G54ARS
Autonomous Robotic Systems
Lecture 9

Particle Filter

Last Week – Kalman Filter

- Basic concepts
  - filtering overview
  - system model
  - filter operation
- Worked example
  - model example
  - illustrative examples of operation
- Properties of Kalman filters
  - the optimal filter
  - misconceptions and myths
- Revision

Kahoot!

https://kahoot.it/

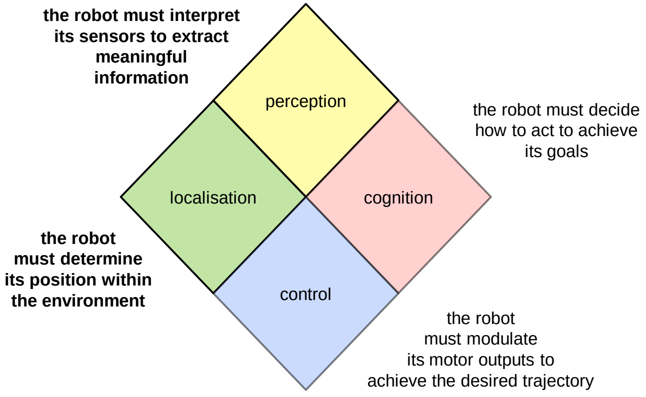
SET – SEM Results, Feedback and Discussion

Particle Filters - Overview

- Basic concepts
  - system model
  - outline algorithm
- Algorithmic detail
  - predict & update phases
  - resampling
- Worked example
  - model example
  - illustrative examples of operation

Basic Concepts

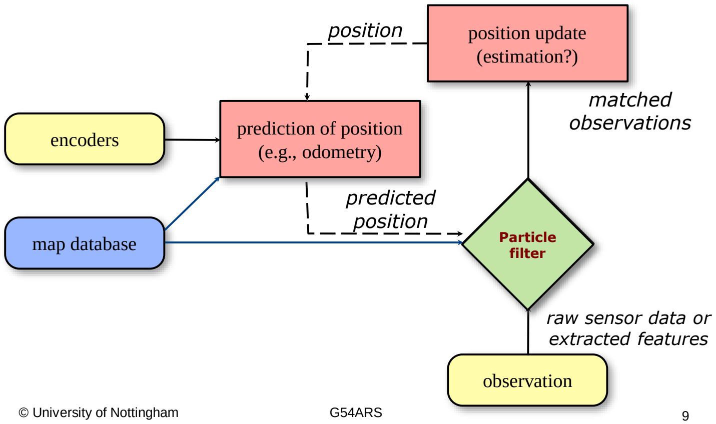
Navigation



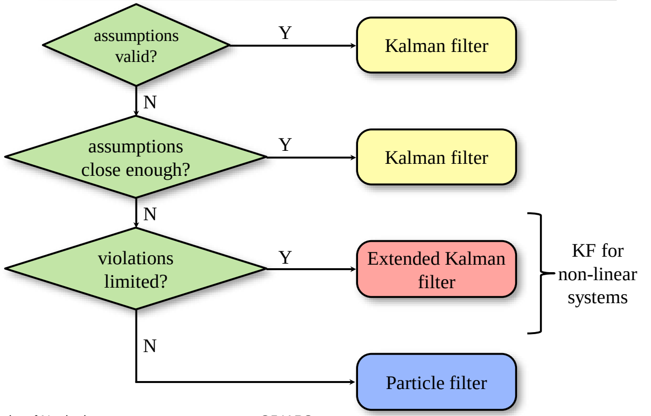
Definition

- A particle filter is a sample-based filter that sequentially estimates the state of a dynamic system from a series of noisy observations
  - introduced by Gordon, Salmond & Smith, 1993
- A PF is similar to a KF in that it
  - combines a prediction phase with an update phase
  - operates sequentially over discrete time-steps
- It differs from a KF, in that it
  - makes no assumptions of process/measurement linearity or noise characteristics
  - maintains a population of estimates

General Schematic



The Real World



System Model

- State, x

$$x \in \Re^n$$

- Non-linear process

$$x_k = f(x_{k-1}, u_k, w_k)$$

- Non-linear measurement model

$$z_k = h(x_k, v_k)$$

Population of Particles

- Each particle, s^i, consists of a state (possible robot pose), x^i, and an associated weight, ω^i, at each time, k

$$s_k^i = (x_k^i, w_k^i)$$

- There is a set, S, of N particles

$$S_k = \{s_k^1, \dots, s_k^N\}$$

Outline Algorithm

- **for** k = 1 ... t
  - **for** each particle // predict
    - calculate new state based on process model
  - **end for**
  - **for** each particle // update
    - calculate new weight based on measurement model
    - normalise weights (to sum to unity)
  - **end for**
  - calculate overall predicted position
  - **if** particles are insufficiently diverse
    - resample particles
  - **end if**
- **end for**

