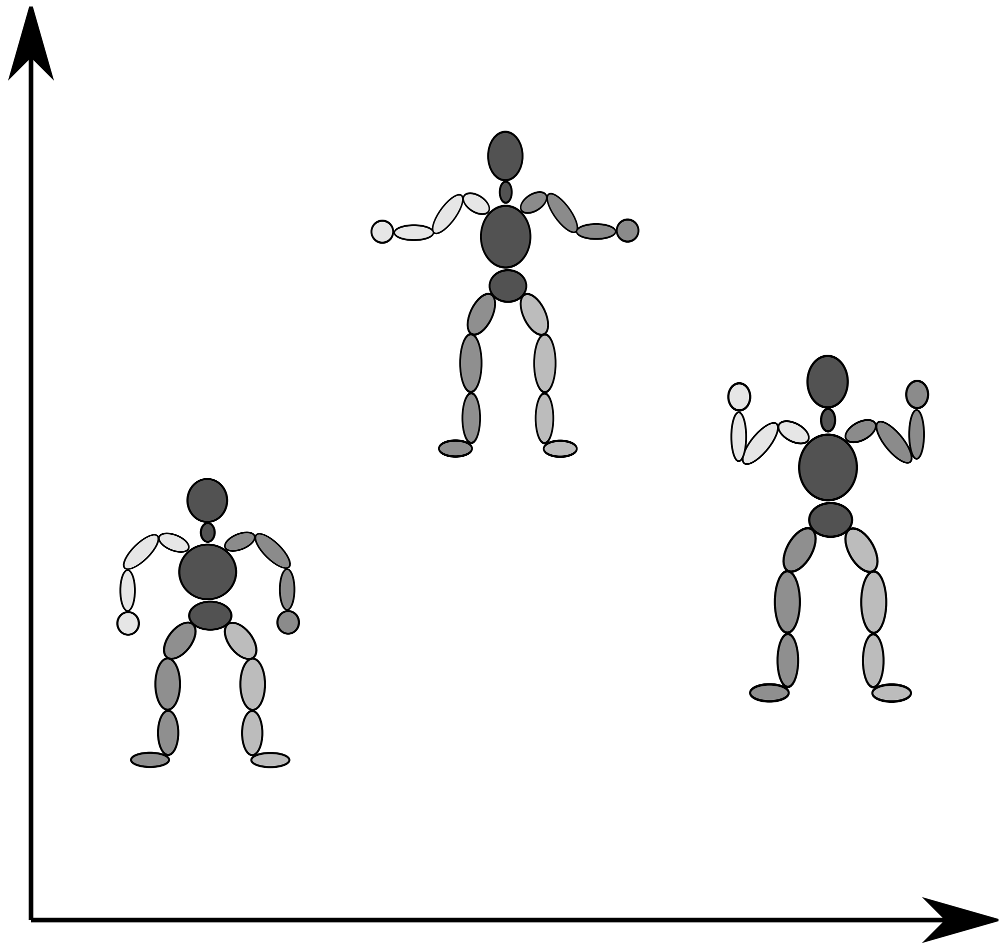
***S. Hauberg & K. S. Pedersen*: Predicting Articulated Human Motion from Spatial Processes. International Journal of Computer Vision, 94 (3): 317-334, 2011**

- Constraining tracking by adding kinematic priors on end-effector motion.



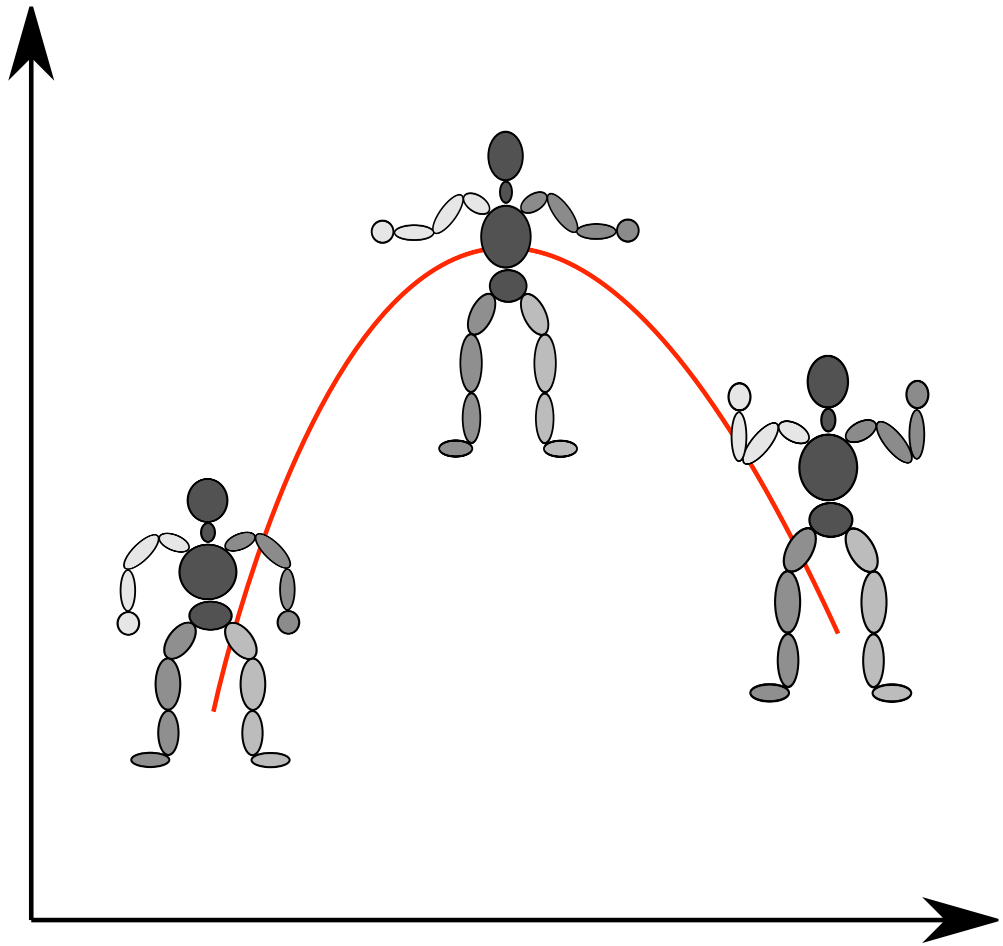
Modelling constraints and comparing poses

