Particules Filter: a robot localization application

OutLine

- ❑ Introduction
- ❑ Basics
- ❑ Motion and Sensors Model

Inspired by :

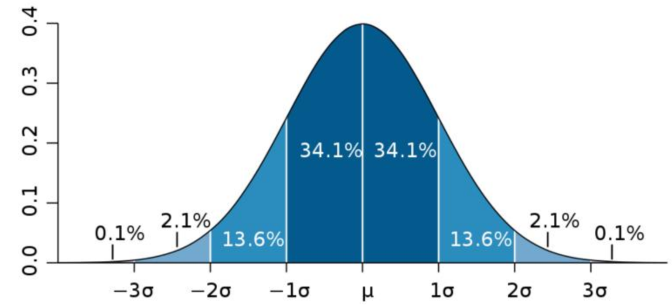
Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

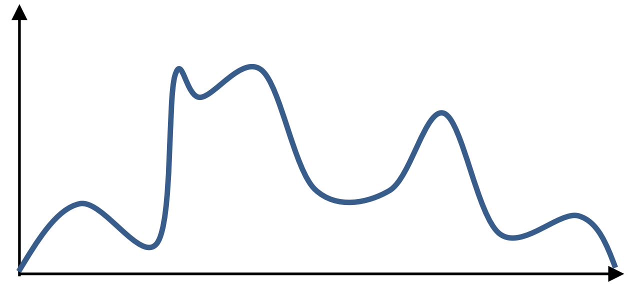
Probabilistic Robotics, Sebastian THRUN, Wolfram BURGARD, Dieter FOX, 2000

Particules filter, What is the need?

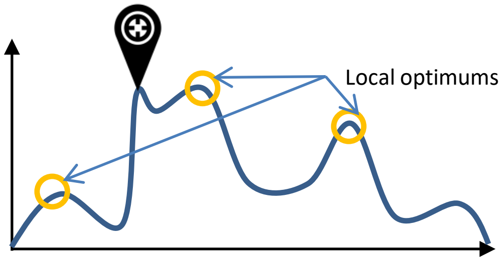
Needs

- ❑ Model distributions
  - ❑ For Gaussian distribution
    - → Kalman Filter
  - ❑ For Arbitrary distribution
    - → ?

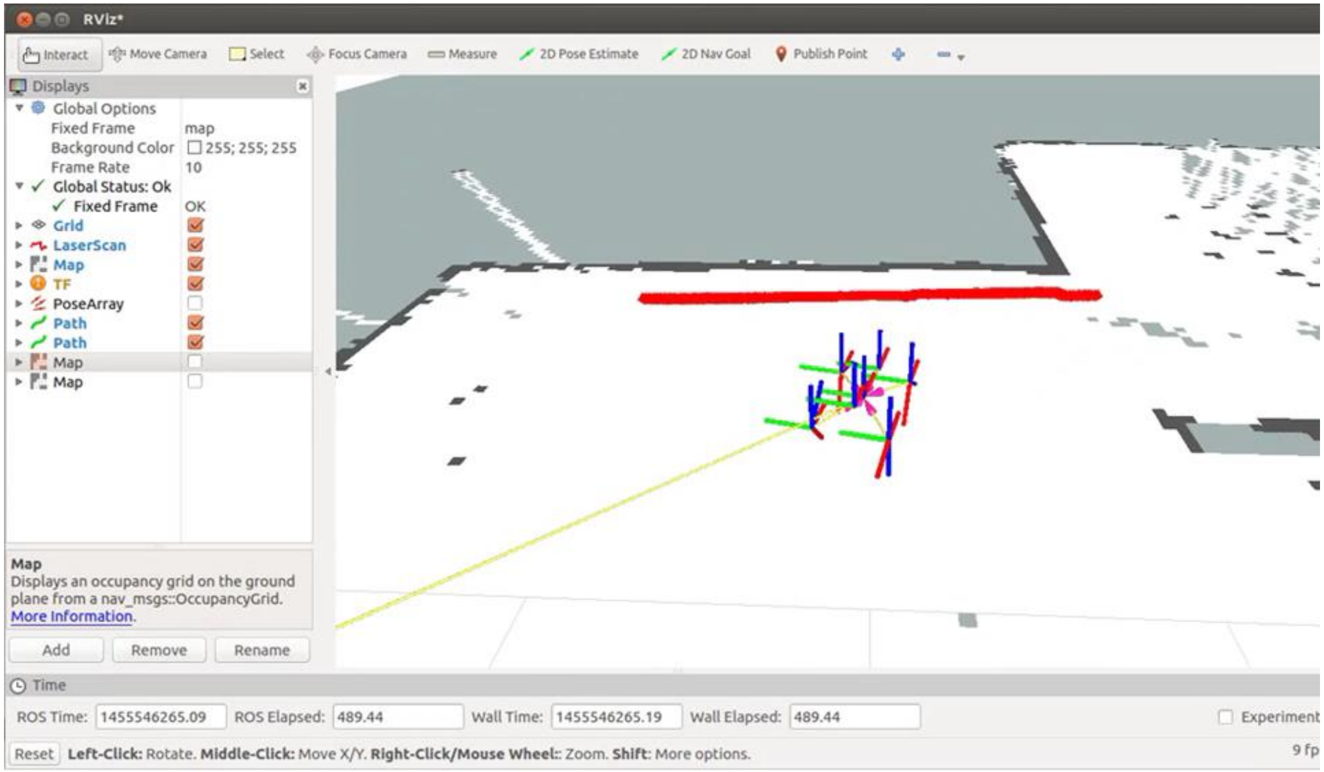


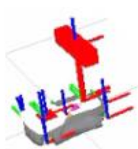


- ❑ Application:
  - ❑ Robot localization
  - ❑ Function optimum finding



Needs



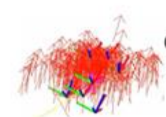

Robot Model


Laser trace


Occupancy Map

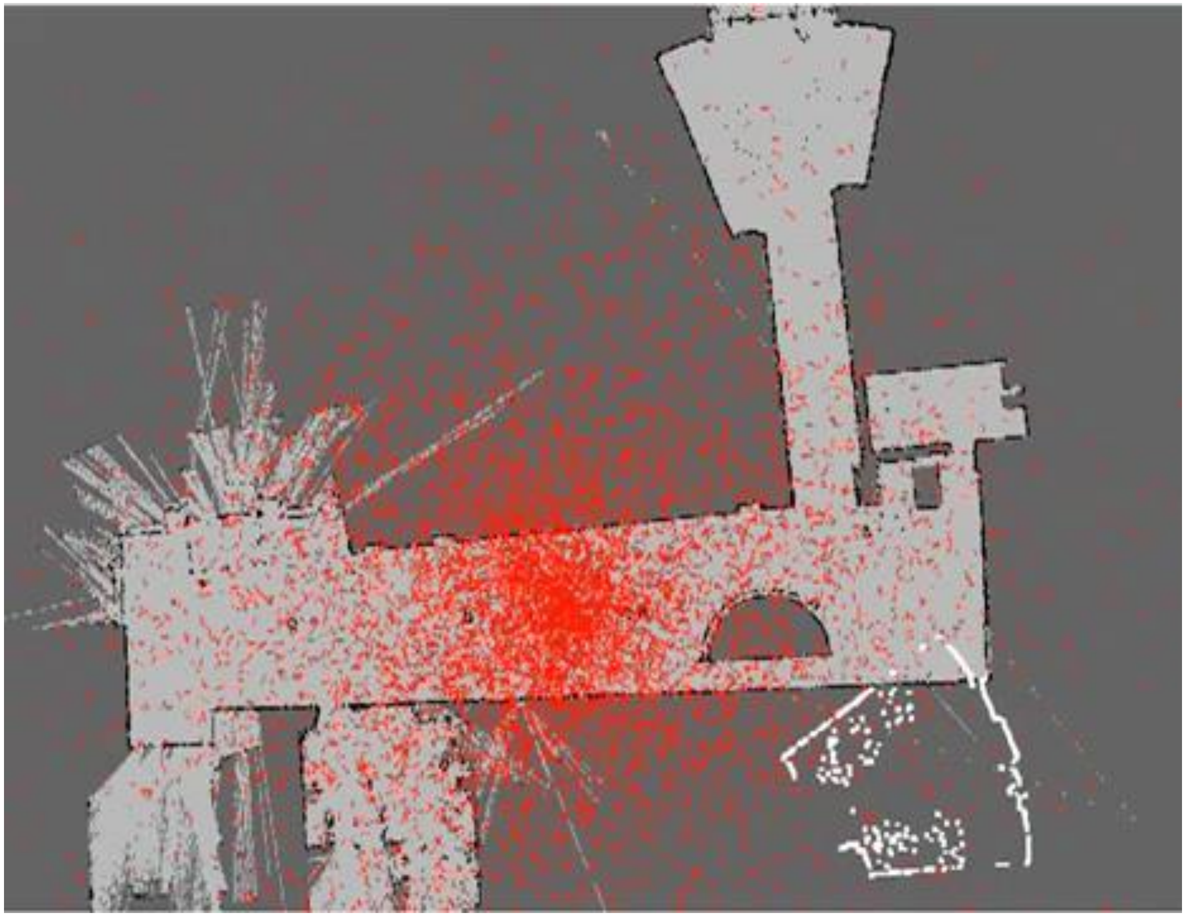

Global Cost map


Local Cost map


Cloud Particles : Probable robot location

http://cpe-dev.fr/navigation-test/

Needs

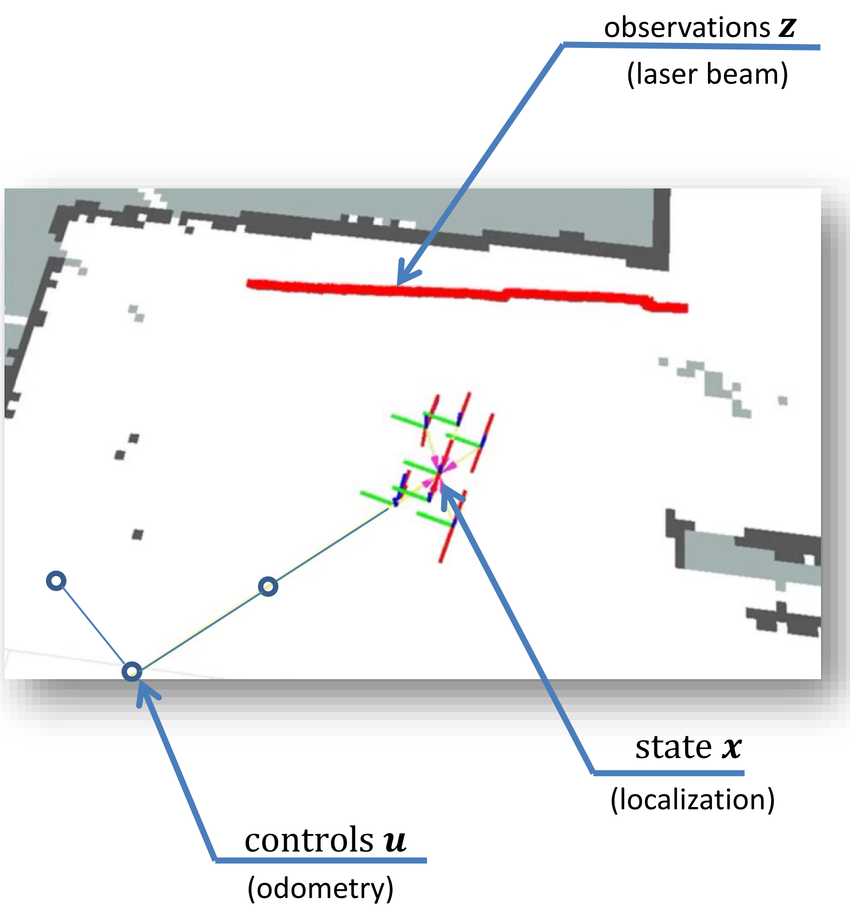


https://www.youtube.com/watch?v=OVoa11xd3vE

Needs

- How to known a system state $\boldsymbol{x}$:
  - Given observations $\boldsymbol{z}$
  - Given controls $\boldsymbol{u}$

$$p(x \mid z, u)$$



Recursive Bayes Filter

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

$$bel(x_t) = \underline{p(x_t \mid z_{1:t}, u_{1:t})}$$

Definition of the belief

Recursive Bayes Filter

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

$$bel(x_t) = p(x_t | z_{1:t}, u_{1:t})$$

$$bel(x_t) = \frac{p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t \mid z_{1:t-1}, u_{1:t})}{p(z_t \mid z_{1:t-1}, u_{1:t})}$$