Algorithmic Detail

Predict Particles

- Predict the state of each particle, by moving it in state space according to the process model

$$x_k^i = f(x_{k-1}^i, u_k, w_k^i)$$

- Note that process noise, w_k, is included in each particle's state update
  - this is a sample drawn from the known process noise distribution (not necessarily Gaussian)
  - if unknown, then use Gaussian

Update Weights

- Update weight of each particle, by estimating the probability of the observed measurement assuming this particle is the actual position

$$\omega_k^i = \omega_k^i P(z_k | x_k^i)$$

- Calculate the residual

$$z_k - h(x_k^i, 0)$$

- Calculate the probability by considering the distribution of the measurement noise
  - if unknown, then assume Gaussian distribution

Determine Position

- Weighted mean

$$\hat{x}_k = \sum_{i=1}^{N} \omega_k^i x_k^i$$

- Best particle

$$\hat{x}_k = x_k^i \quad | \; \omega_k^i = \max_i \omega_k^i$$

- Robust mean
  – weighted mean of particles within given distance of best

$$\hat{x}_k = \sum_{i=1}^{N} \omega_k^i x_k^i : |x_k^i - x_k^{best}| \leq \epsilon$$

Resample

- A common problem is that the population may quickly converge to a single state
  – one particle with weight one; others zero
- Calculate the effective sample size
  – usually termed *ESS* or N_{eff}

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} (\omega_k^i)^2}$$

- If N_{eff} falls below a threshold (often $N/5$), then resampling is performed

– pick random new population, in proportion to weights

Systematic Resampling Algorithm

$$
\begin{array}{l}
S' = \emptyset \\
\Delta = \mathrm{rand}(0, N^{-1}] \\
c = \omega_k^1 \\
i = 1 \\
\textbf{for } j = 0 \ldots N-1 \\
\quad u = \Delta + j/N \\
\quad \textbf{while } u > c \\
\quad\quad i = i + 1 \\
\quad\quad c = c + \omega_k^i \\
\quad \textbf{end while} \\
\quad S' = S' \cup \{(x_k^i, 1/N)\} \\
\textbf{end for}
\end{array}
$$

Systematic Resampling Algorithm

Worked Example, e.g.:

$$S = \{(x_k^1, w_k^1 = 0.1), (x_k^2, w_k^2 = 0.2), (x_k^3, w_k^3 = 0.7)\}$$

$S' = ?$

Implementation

- Number of particles
  - more particles ⇒ better approximation
  - more particles ⇒ more computation
- PFs approach the Bayesian optimal estimate
  - if number of particles is sufficiently large!
- Many detailed variations
  - bootstrap particle filter
  - auxiliary sampling importance resampling
  - regularised particle filter
  - local linearisation particle filter
  - multiple-model particle filter

Worked Example

Example System

- Mobile robot with pose $\mathsf{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$
- Moving with speed s in direction Θ $\mathsf{u} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix}$

$$\mathsf{x_k} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}}_{x_{k-1}} + \underbrace{\begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix}}_{u} + \mathsf{w_k}$$

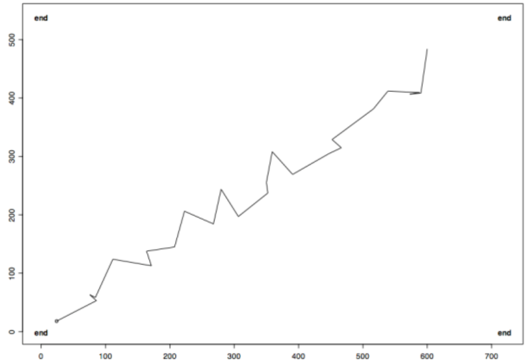
$$\mathsf{z_k} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{H} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \mathsf{v_k}$$

Covariances

$$\mathsf{Q} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix}$$

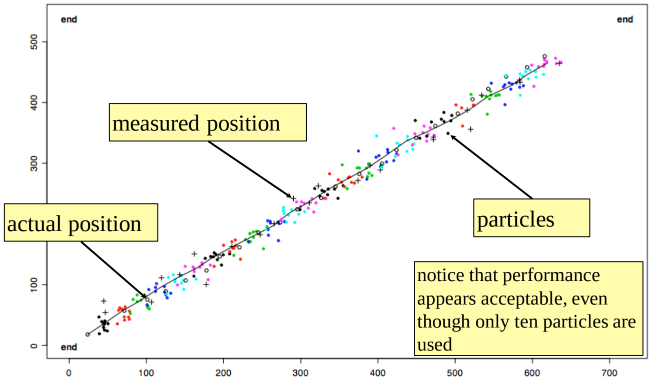
$$\mathsf{R} = \begin{bmatrix} \sigma_{x_m}^2 & 0 \\ 0 & \sigma_{y_m}^2 \end{bmatrix}$$

