Particles Filter: a robot localization application

OutLine

- □ Introduction
- □ Basics
- □ Motion and Sensors Model

Inspired by :

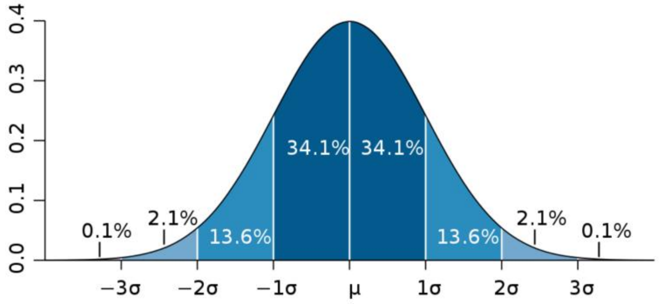
Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg
Probabilistic Robotics, Sebastian THRUN, Wolfram BURGARD, Dieter FOX, 2000

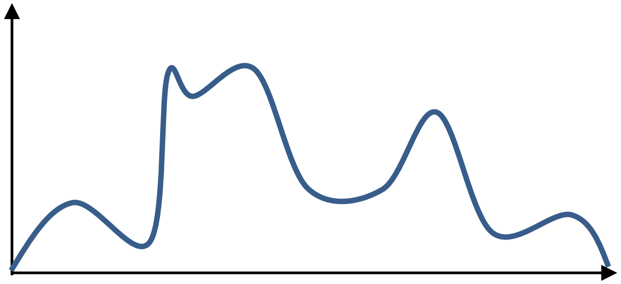
Particles filter, What is the need?

Particles Filter: What is the need?

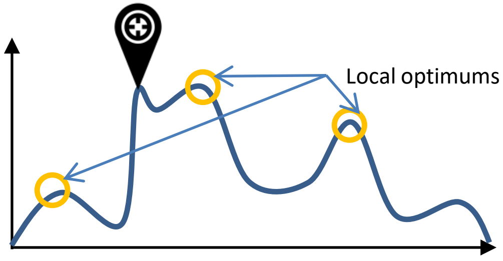
Needs

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





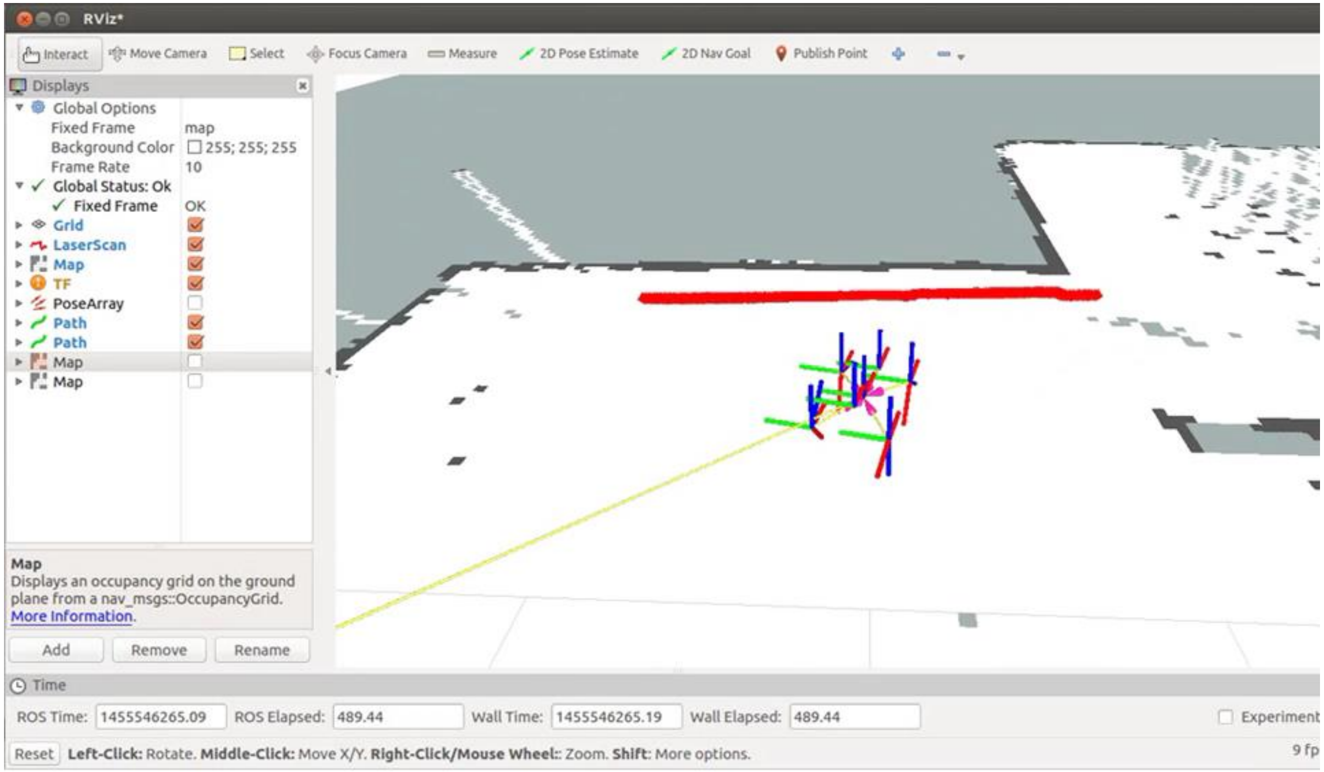
- ❑ Application:
  - ❑ Robot localization
  - ❑ Function optimum finding



Copyright © Jacques Saraydaryan

Particles Filter: What is the need?

Needs



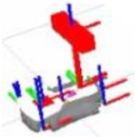
Global Options
Fixed Frame map
Background Color □ 255; 255; 255
Frame Rate 10
✓ Global Status: Ok
✓ Fixed Frame OK
Grid ☑
LaserScan ☑
Map ☑
TF ☑
PoseArray □
Path ☑
Path ☑
Map □
Map □

Map
Displays an occupancy grid on the ground plane from a nav_msgs::OccupancyGrid. More Information.

ROS Time: 1455546265.09 ROS Elapsed: 489.44 Wall Time: 1455546265.19 Wall Elapsed: 489.44 □ Experiment

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options.