$\xrightarrow{\text{Bayes Law}}$ $p(A|B) = \frac{p(B|A)p(A)}{p(B)}$

$$= \eta \, p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t \mid z_{1:t-1}, u_{1:t})$$

$$= \eta \, p(z_t | x_t) \, p(x_t \mid z_{1:t-1}, u_{1:t})$$

$\xrightarrow{\text{Markov assumption}}$ $p(z_t | x_t, z_{1:t-1}, u_{1:t}) \equiv p(z_t | x_t)$

$$= \eta \, p(z_t | x_t) \, p(x_t \mid z_{1:t-1}, u_{1:t})$$

Recursive Bayes Filter

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

$$bel(x_t) = p(x_t | z_{1:t}, u_{1:t})$$

$$bel(x_t) = \frac{p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t \mid z_{1:t-1}, u_{1:t})}{p(z_t | \, z_{1:t-1}, u_{1:t})}$$

$$= \eta \, p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t \mid z_{1:t-1}, u_{1:t})$$

$$= \eta \, p(z_t | x_t) \, p(x_t \mid z_{1:t-1}, u_{1:t})$$

Law of total probabilities $\downarrow$

$$P(A) = \int P(A|B).P(B) \, dB$$

$$= \eta \, p(z_t | x_t) \int p(x_t \mid \underline{x_{t-1}}, z_{1:t-1}, u_{1:t}) \, p(\underline{x_{t-1}} \mid z_{1:t-1}, u_{1:t}) \underline{dx_{t-1}}$$

Recursive Bayes Filter

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

$$bel(x_t) = p(x_t | z_{1:t}, u_{1:t})$$

$$bel(x_t) = \frac{p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t \mid z_{1:t-1}, u_{1:t})}{p(z_t | \, z_{1:t-1}, u_{1:t})}$$

$$= \eta \, p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t \mid z_{1:t-1}, u_{1:t})$$

$$= \eta \, p(z_t | x_t) \, p(x_t \mid z_{1:t-1}, u_{1:t})$$

$$= \eta \, p(z_t | x_t) \int p(x_t \mid x_{t-1}, \underline{z_{1:t-1}, u_{1:t}}) \, p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) dx_{t-1}$$

Markov Assumption

$$= \eta \, p(z_t | x_t) \int p(x_t \mid x_{t-1}, \underline{u_t}) \, p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) dx_{t-1}$$

Recursive Bayes Filter

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

$$bel(x_t) = p(x_t | z_{1:t}, u_{1:t})$$

$$bel(x_t) = \frac{p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t \mid z_{1:t-1}, u_{1:t})}{p(z_t \mid z_{1:t-1}, u_{1:t})}$$

$$= \eta\, p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t \mid z_{1:t-1}, u_{1:t})$$

$$= \eta\, p(z_t | x_t)\, p(x_t \mid z_{1:t-1}, u_{1:t})$$

$$= \eta\, p(z_t | x_t) \int p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t})\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) dx_{t-1}$$

$$= \eta\, p(z_t | x_t) \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, \underline{u_{1:t}}) dx_{t-1}$$

Markov Assumption

Future command u_t as no influence on current state x_{t-1}

$$= \eta\, p(z_t | x_t) \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, \underline{u_{1:t-1}}) dx_{t-1}$$

CPE LYON
École supérieure de chimie physique électronique de Lyon

Recursive Bayes Filter

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

$$bel(x_t) = p(x_t | z_{1:t}, u_{1:t})$$

$$
\begin{aligned}
bel(x_t) &= \frac{p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t \mid z_{1:t-1}, u_{1:t})}{p(z_t \mid z_{1:t-1}, u_{1:t})} \\
&= \eta\, p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t \mid z_{1:t-1}, u_{1:t}) \\
&= \eta\, p(z_t | x_t)\, p(x_t \mid z_{1:t-1}, u_{1:t}) \\
&= \eta\, p(z_t | x_t) \int p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t})\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) dx_{t-1} \\
&= \eta\, p(z_t | x_t) \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) dx_{t-1} \\
&= \eta\, p(z_t | x_t) \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1}) dx_{t-1}
\end{aligned}
$$

$$\boxed{bel(x_t) = \eta\, p(z_t | x_t) \int p(x_t \mid x_{t-1}, u_t)\, bel(x_{t-1}) dx_{t-1}}$$

Recursive Bayes Filter

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

$$bel(x_t) = \eta \, \underbrace{p(z_t | x_t)}_{\text{OBSERVATION}} \int \underbrace{p(x_t \,| x_{t-1}, u_t)}_{\text{MOTION / CONTROL}} \, bel(x_{t-1}) dx_{t-1}$$

OBSERVATION

→ Correction step

MOTION / CONTROL

→ Prediction step

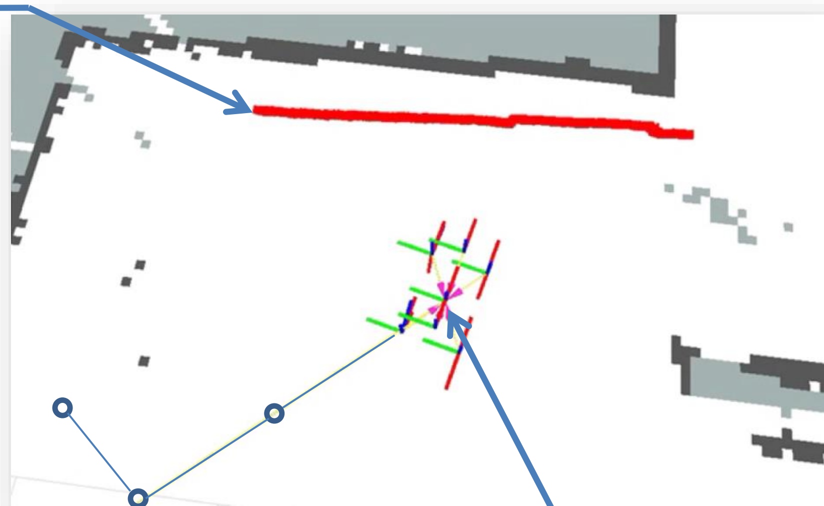
Recursive Bayes Filter

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

$$bel(x_t) = \eta \, \underbrace{p(z_t|x_t)}_{\text{OBSERVATION}} \int \underbrace{p(x_t \,|x_{t-1}, u_t)}_{\text{MOTION / CONTROL}} \, bel(x_{t-1}) dx_{t-1}$$

observations $\boldsymbol{z}$

(lazer beam)



controls $\boldsymbol{u}$

(odometry)

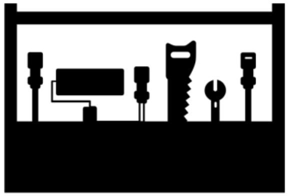
state $\boldsymbol{x}$

(localization)

Bayes Filter

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

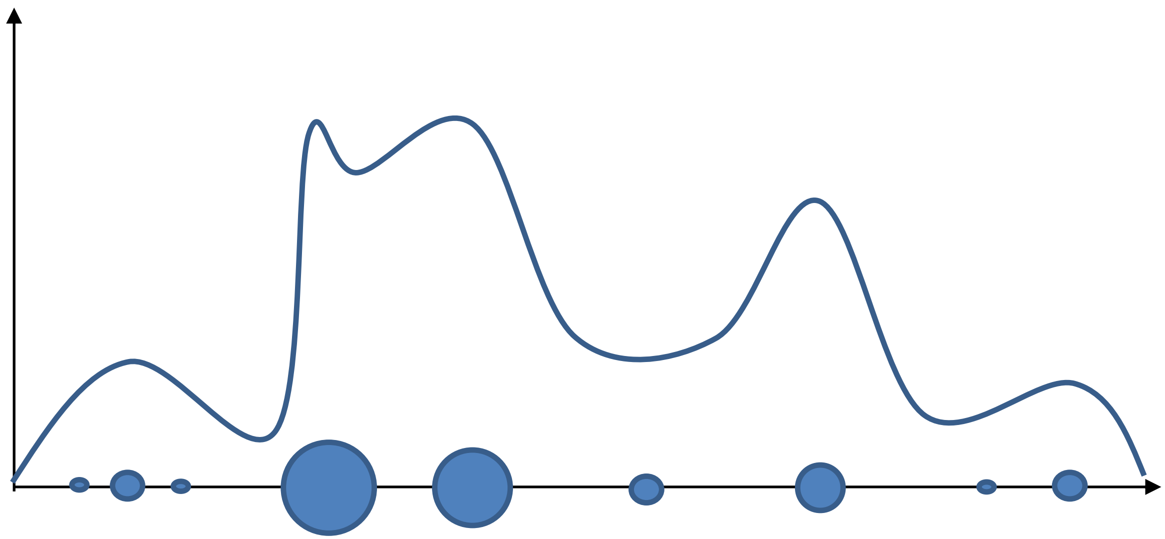
- Framework (template) for recursive state estimation
- Different possible instances depending:
  - **Model** for motion/control and observation (**linear vs non-linear**)
  - **Parametric vs non-parametric** filter
  - Dealing with **Gaussian distribution or not**



(odometry)