Unfiltered

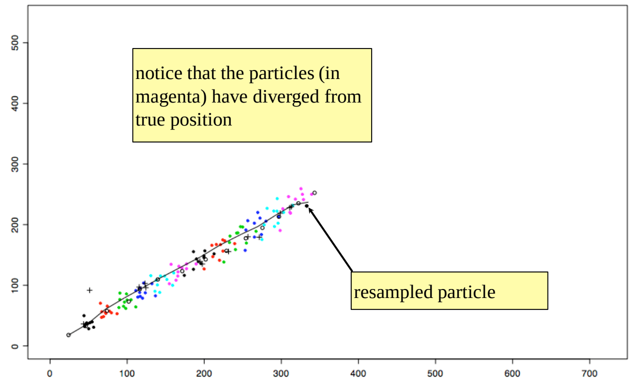


$\sigma_x^2 = \sigma_y^2 = 4$ $\sigma_{x_m}^2 = \sigma_{y_m}^2 = 16$

Particle Filter



An Example of Resampling



Worked Example

Time	Position	Weight	Weighted Position	Best Weight
t = 1	(4,2)	0.2		
	(3,3)	0.2		
	(8,4)	0.2		
	(5,4)	0.3		
	(7,6)	0.1		
POSITION ESTIMATE:				
t = 2	(11,10)	0.4		
	(12,11)	0.3		
	(9,12)	0.0		
	(10,12)	0.1		
	(12,9)	0.2		
POSITION ESTIMATE:				

- Calculate the estimate of the position at both times, using both the *weighted mean* method and the *best particle* method
- Suppose that the robust mean method, with a distance limit of 2.5, was used at time t=1. Which particles would now contribute to the estimate of position?

Properties of Particle Filters

- While it's not entirely accurate to state that PFs have no underlying assumptions, they can be used in very many (all?) situations
  - particle states must be predicted
  - noise must be added to process
  - an estimate of measurement errors required
- However, they are (far) more computationally expensive than Kalman filters (and EKFs)
  - if KF assumptions are valid, then they are better

Summary

- Particle Filters
  - basic concepts
    - system model
    - outline algorithm
  - algorithmic detail
    - predict & update phases
    - resampling
  - worked example
    - model example
    - illustrative examples of operation
- Next: Revision

Revision

Revision Overview

- Module Assessment Review
- Exam Structure
- Brief Revision of Topics

Review of Module Assessment

- Examination – 50%
  - Covering all material covered in lectures and reading week. (This includes scientific papers, articles, videos, etc.)
- Coursework – 50%
  - Working in teams throughout term
  - 35% Laboratory assignment & individual associated report
  - 15% Five lab sheets/demos per team (3% each)
  - The lab is only accessible during your allocated lab and practice sessions.
    - i.e. **attend and prepare your sessions!**

G54ARS Exam Structure

- Time: 2 hours
  - *Only silent, self-contained calculators with a Single-Line Display or Dual-Line Display are permitted.*
  - *No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.*
  - No dictionaries (except standard dictionary if English is not your first language)
- Questions:
  - All questions are mandatory

G54ARS Review of Topics

Updated Weekly Topics - 1

Week	Lecture
2	Introduction & Overview (Autonomous) Robots =? Foundations of Robotic Systems Architectures & Behaviours
3	Brooks' Subsumption Architecture - Theory Robot Hardware
4	DARPA Grand Challenge PID Control
5	Fuzzy Control
6	Ultrasonic Sensor Models Localisation and Mapping

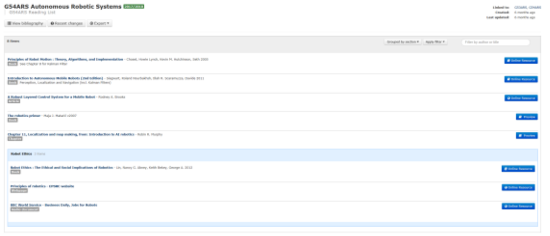
Updated Weekly Topics - 2

Week	Lecture
7	Sensor Fusion
8	Guest Lecture: Dr Barbara Bruni, University of Genova
9	Reading week (Ethics and Robotics) **Lab will be open in lecture slot**
10	Kalman Filters
11	Particle Filters & Revision

Note:
Weekly topics may change subject to timing and new material.
All lecture notes will be available online.

Note on Reading List

- Reading List on Moodle
- Core reading Week materials
- Other sources – for better understanding



Exam content

- Covering **all** material covered in lectures **and** reading week
- General pointers:
  - Focus on understanding and being able to explain concepts and key terms.
  - Be ready to explain challenges in Autonomous Robotic Systems.
  - Remember and be able to explain key equations.
  - Practise key calculations (no continuous integration/derivatives).

Exam - tips

- All parts of the exam are compulsory
  - Do NOT study selectively
- Time is short (2 hours)
  - Keep your answers concise – no need for “essays”
- Use diagrams where useful/needed
  - Explain and refer to the diagrams from your written answer
  - Annotate diagrams

Q & A

Next Week:
No lecture – Send revision Questions by email the end of Wednesday 13th December 2017