9 fp



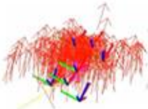
Robot Model

Laser trace

Occupancy Map

Global Cost map

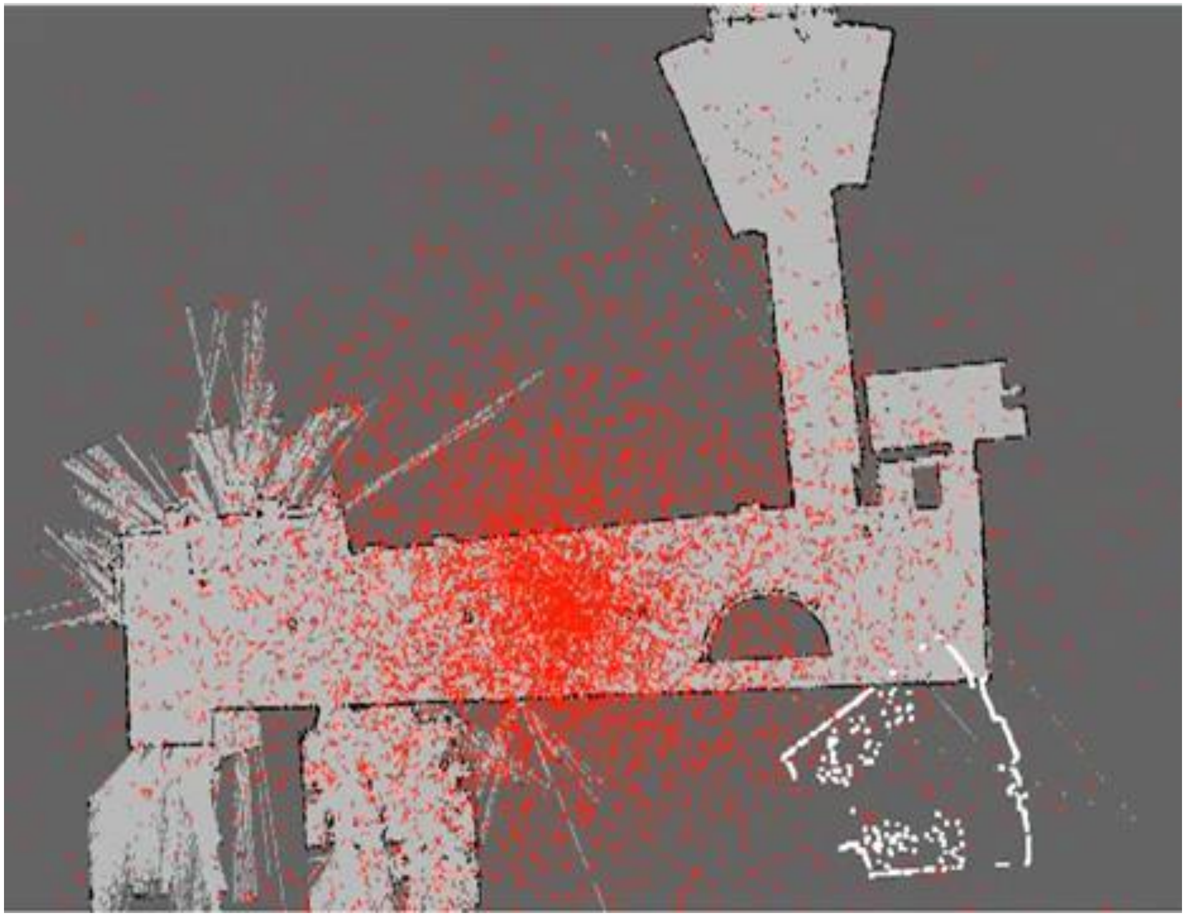
Local Cost map



Cloud Particles : Probable robot location

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

Needs



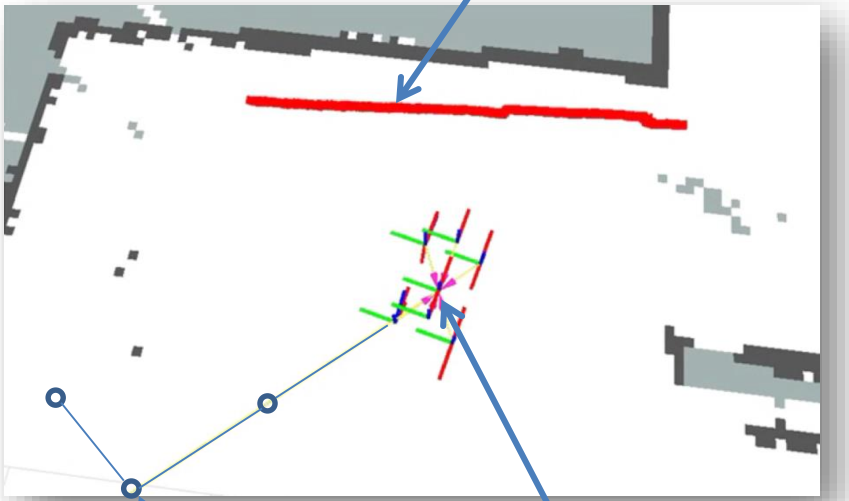
https://www.youtube.com/watch?v=OVoa11xd3vE

Particles Filter: What is the need?

Needs

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

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



observations $\boldsymbol{z}$
(laser beam)

state $\boldsymbol{x}$
(localization)

controls $\boldsymbol{u}$
(odometry)

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

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

$$bel(x_t) = \frac{p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t \mid z_{1:t-1}, u_{1:t})}{p(z_t | \, z_{1:t-1}, u_{1:t})} \quad \xrightarrow{\text{Bayes Law}} \quad p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

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

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

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

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 ↓

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

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

Recursive Bayes Filter

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

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

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

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

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

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

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

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

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

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

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

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

Recursive Bayes Filter

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

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

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

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

Recursive Bayes Filter

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

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

OBSERVATION

→ Correction step

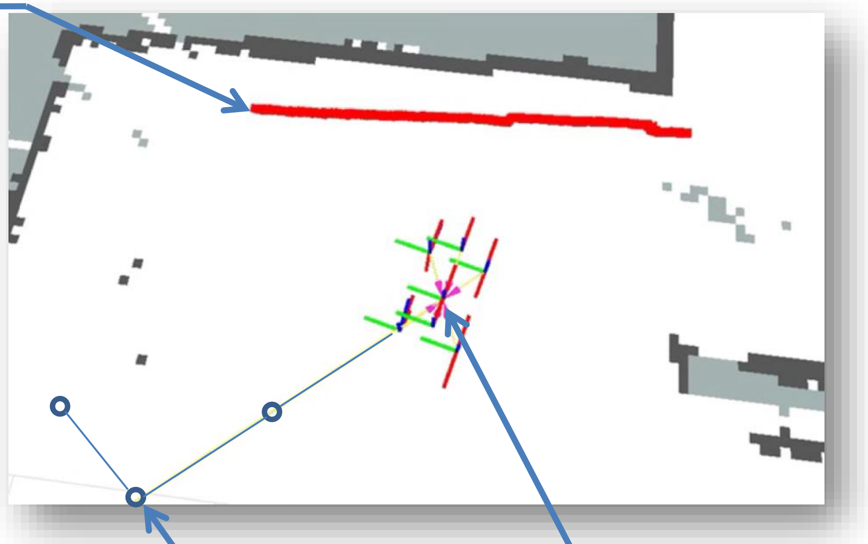
MOTION / CONTROL

→ 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 $\boldsymbol{z}$

(laser beam)

controls $\boldsymbol{u}$

(odometry)

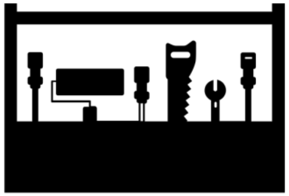
state $\boldsymbol{x}$

(localization)