Particules Filter : Basics

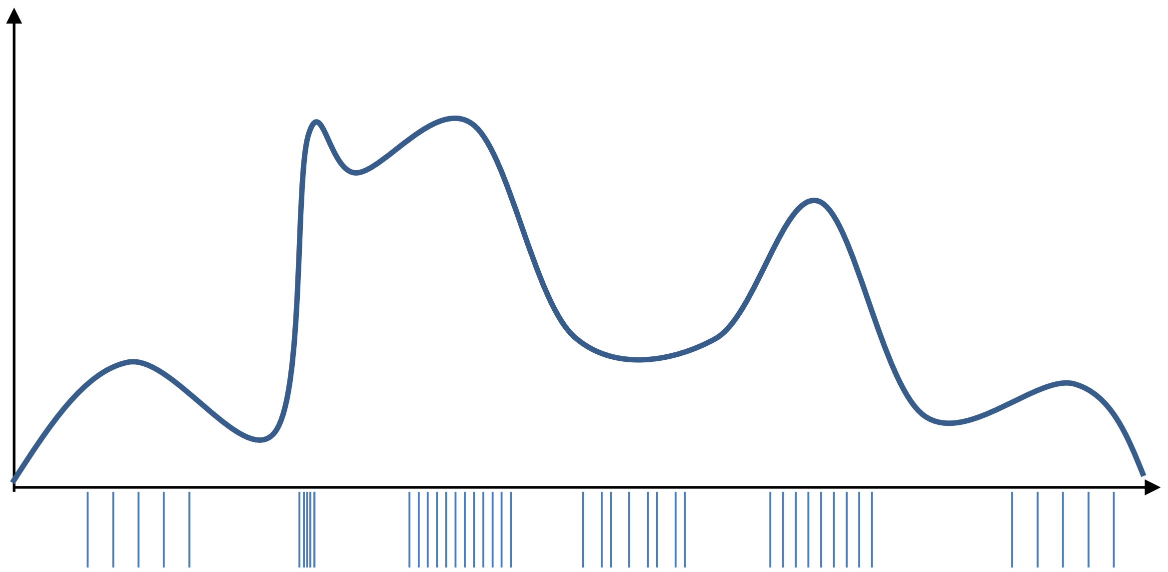
Principles



1

Set of **samples** (particules) **are distributed** across the environnment. **Each sample is weighted** according to the distribution

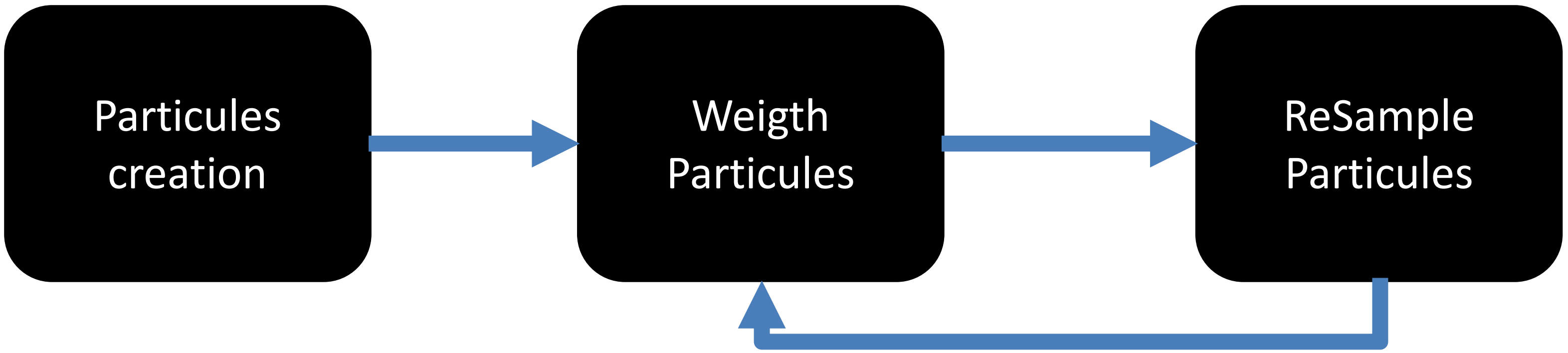
Principles



2

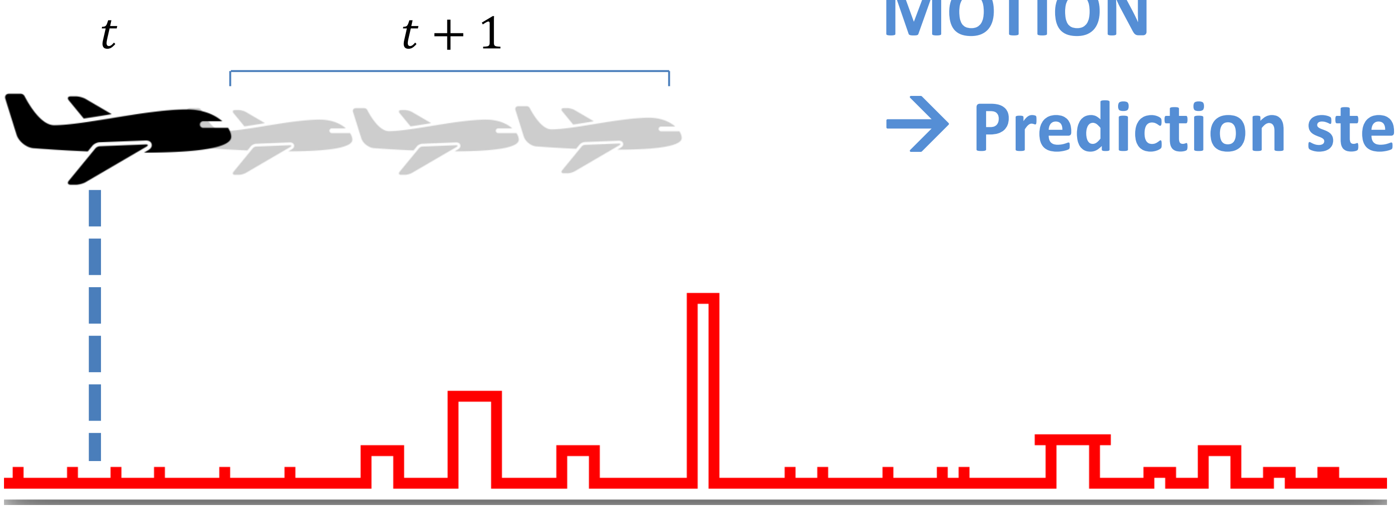
The **heigher** the sample **weight** is, the more particules are **distributed around**

Principles



How to obtain such samples ?

Example



MOTION

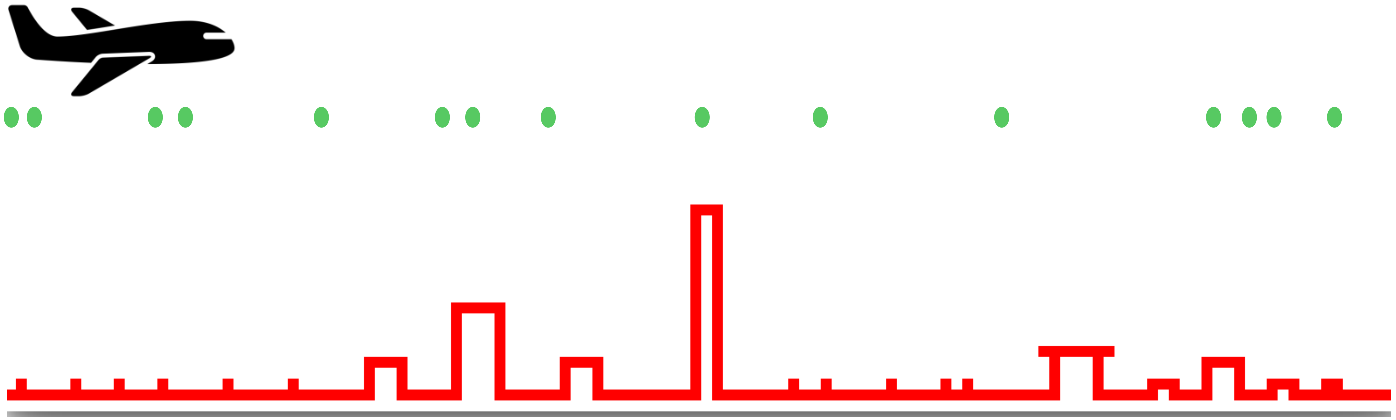
→ Prediction step

OBSERVATION

→ Correction step

1

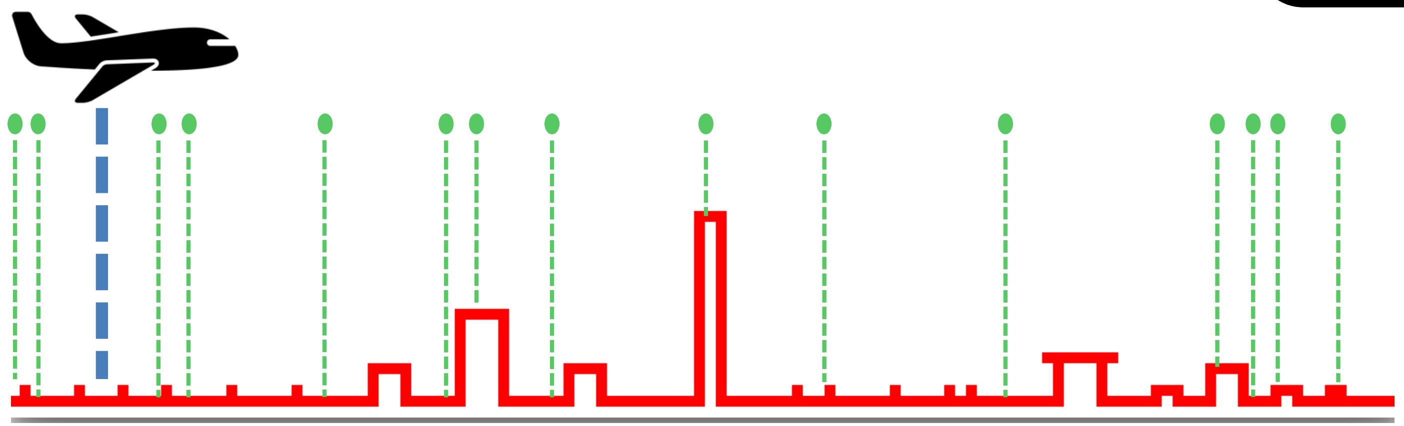
Particules Creation



If no initial knowledge, uniform distribution of the particules accross the environment

2

Weight Particules

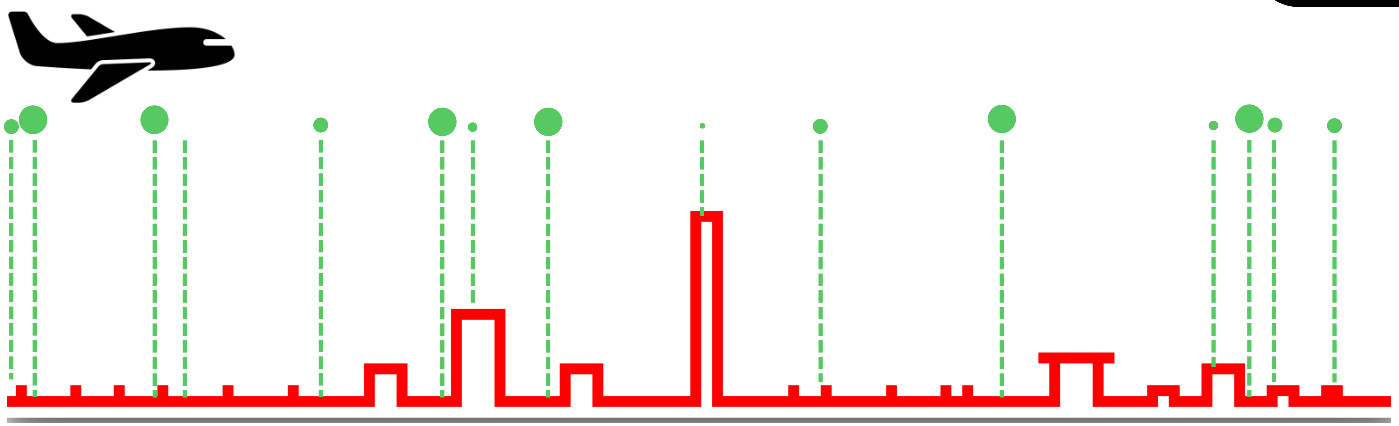


Weight particules according the current **observation** and the **estimate observation** at the particule position