How hard can it be?



Modelling constraints and comparing poses

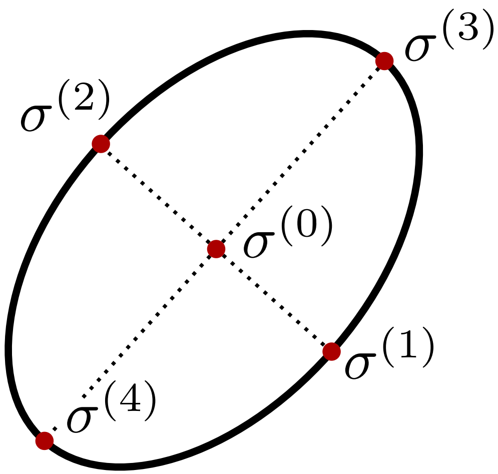
Natural constraints on motion and body poses form a Riemannian manifold.

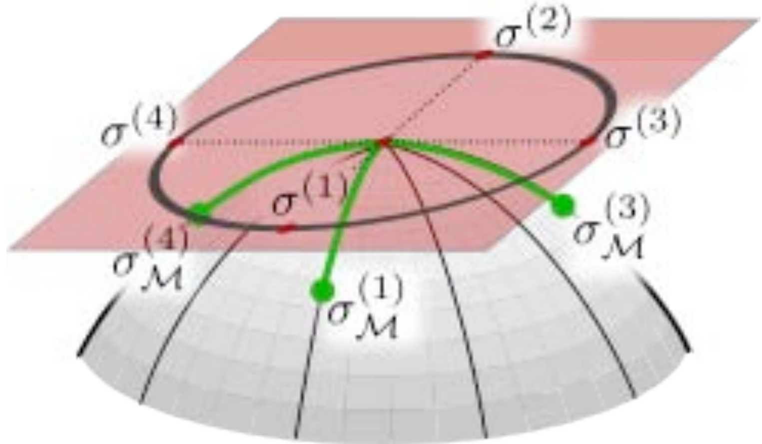


Manifold-valued Unscented Kalman Filtering

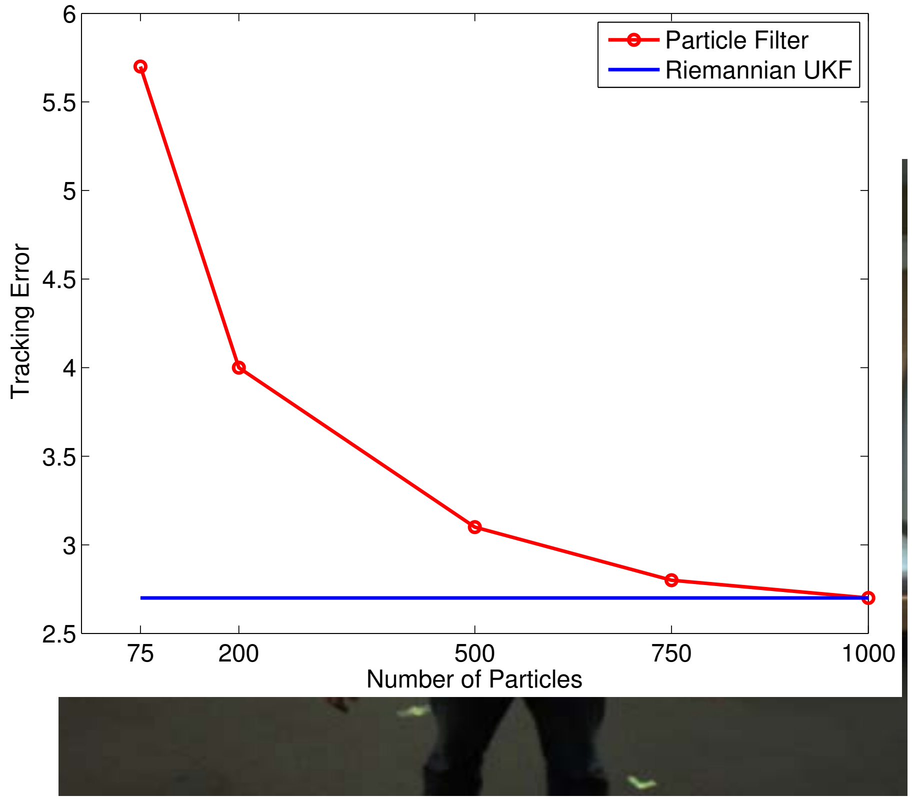
S. Hauberg, F. Lauze and K. S. Pedersen: **Unscented Kalman Filtering on Riemannian Manifolds. Journal of Mathematical Imaging and Vision, Vol: 46(1): 103-120, 2013**