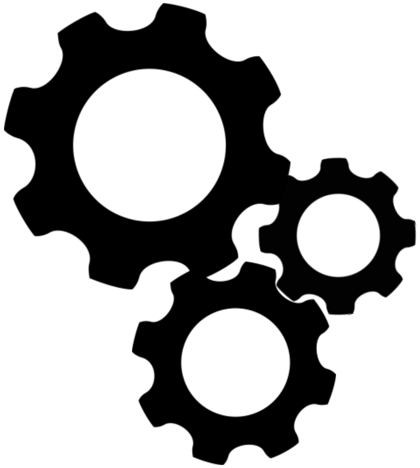
Bayes Filter

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

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

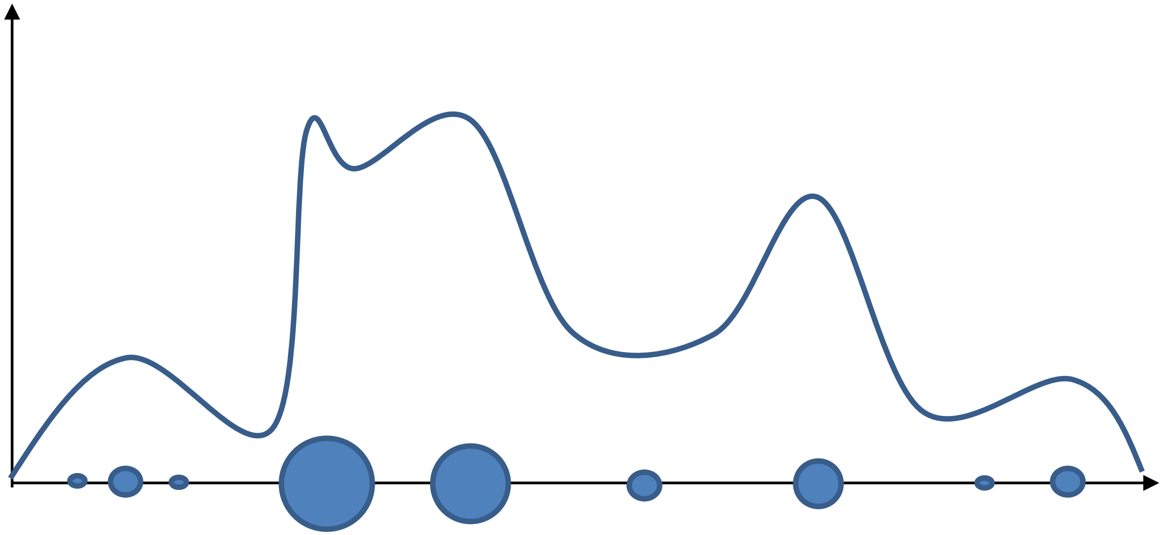


(odometry)



Particles Filter : Basics

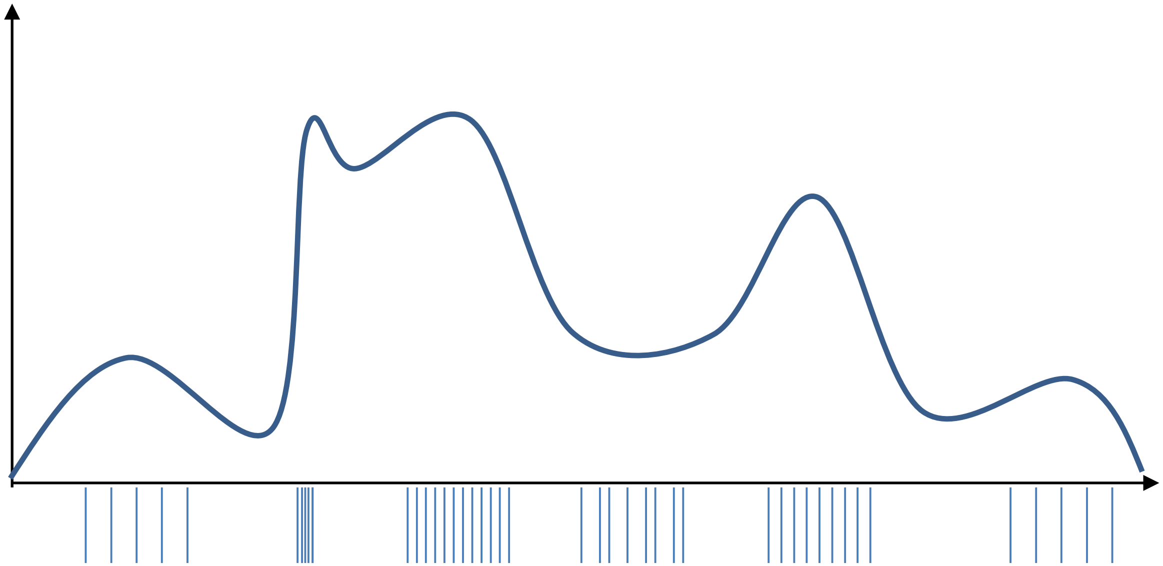
Principles



1

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

Principles



2

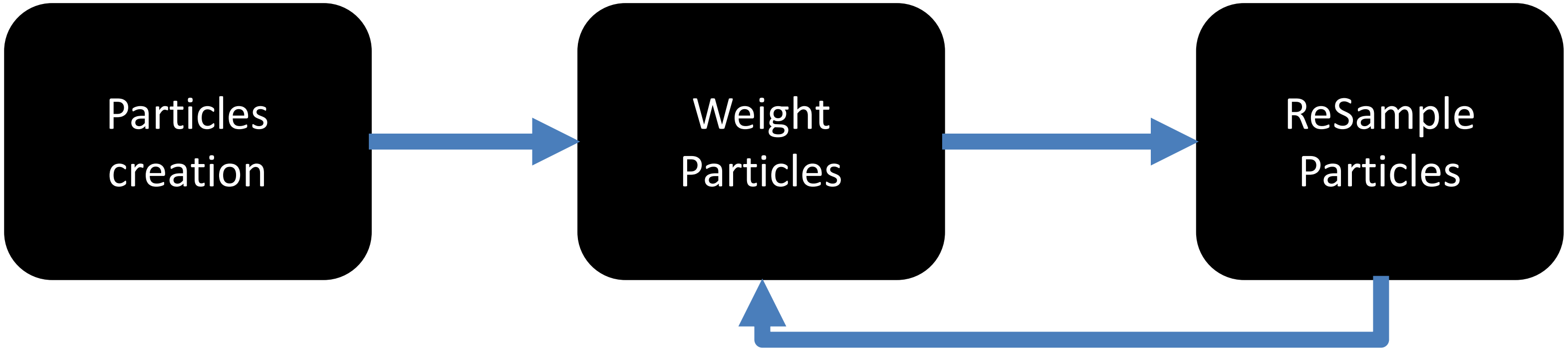
The **heigher** the sample **weight** is, the more particules are **distributed around**

Principles

The resampling step is a probabilistic ***implementation of the Darwinian idea of survival of the fittest****: It refocuses the particle set to regions in state space with high posterior probability.*