Observation models is **combined** with an **Error model** (Gaussian error model is commonly used)

2

Weight
Particules

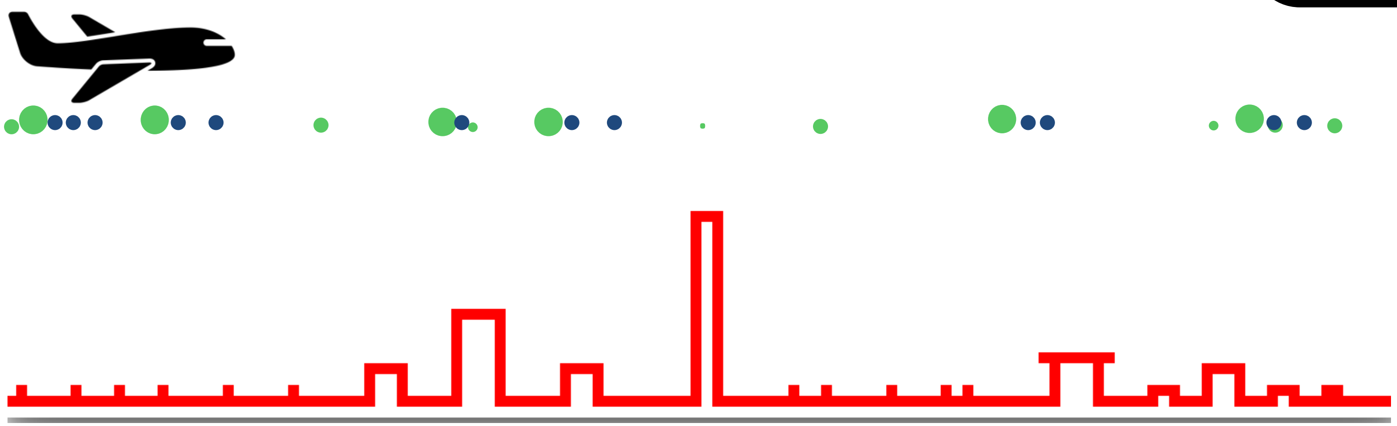


Weight particules according the current **observation** and the **estimate observation** at the particule position

Observation models is **combined** with an **Error model** (Gaussian error model is commonly used)

3

Resample Particules

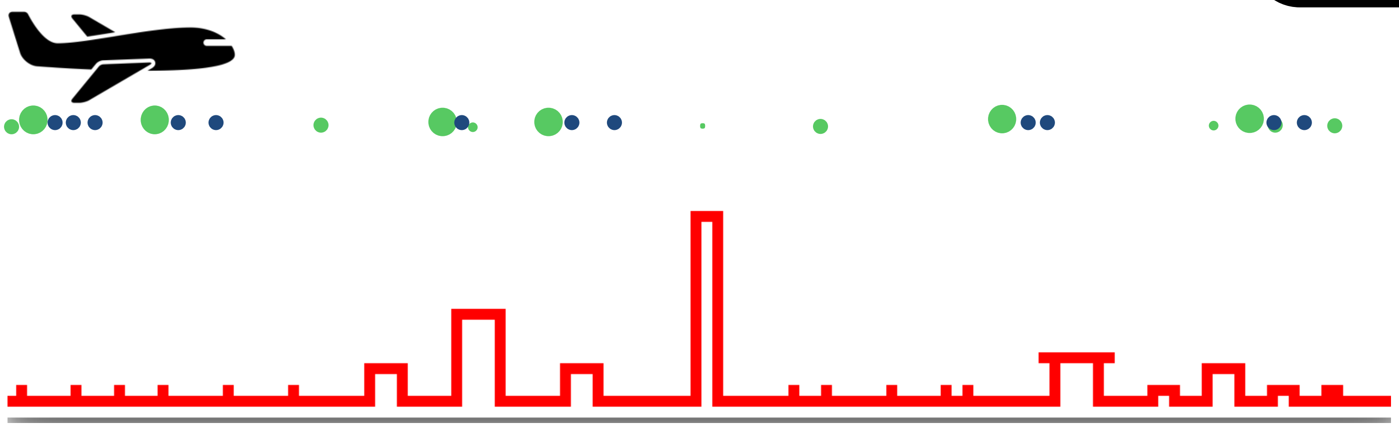


Resample Particules **according** their **weight.**

Spread new **particules** close **to heigher weighted particules according** to **a motion model** (also combined with an error model)

3

Resample Particules



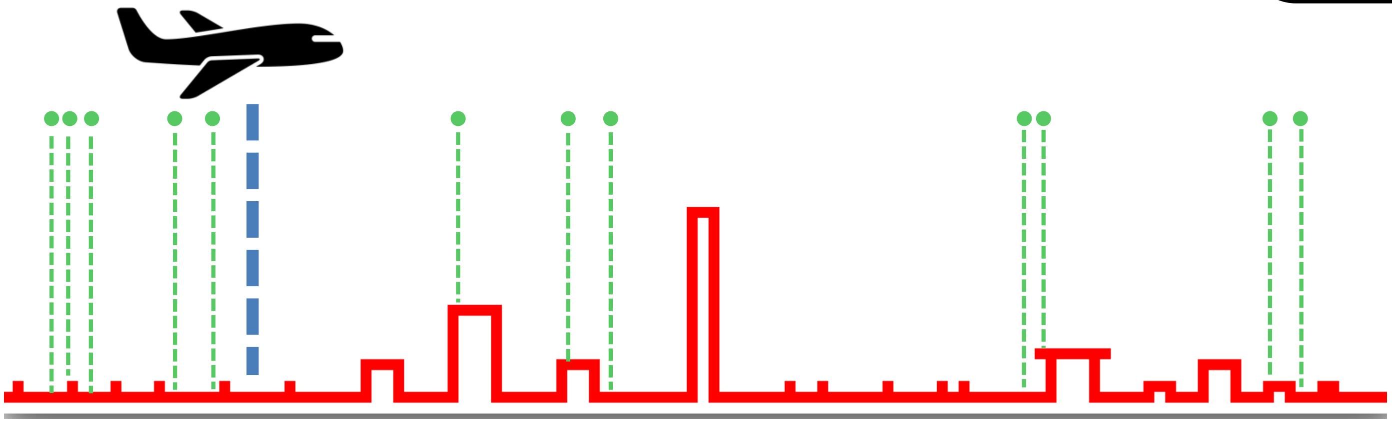
Resample Particules **according** their **weight.**

Spread new **particules** close **to heigher weighted particules according** to **a motion model** (also combined with an error model)

Remove old particules

X

Iterate

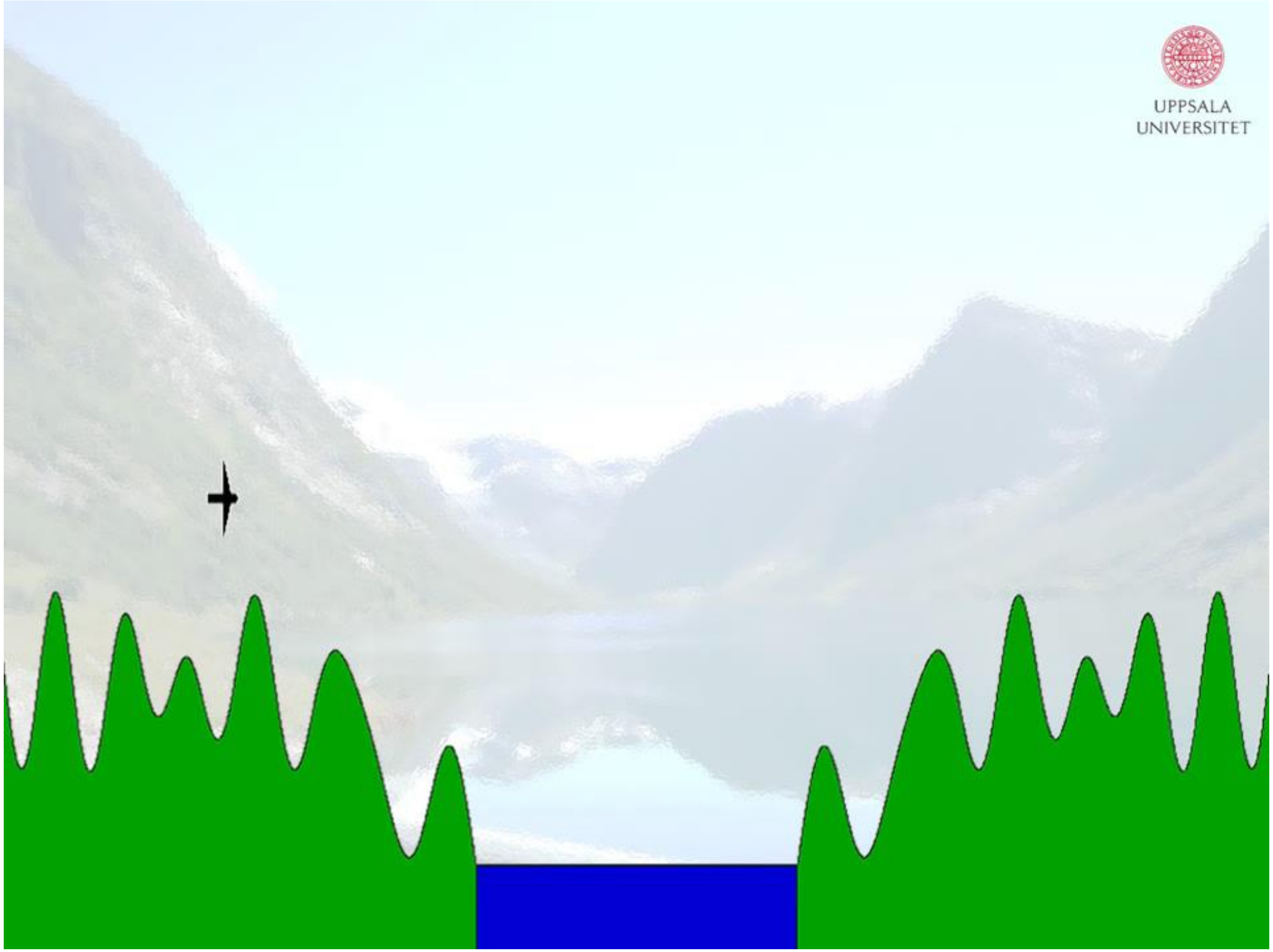


Make Observation at t+1 (plane move)