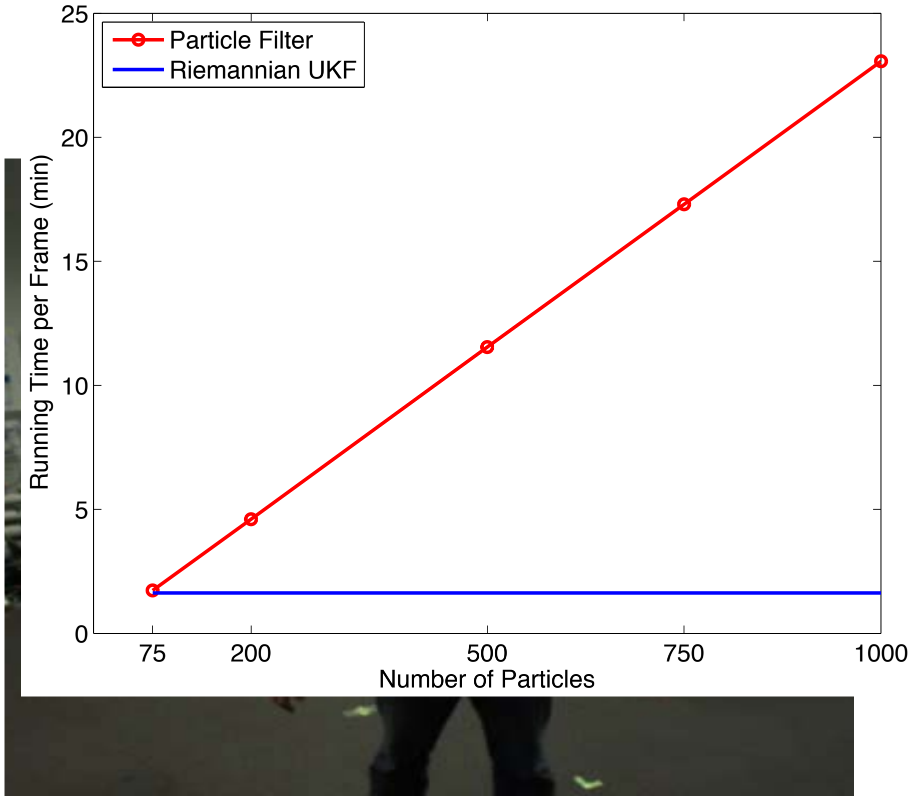
- Natural constraints on motion and body poses form a Riemannian manifold.
- We generalize unscented Kalman filtering to manifolds.





Manifold-Valued Unscented Kalman Filtering (MUKF)





Particle filtering benefits, necessities, problems:

- Benefit: Allows us to track states under general (multi-modal) probability distributions and with non-linear dynamics.

- Necessity: We need to be able to sample from the dynamical model / transition probability as well as evaluate the conditional observation probability density.

- Problems: The precision of the particle approximation of the filtering distribution is dependent on the number of particles. Unfortunately, this does not scale well with dimensionality of state space (curse of dimensionality).

Particle filter summary

- Particle filtering has its roots in Sampling Importance Resampling.
- Approximate distributions with a set of particles.
- Iterate through the prediction and correction steps.
- Bootstrap particle filter perform re-sampling of particles.
- Allows us to track states under general (multi-modal) probability distributions and with non-linear dynamics.
- Different variants of the particle filter exist.
- Convergence results exist for both Kalman and bootstrap particle filters.

Literature

- Bishop Ch. 13.3
- S. Thrun, W. Burgard, D. Fox: Probabilistic Robotics. The MIT Press, 2005. Ch. 4.
- O. Cappé, S. J. Godsill & E. Moulines: An Overview of existing methods and recent advances in sequential Monte Carlo. In IEEE Proceedings, 95(5): 899-924, May, 2007.

Interested in convergence results?

- D. Crisan and A. Doucet: A Survey of Convergence Results on Particle Filtering Methods for Practitioners. In IEEE Transaction on Signal Processing, 50(3): 736-746. 2002.

(All references available in Absalon under the menu item Course material)