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

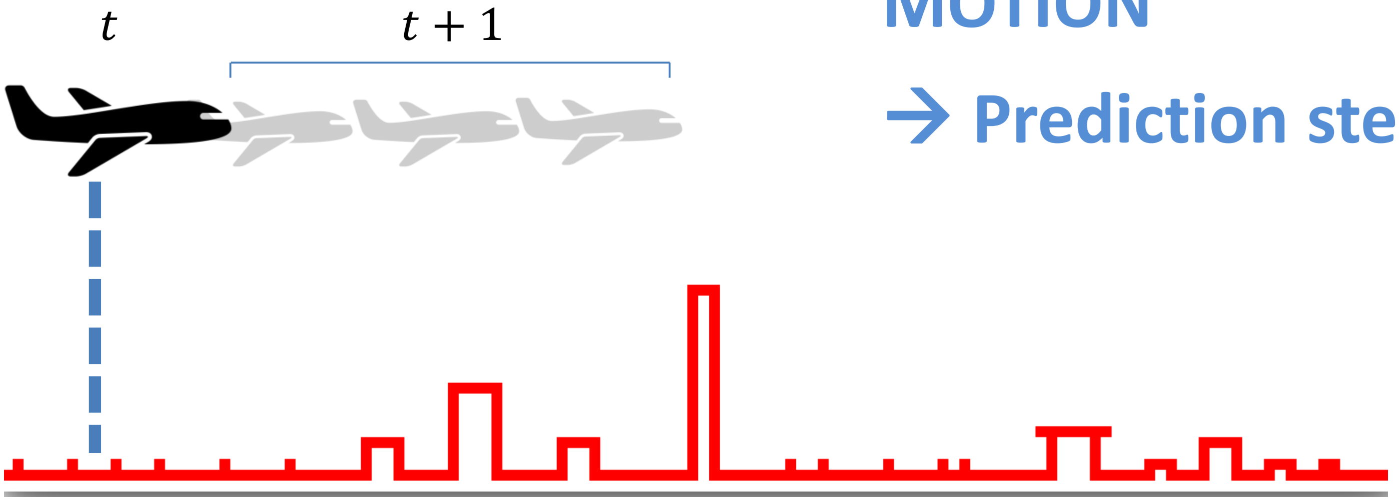
Principles



How to obtain such samples ?

Example

Particles Filter : Basics



MOTION

→ Prediction step

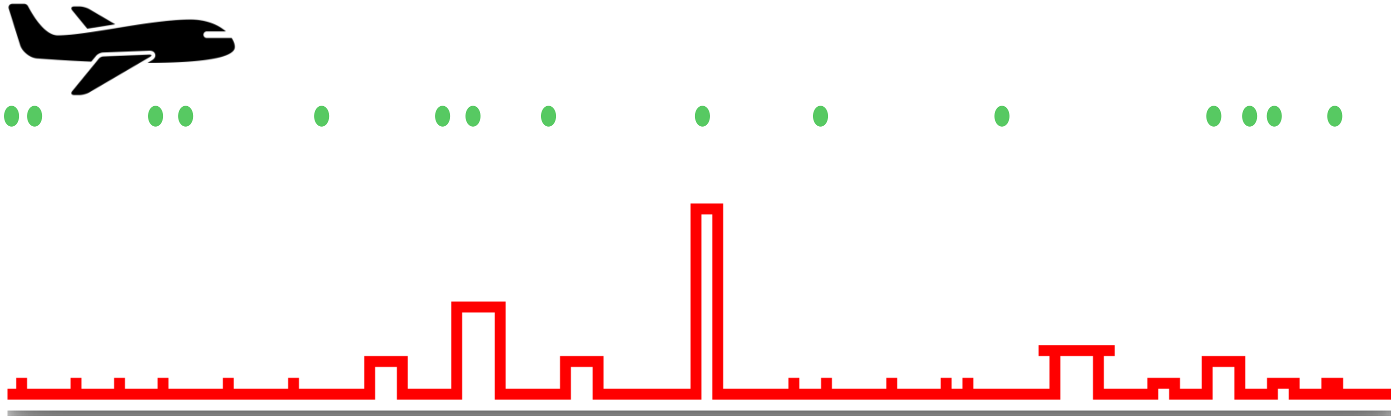
OBSERVATION

→ Correction step

Particles Filter : Basics

1

Particles Creation

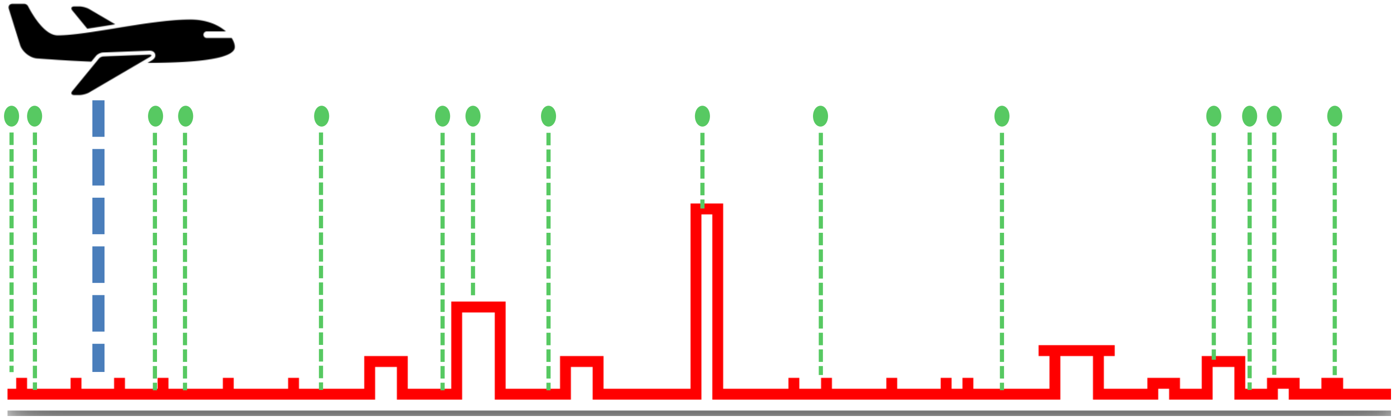


If no initial knowledge, uniform distribution of the particles accross the environment

Particles Filter : Basics

2

Weight Particles



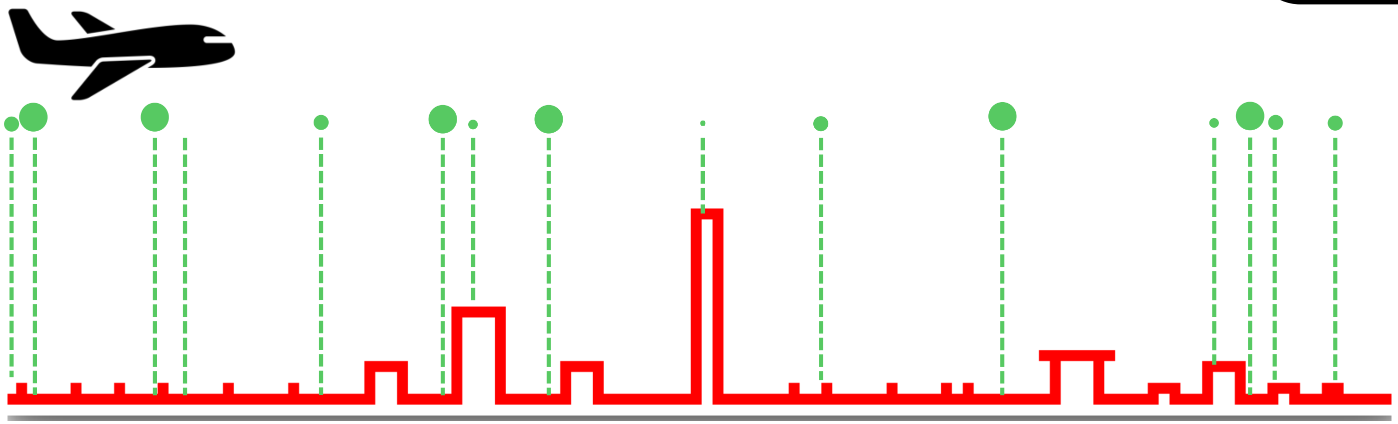
Weight particles according the current **observation** and the **estimate observation** at the particle position