Weight Particules

Remove old Particules

Example

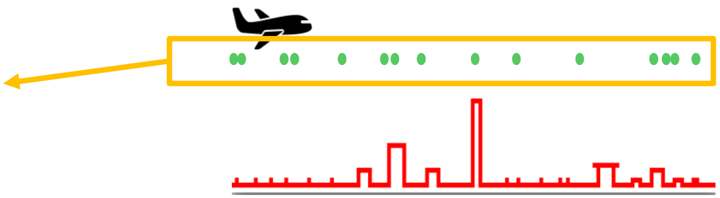


https://www.youtube.com/watch?v=aUkBa1zMKv4

Particules Filter

- Mathematic explanation
  - let X_t the set of particules $p_t{}^{[m]}$ such as

$$X_t := x_t{}^{[0]}, x_t{}^{[1]}, \ldots, x_t{}^{[M]}$$



Where $1 \leq m \leq M$

  - Inspired by Baye filter

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

Particule m depends of a previous particule at $t-1$

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

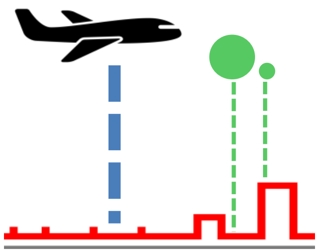
Particule m get an importance factor according observation

$$w_t{}^{[m]} = p(z_t | x_t{}^{[m]})$$

Particules Filter

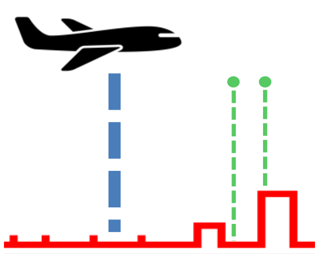
- Mathematic explanation
  - the particule set can be redefined as followed:

Importance Weight



$$X_t := \left\{\left\langle x_t^{[m]}, w_t^{[m]} \right\rangle\right\}_{m=1,\dots,M}$$

State Hypothesis



Particules Filter

- Algorithm

Algorithm Particle_filter$(\mathcal{X}_{t-1}, u_t, z_t)$**:**

$\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$

for $m = 1$ to M do

$\boxed{\text{sample } x_t^{[m]} \sim p(x_t \mid u_t, x_{t-1}^{[m]})}$ How to get samples ?

$\boxed{w_t^{[m]} = p(z_t \mid x_t^{[m]})}$ How to weight ?

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

endfor

for $m = 1$ to M do

$\boxed{\text{draw } i \text{ with probability} \propto w_t^{[i]}}$ How to get new particules ?

add $x_t^{[i]}$ to $\mathcal{X}_t$

endfor

return $\mathcal{X}_t$

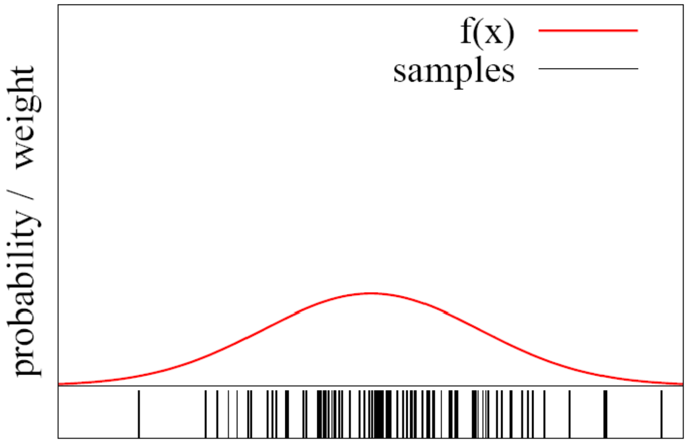
Probabilistic Robotics, Sebastian THRUN,
Wolfram BURGARD, Dieter FOX, 2000

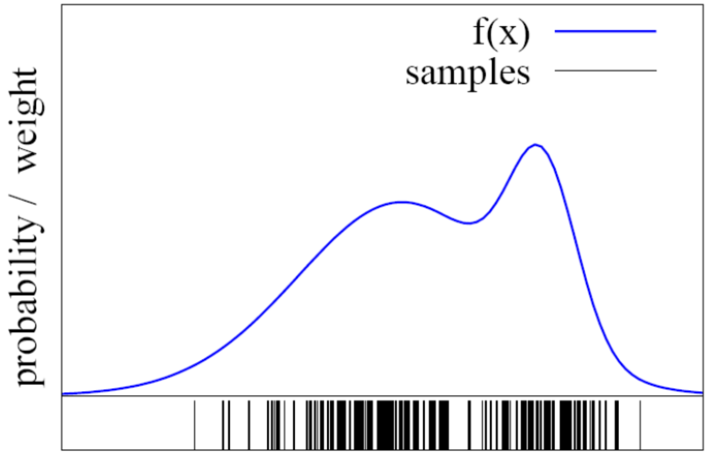
CPE LYON
ÉCOLE SUPÉRIEURE DE CHIMIE PHYSIQUE ÉLECTRONIQUE DE LYON

Particules Filter

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

- How to get samples ?





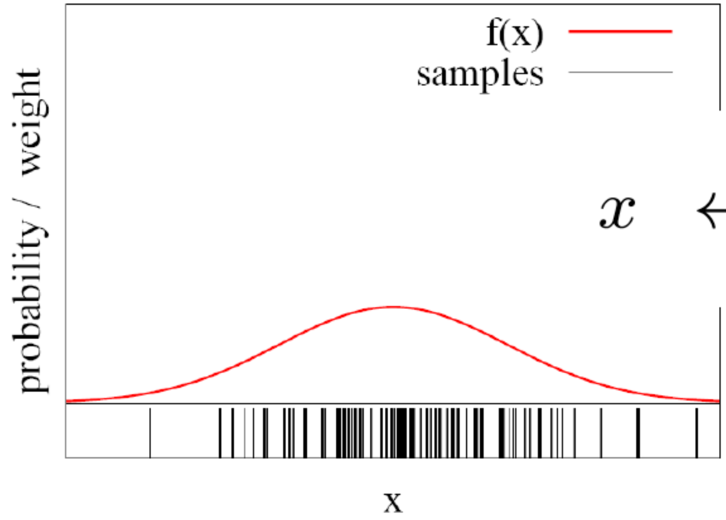
The **more particles fall into a region**, the **higher** the **probability** of the region

How to obtain such samples?

Particules Filter

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

- How to get samples ?
  - Closed Form Sampling is Only Possible for a Few Distributions
  - E.g Gaussian



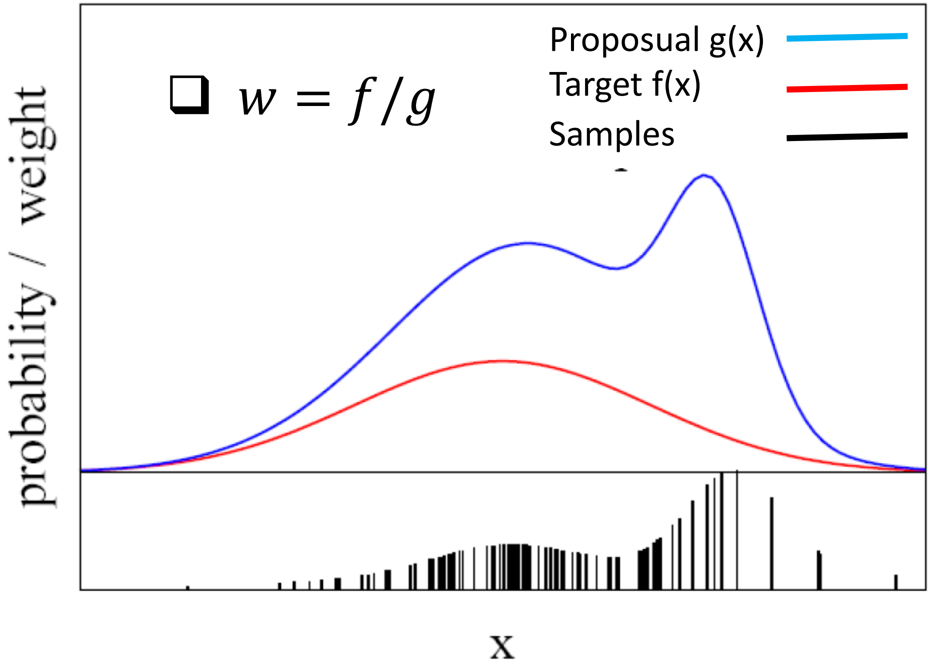
$$x \leftarrow \frac{1}{2} \sum_{i=1}^{12} \text{rand}(-\sigma, \sigma)$$

How to sample from **other** distributions?

Particules Filter

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

- How to get samples ?
  - Use a different distribution $\boldsymbol{g}$ to generate samples from $\boldsymbol{f}$
  - According for the " differences between $\boldsymbol{g}$ and $\boldsymbol{f}$" using a weight $\boldsymbol{w} = \boldsymbol{f}/\boldsymbol{g}$



Particules Filter

- How to Weight ?
- According for the “ differences between g and f” using a weight $w = f/g$

$$w_t^{[m]} = \frac{target\ f\left(x_t^{[m]}\right)}{proposual\ g(x_t^{[m]})}$$

$$\sim p(z_t|x_t^{[m]})$$

Cf Probabilistic Robotics, Sebastian THRUN, Wolfram BURGARD, Dieter FOX, 2000, Chapter 4

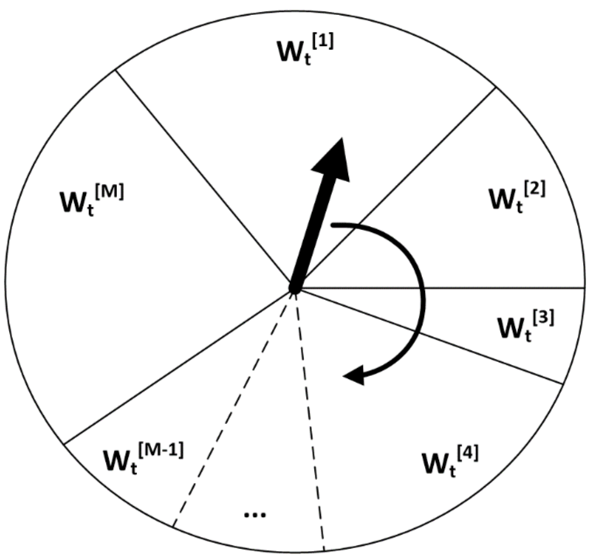
Particules Filter

- How to get new particules ?

$$X_t := \left\{ \left\langle x_t{}^{[m]}, w_t{}^{[m]} \right\rangle \right\}_{m=1,\ldots,M}$$

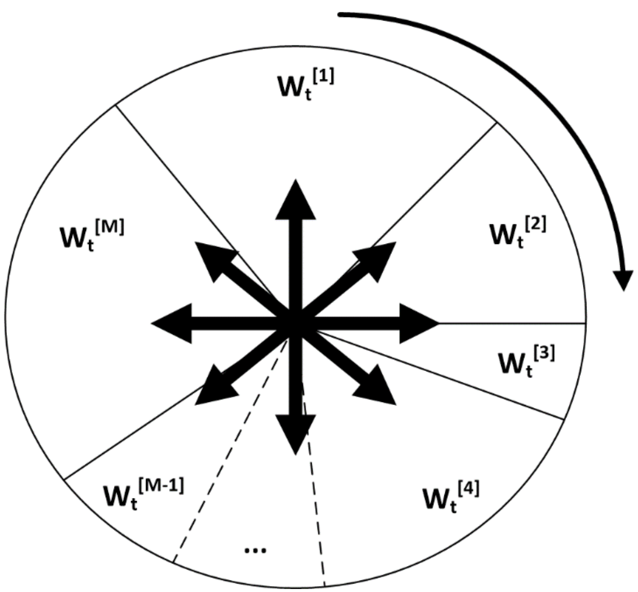
The **more particles fall into a region**, the **higher** the **probability** of the region

Roulette wheel



O (M Log M)

Stochastic universal sampling



O (M)

Particules Filter

- Example

Algorithm Particle_filter($\mathcal{X}_{t-1}, u_t, z_t$):

$\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$

for $m = 1$ to M do

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

  $w_t^{[m]} = p(z_t \mid x_t^{[m]})$

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

endfor

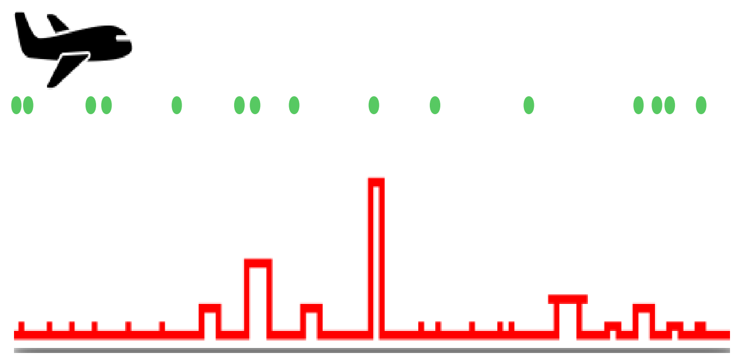
for $m = 1$ to M do

  draw i with probability $\propto w_t^{[i]}$

  add $x_t^{[i]}$ to $\mathcal{X}_t$

endfor

return $\mathcal{X}_t$



Probabilistic Robotics, Sebastian THRUN, Wolfram BURGARD, Dieter FOX, 2000

Particules Filter

- Example

How to get samples ? $\boldsymbol{x_t}^{[\boldsymbol{m}]} \sim \boldsymbol{p}(\boldsymbol{x_t}|\boldsymbol{u_t}, \boldsymbol{x_{t-1}}^{[\boldsymbol{m}]})$

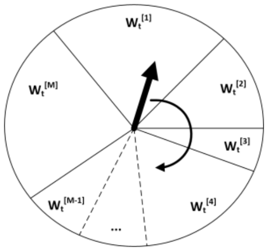
$$u_t \sim Rand\ Uniform\ [-1, 10] = \Delta x_t$$

$$x_t^{[m]} = x_{t-1}^{[m]} + \Delta x_t$$

How to weight ? $\boldsymbol{w_t}^{[\boldsymbol{m}]} \sim \boldsymbol{p}(\boldsymbol{z_t}|\boldsymbol{x_t}^{[\boldsymbol{m}]})$

$$w_t^{[m]} = \begin{cases} 1\ if\ z_t = esimate(\boldsymbol{z_t}^{[\boldsymbol{m}]}) \\ 0 \end{cases}$$

How to get new particules ?



Particules Filter

- Example

...

for $m = 1$ to M do

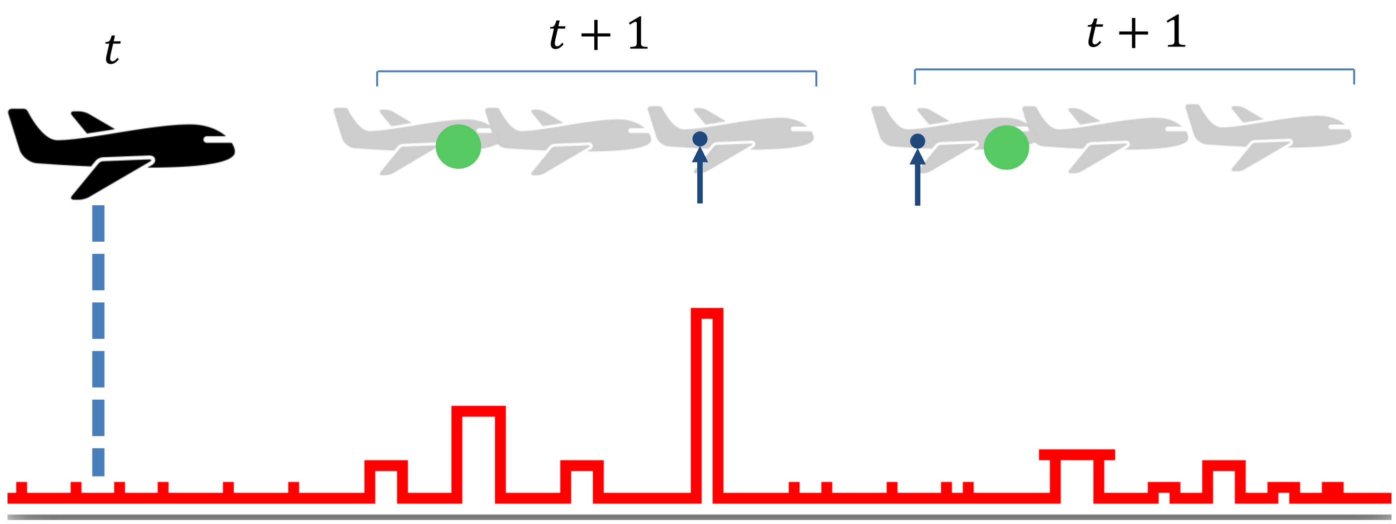
sample $x_t^{[m]} \sim p(x_t \mid u_t, x_{t-1}^{[m]})$

$w_t^{[m]} = p(z_t \mid x_t^{[m]})$

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

endfor

...



$$u_t \sim Rand\ Uniform\ [-1, 10] = \Delta x_t$$

Particules Filter

- ❑ Example

...

for $m = 1$ to M do

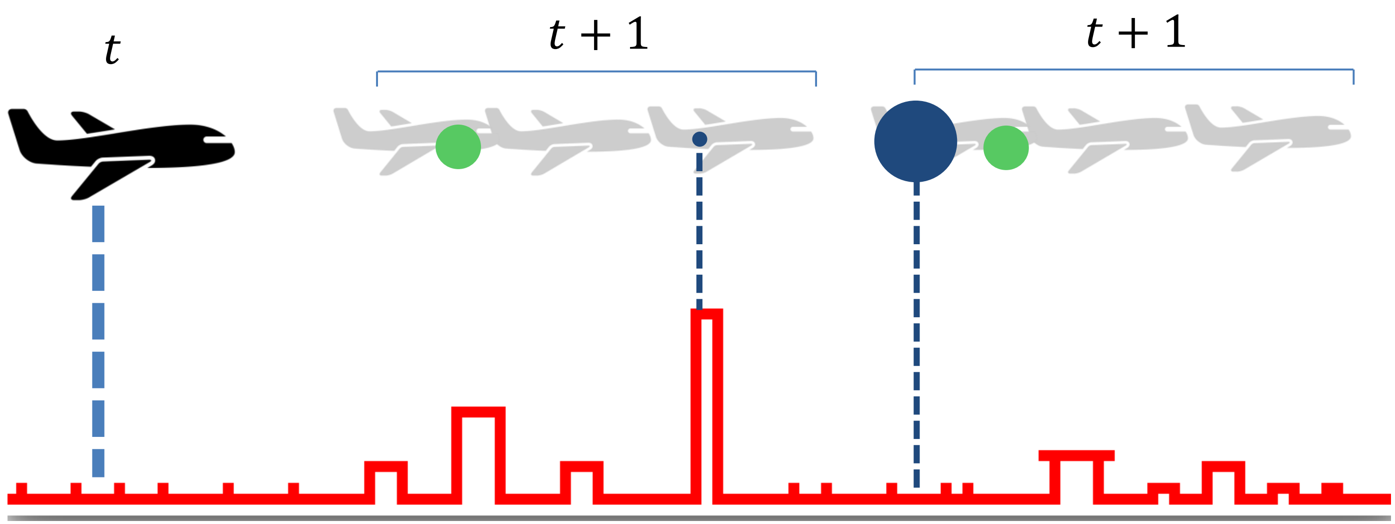
  sample $x_t^{[m]} \sim p(x_t \mid u_t, x_{t-1}^{[m]})$

  $w_t^{[m]} = p(z_t \mid x_t^{[m]})$

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

endfor

...



$$w_t^{[m]} = \begin{cases} 1 \; if \; z_t = esimate(\boldsymbol{z_t}^{[m]}) \\ 0 \end{cases}$$

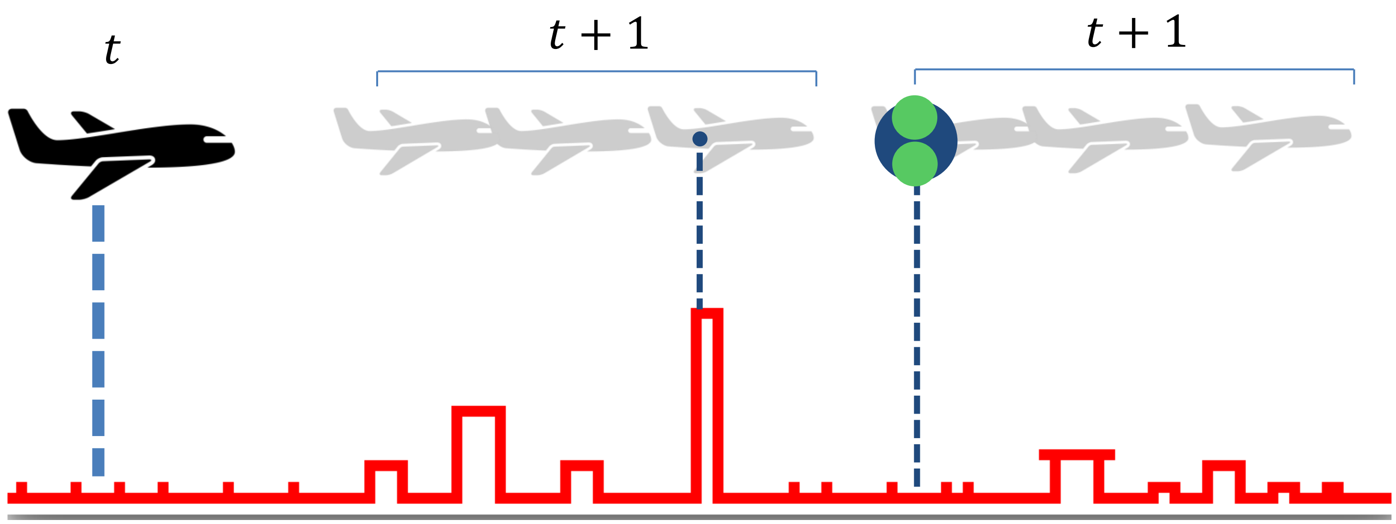
Particules Filter

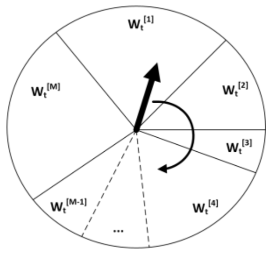
- ❑ Example

...

for $m = 1$ to M do
  draw i with probability $\propto w_t^{[i]}$
  add $x_t^{[i]}$ to $\mathcal{X}_t$
endfor

...