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

Particles Filter : Basics

2

Weight Particles



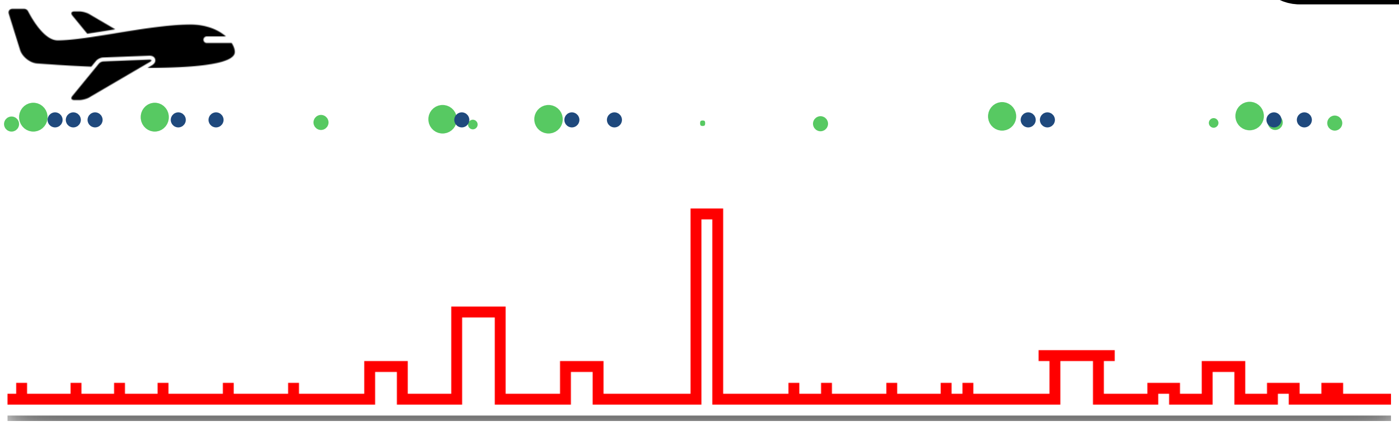
Weight particles according the current **observation** and the **estimate observation** at the particle position

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

Particles Filter : Basics

3

Resample Particles



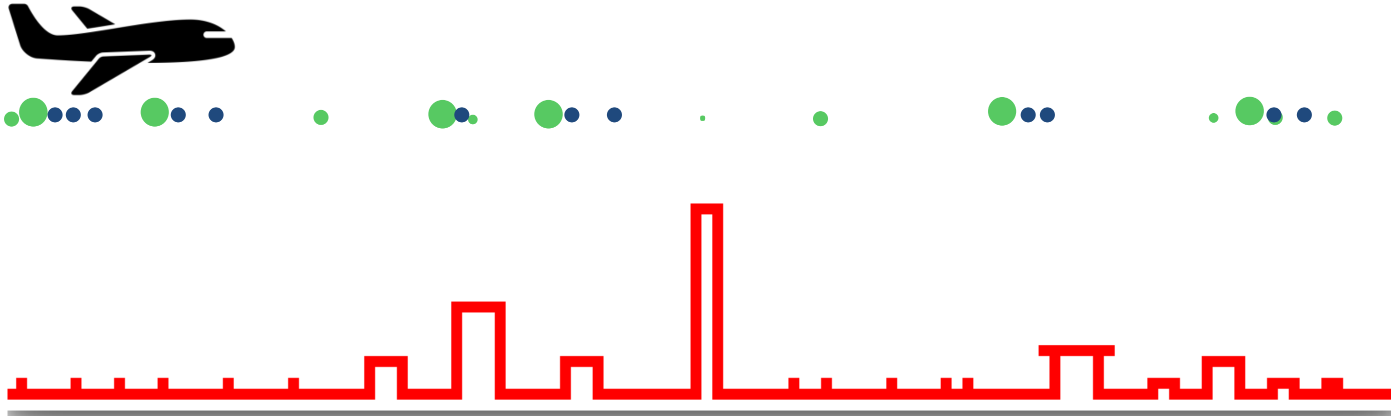
Resample Particles **according** their **weight.**

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

Particles Filter : Basics

3

Resample Particles

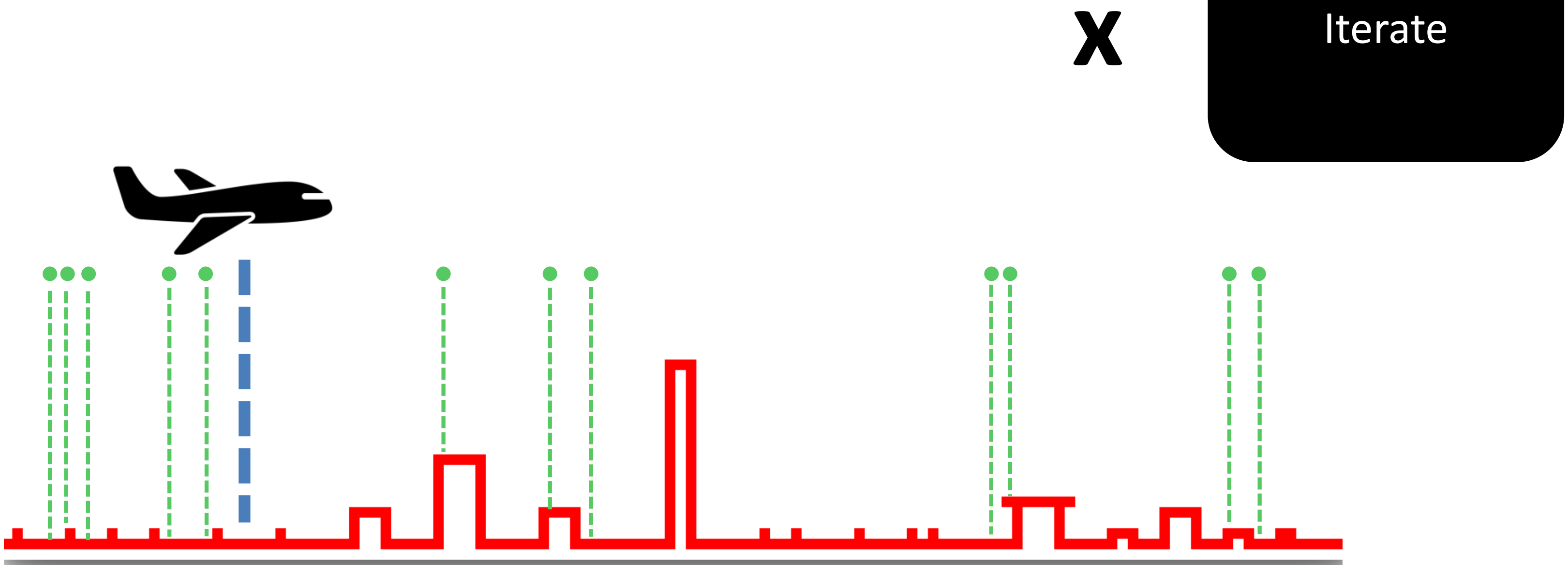


Resample Particles **according** their **weight.**

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

Remove old particles

Particles Filter : Basics

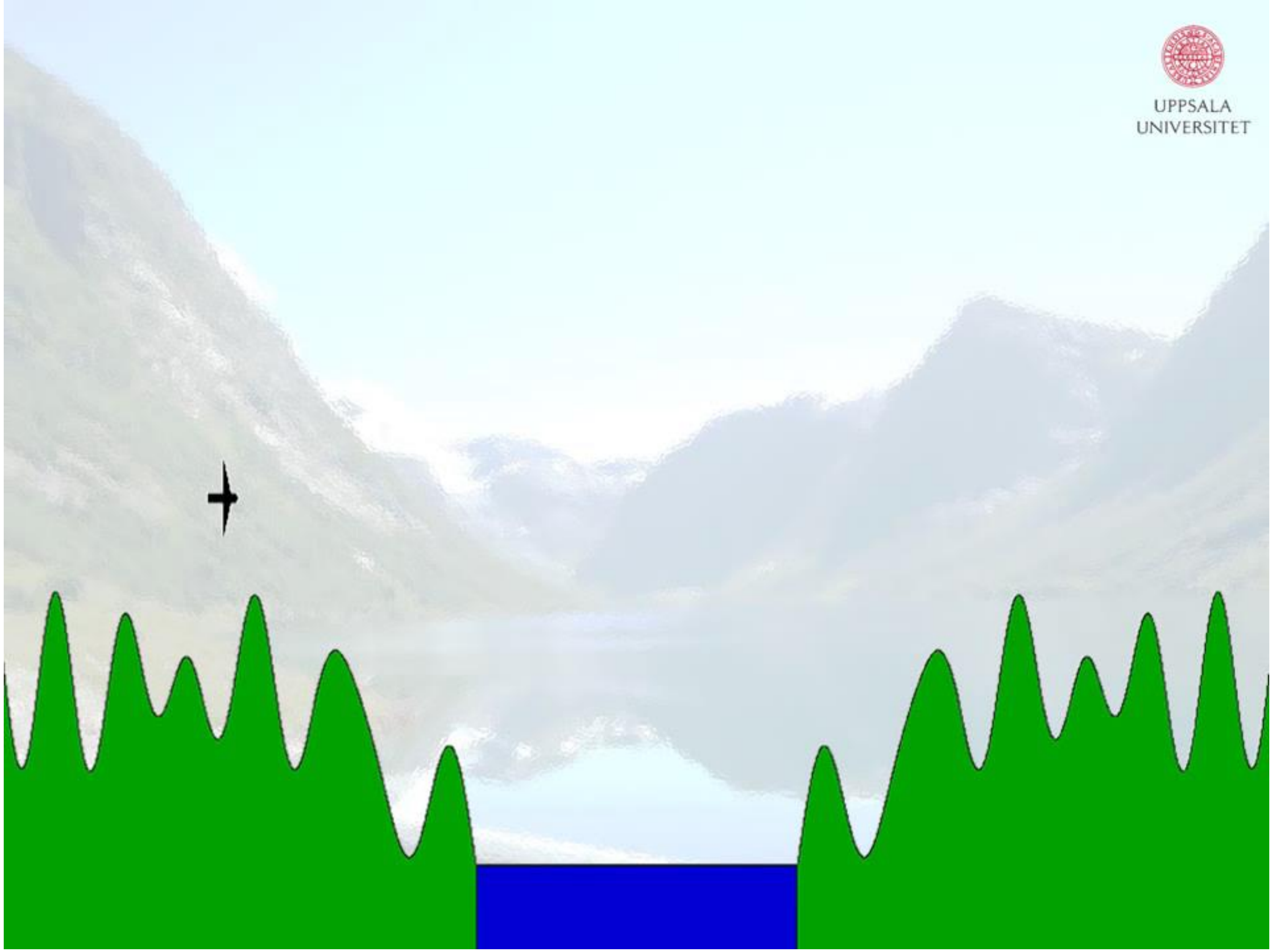


Make Observation at t+1 (plane move)

Weight Particles

Remove old Particles

Example



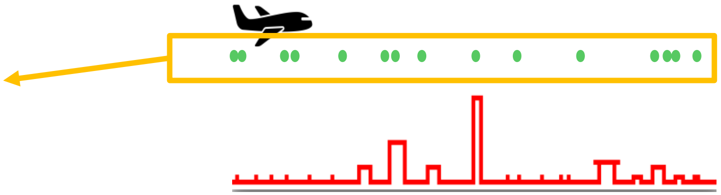
https://www.youtube.com/watch?v=aUkBa1zMKv4

Particles 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 $t-1$

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

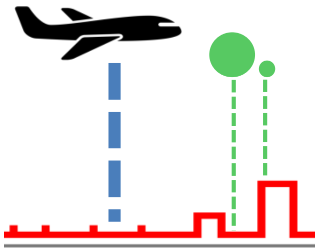
Particule m get an importance factor according observation

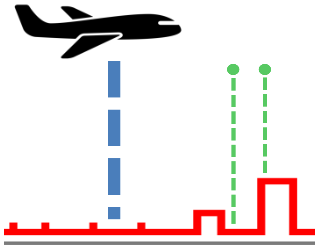
$$w_t{}^{[m]} = p(z_t | x_t{}^{[m]})$$

Particles Filter

- ❑ Mathematic explanation
  - ❑ the particle set can be redefined as followed:

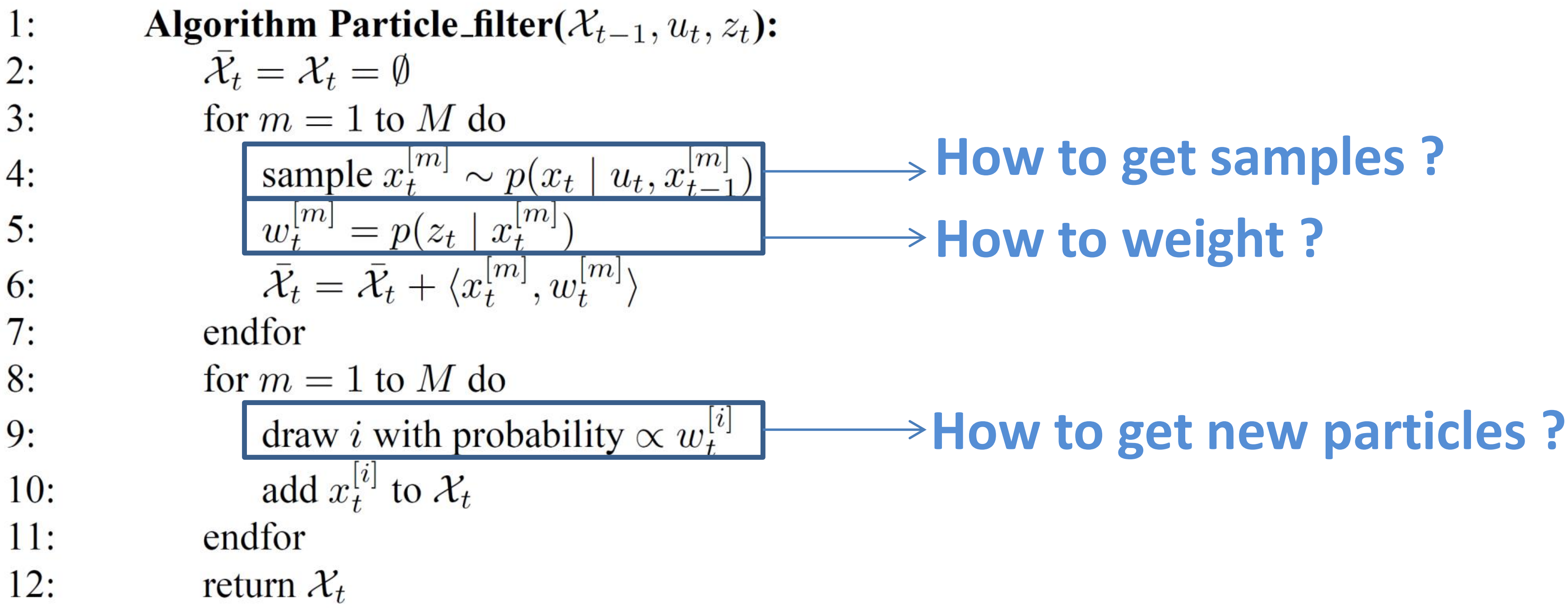
$$X_t := \left\{ \left\langle \underbrace{x_t^{[m]}}_{\text{State Hypothesis}}, \overbrace{w_t^{[m]}}^{\text{Importance Weight}} \right\rangle \right\}_{m=1,\ldots,M}$$