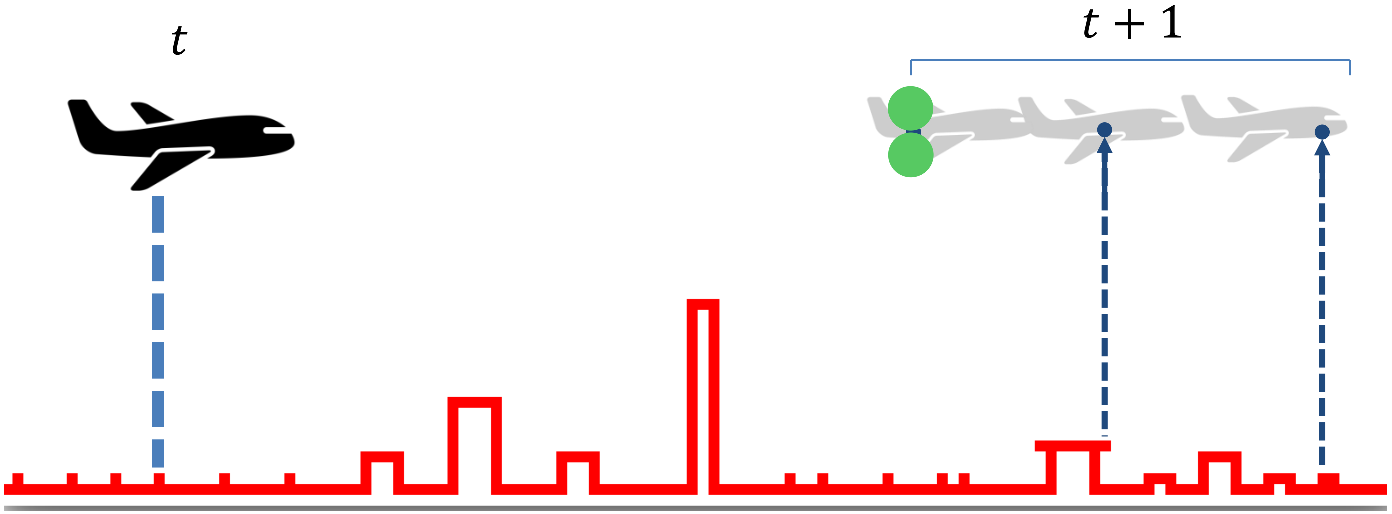


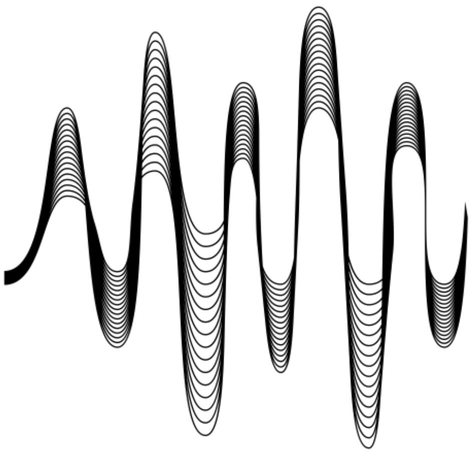


Particules Filter

- ❑ Example

$$
\begin{aligned}
&\ldots\\
&\text{for } m = 1 \text{ to } M \text{ do}\\
&\qquad \text{sample } x_t^{[m]} \sim p(x_t \mid u_t, x_{t-1}^{[m]})\\
&\qquad w_t^{[m]} = p(z_t \mid x_t^{[m]})\\
&\qquad \bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle\\
&\text{endfor}\\
&\ldots
\end{aligned}
$$





Motion and Sensors data Modeling

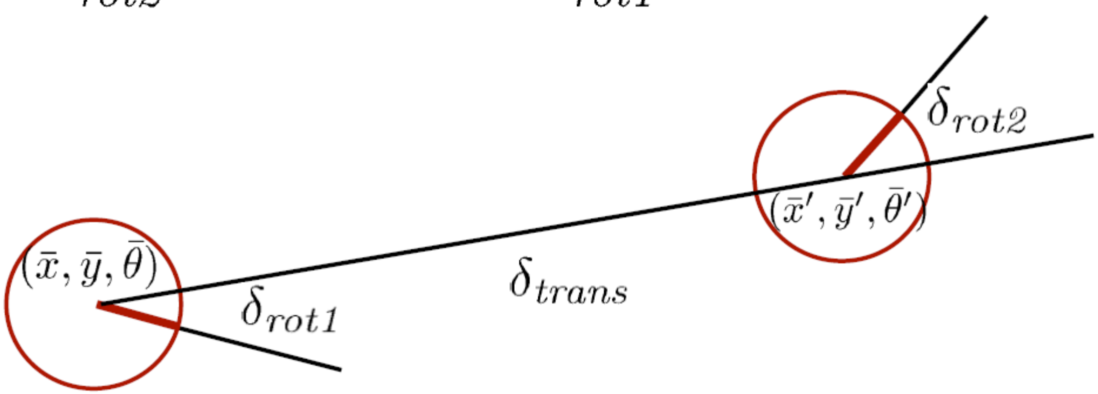
Motion Model

- Odometry model (e.g vehicule)

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

- Robot moves from $(\bar{x}, \bar{y}, \bar{\theta})$ to $(\bar{x}', \bar{y}', \bar{\theta}')$
- Odometry information $u = (\delta_{rot1}, \delta_{trans}, \delta_{rot2})$

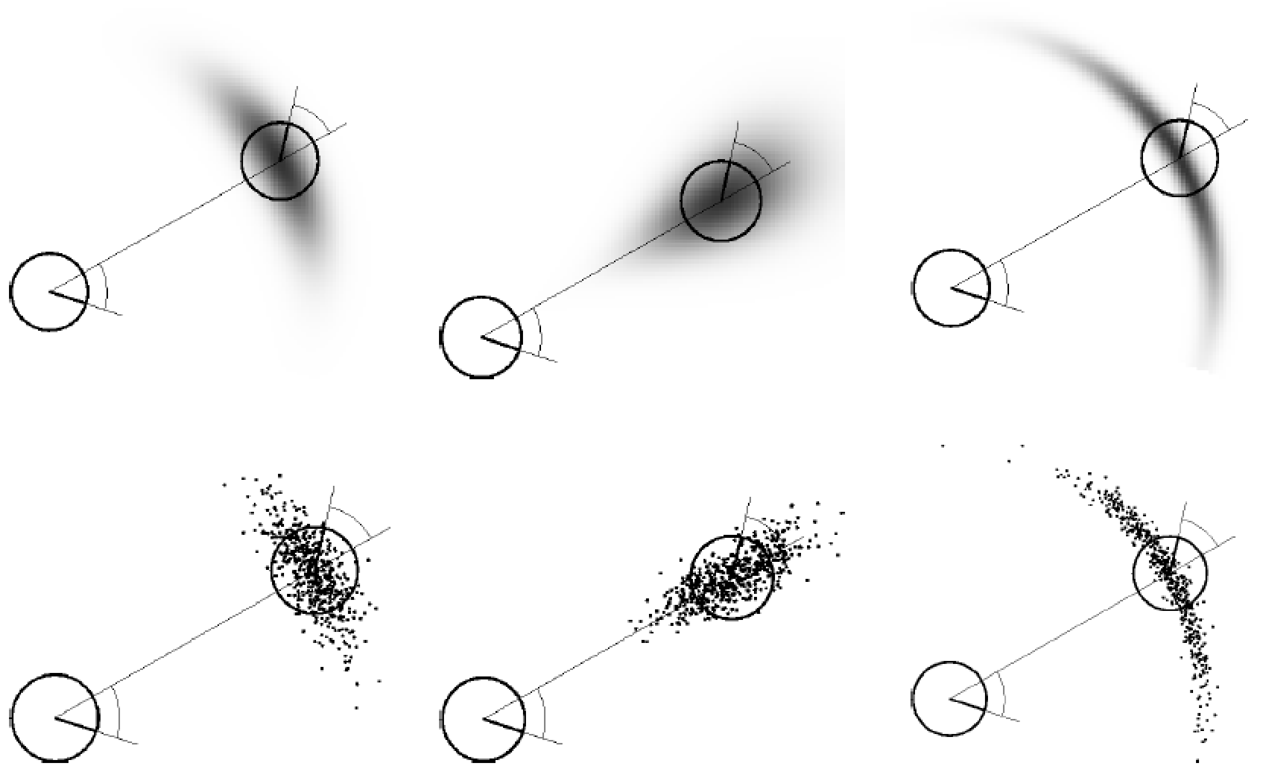
$$\begin{aligned}
\delta_{trans} &= \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2} \\
\delta_{rot1} &= \operatorname{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta} \\
\delta_{rot2} &= \bar{\theta}' - \bar{\theta} - \delta_{rot1}
\end{aligned}$$



Motion Model

- Odometry model (e.g vehicule)

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg



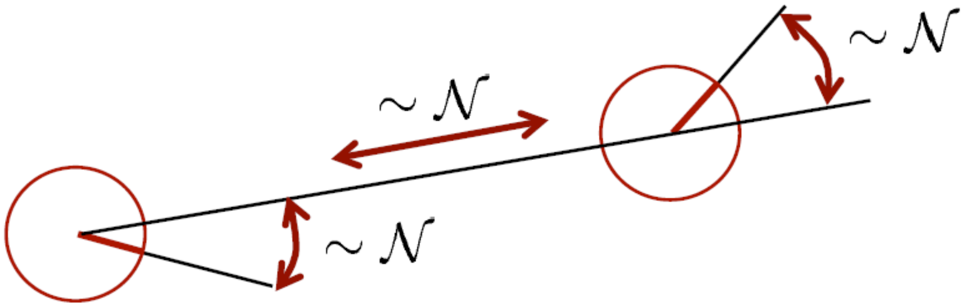
Motion Model

- Odometry model (e.g vehicule)

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

- Noise in odometry $u = (\delta_{rot1}, \delta_{trans}, \delta_{rot2})$
- Example: Gaussian noise