Particles 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
        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 $\mathcal{X}_t$
```

How to get samples ?

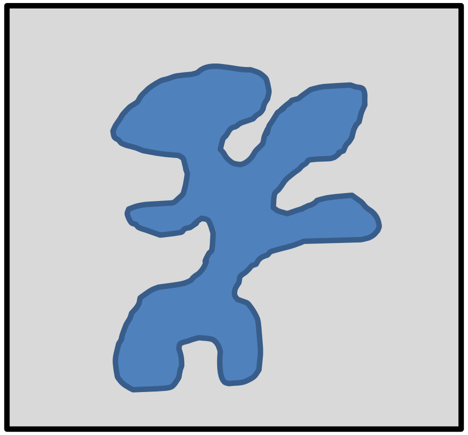
How to weight ?

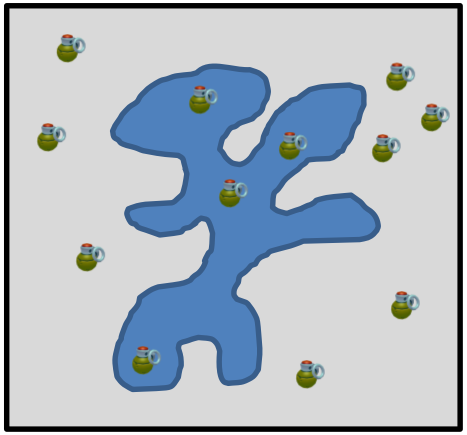
How to get new particles ?

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

Particles Filter

- ❑ How to get new particles ?
  - ❑ Particles Filter also called **Sequential Monte Carlo**
  - ❑ What are Monte Carlo Method ?
    - ❑ Basic (historical) Example : How to compute a lake area ?





S *total area*
***S**lake* *lake area*
N *number of round shot*
X *number of Round Shot on the ground*

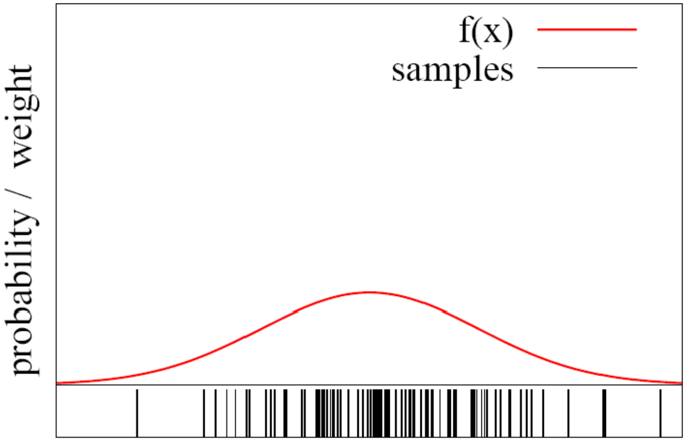
$$\frac{S}{S_{lake}} = \frac{N}{N-X}$$

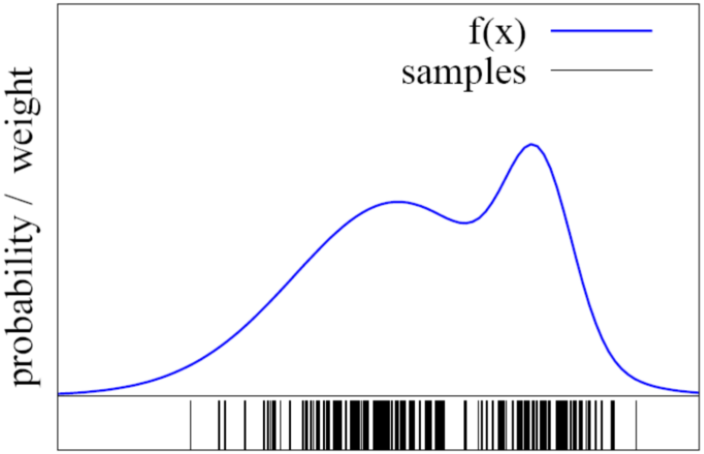
$$S_{lake} = \frac{(N-X)}{N} \times S$$

Particles Filter

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

❑ How to get new particles ?





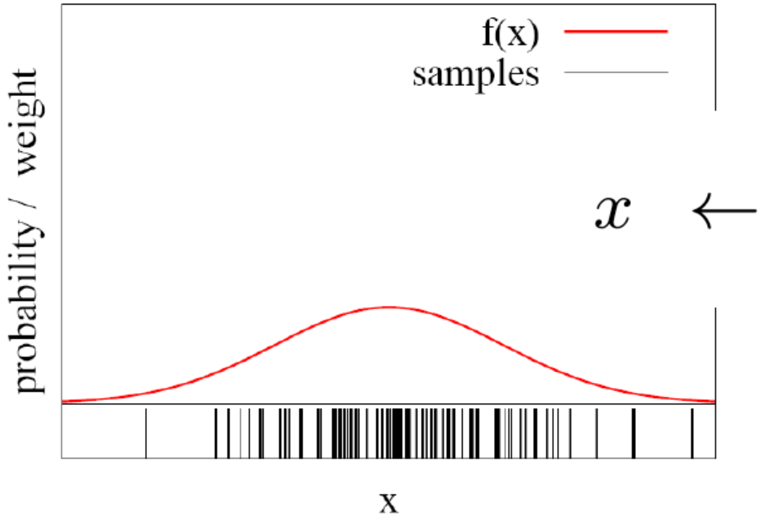
The **more particles fall into a region**, the **higher** the **probability** of the region

How to obtain such samples?

Particles Filter

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

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



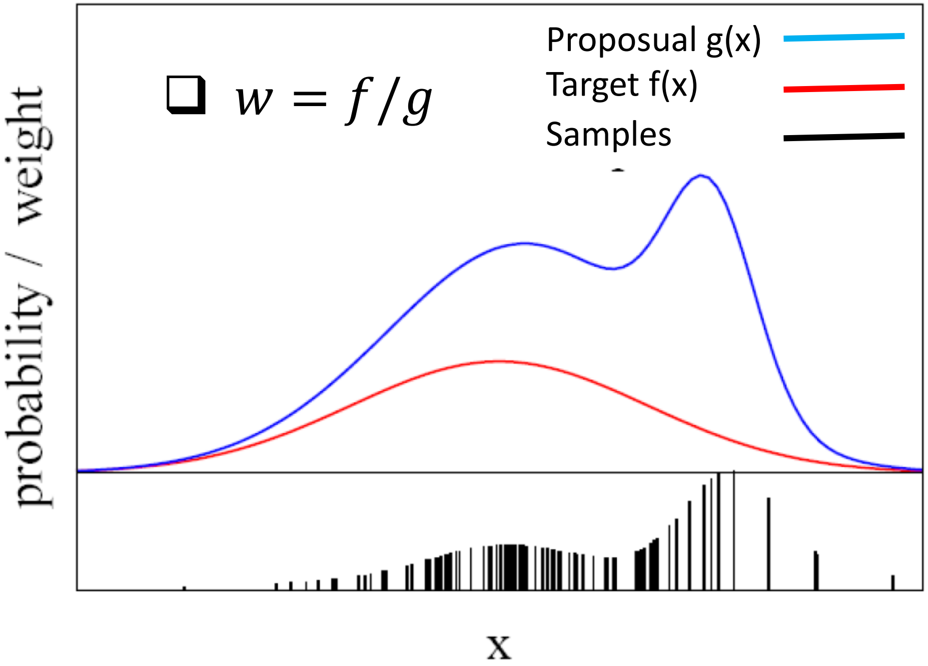