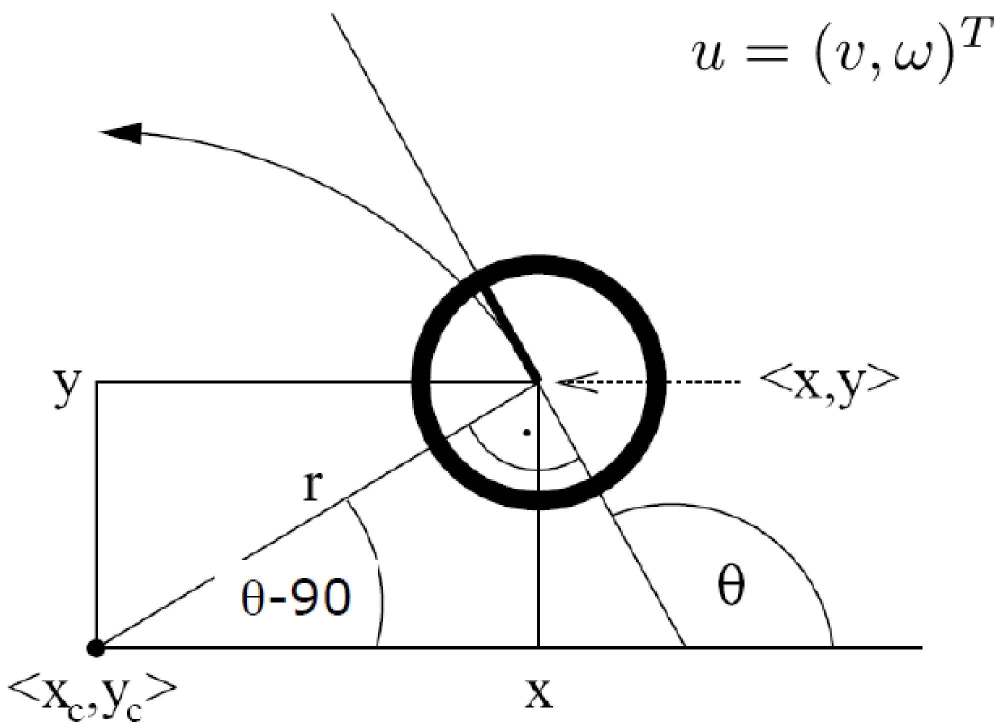
$$u \sim \mathcal{N}(0, \Sigma)$$



Motion Model

- Velocity model (e.g vehicule)

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg



Motion Model

- Velocity model (e.g vehicule)

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

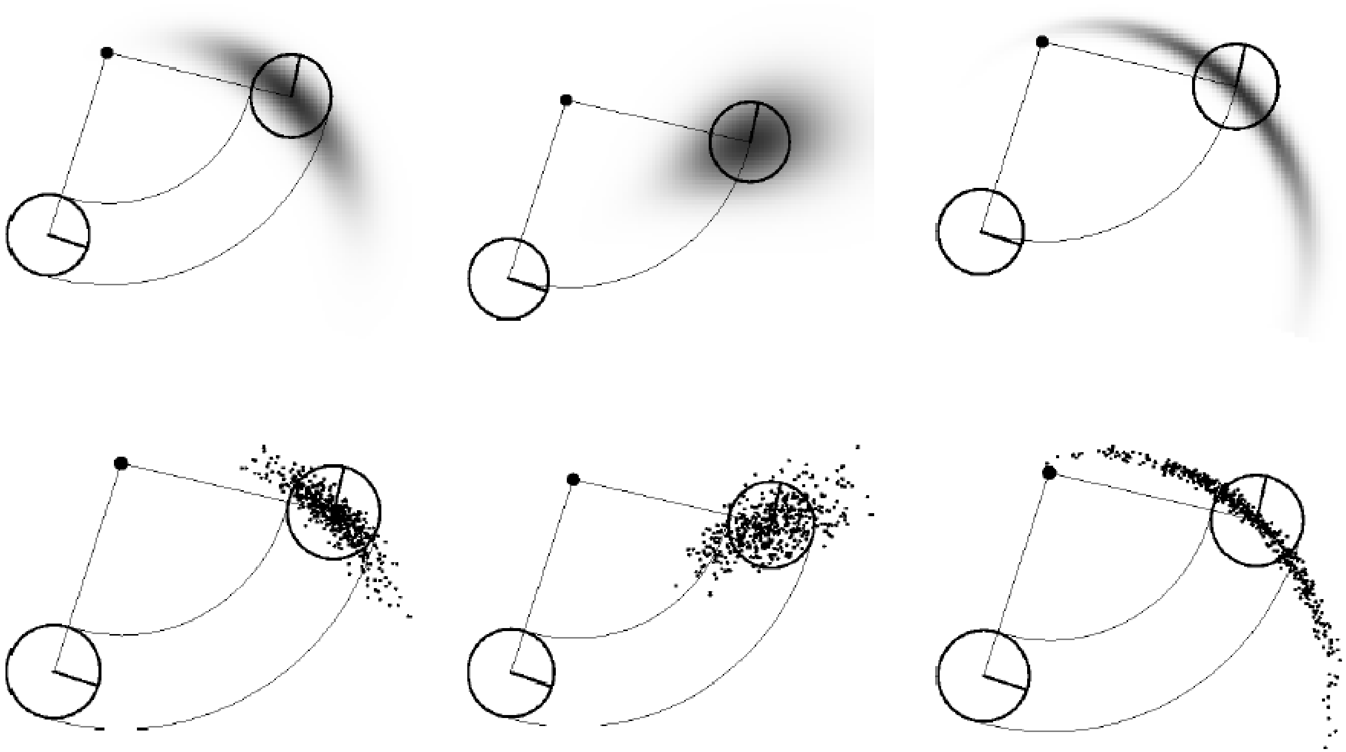
$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v}{\omega}\sin\theta + \frac{v}{\omega}\sin(\theta + \omega\Delta t) \\ \frac{v}{\omega}\cos\theta - \frac{v}{\omega}\cos(\theta + \omega\Delta t) \\ \omega\Delta t + \gamma\Delta t \end{pmatrix}$$

Term to account for the final rotation

Motion Model

- Velocity model (e.g vehicule)

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg



Sensor Model

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

Model for Laser Scanners

- Scan *z* consists of *K* measurements.

$$z_t = \{z_t^1, \ldots, z_t^k\}$$

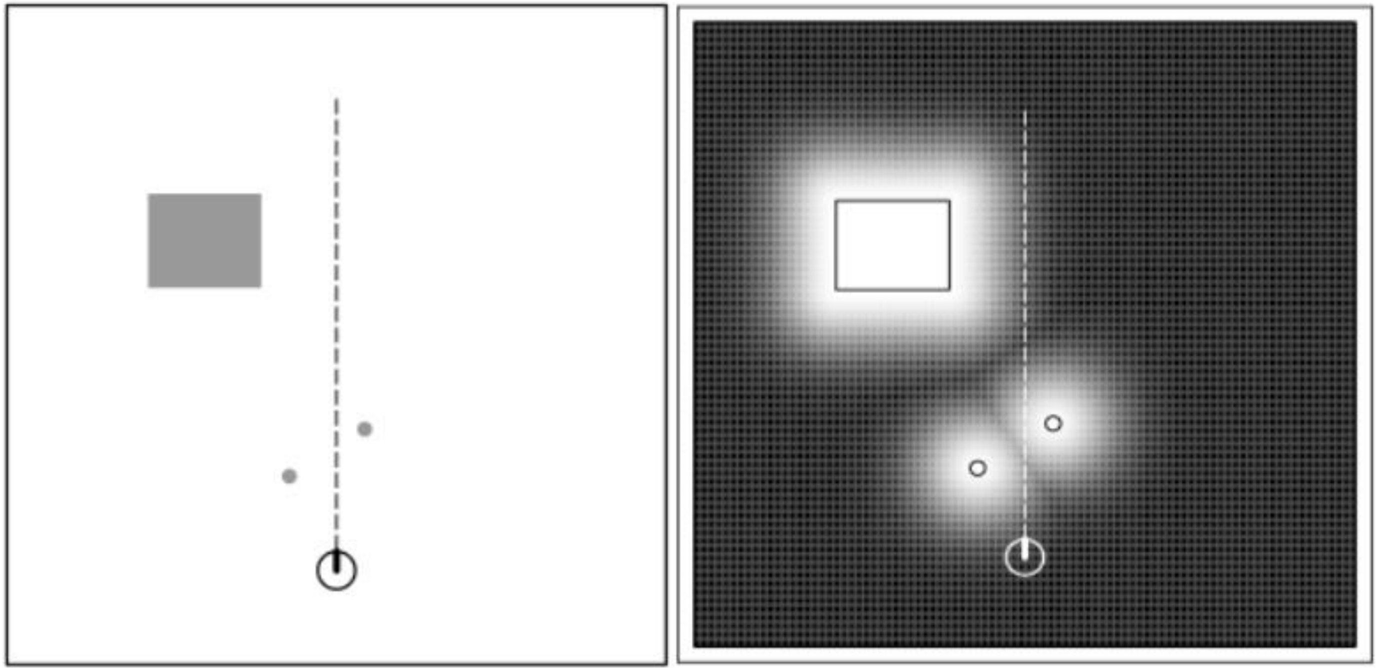
- Individual measurements are independent given the robot position

$$p(z_t \mid x_t, m) = \prod_{i=1}^{k} p(z_t^i \mid x_t, m)$$

Sensor Model

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

Beam-Endpoint Model

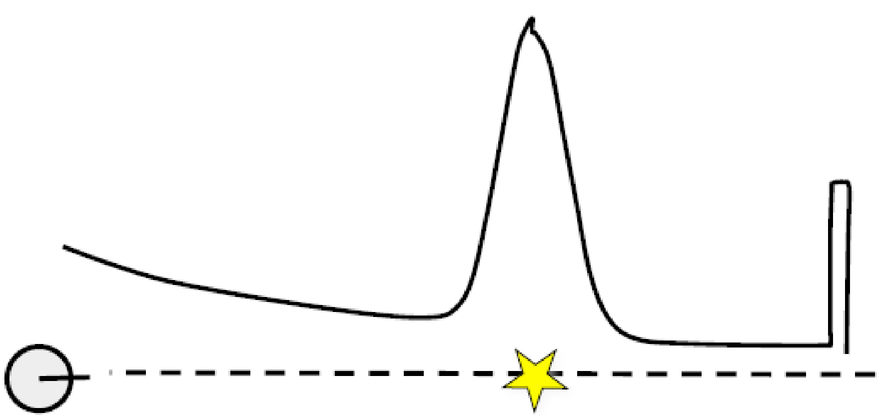


Sensor Model

Slid from :Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

Ray-cast Model

- Ray-cast model considers the first obstacle long the line of sight
- Mixture of four models



References

References (1/2)

- Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg
- Probabilistic Robotics, Sebastian THRUN, Wolfram BURGARD, Dieter FOX, 2000
- The particule Filter explained without equations, Andreas Svensson, UPPSALA UNIVERSITET

- Lectures video :
  - https://www.youtube.com/watch?v=5Pu558YtjYM
  - https://www.youtube.com/watch?v=aUkBa1zMKv4

Jacques Saraydaryan

Jacques.saraydaryan@cpe.fr