$$x \leftarrow \frac{1}{2} \sum_{i=1}^{12} \mathrm{rand}(-\sigma, \sigma)$$

How to sample from **other** distributions?

Particles Filter

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

- ❑ How to get new particles ?
  - ❑ 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 $\boldsymbol{w} = \boldsymbol{f}/\boldsymbol{g}$



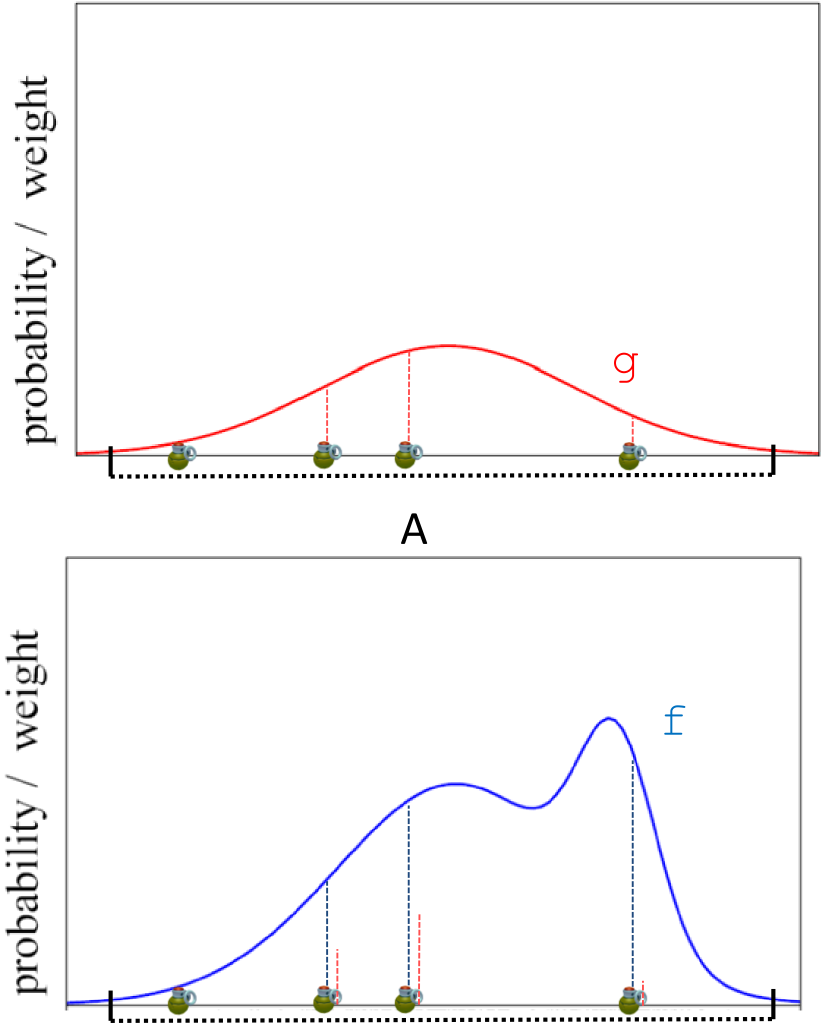
Particles Filter

- ❑ How to get new particles ?
  - ❑ Explanations

$$\frac{1}{M}\sum_{m=1}^{M} I(x^{[m]} \in A) \rightarrow \int_{A} g(x)dx$$

$$w^{[m]} = \frac{f(x^{[m]})}{g(x^{[m]})}$$

$$\frac{1}{\sum_{m=1}^{M} w^{[m]}}\sum_{m=1}^{M} I(x^{[m]} \in A)w^{[m]} \rightarrow \int_{A} f(x)dx$$



Thrun S. and Burgard W. and Fox D. Probabilistic Robotics (book), 2005

Particles Filter

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

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

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

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

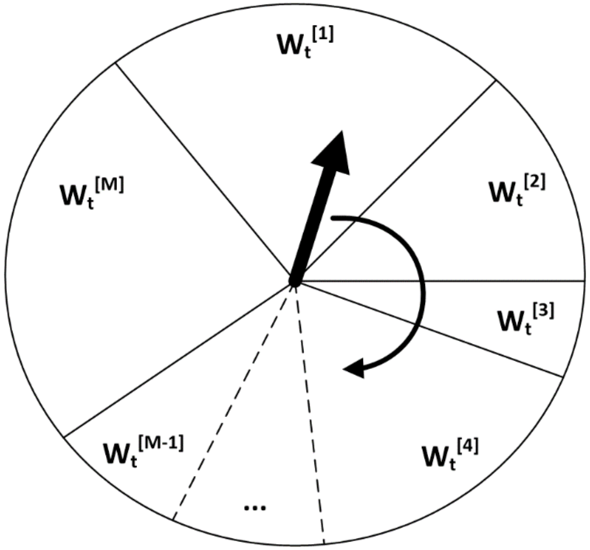
Particles Filter

- ❑ How to get new particules ?

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

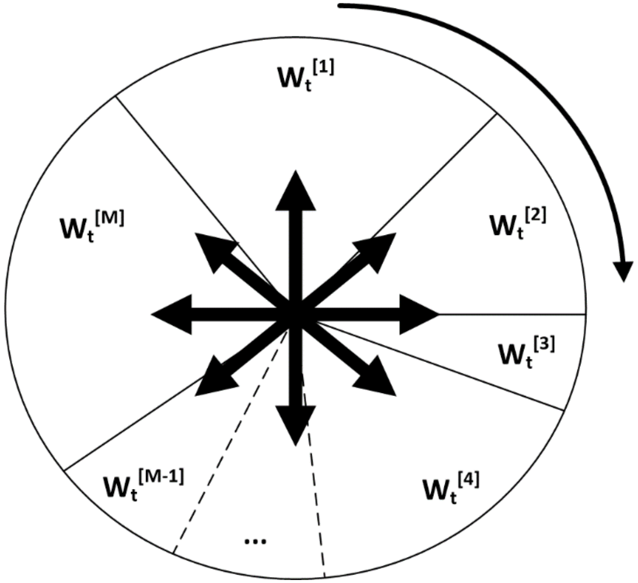
The **more particles fall into a region**, the **higher** the **probability** of the region

Roulette wheel



O (M Log M)

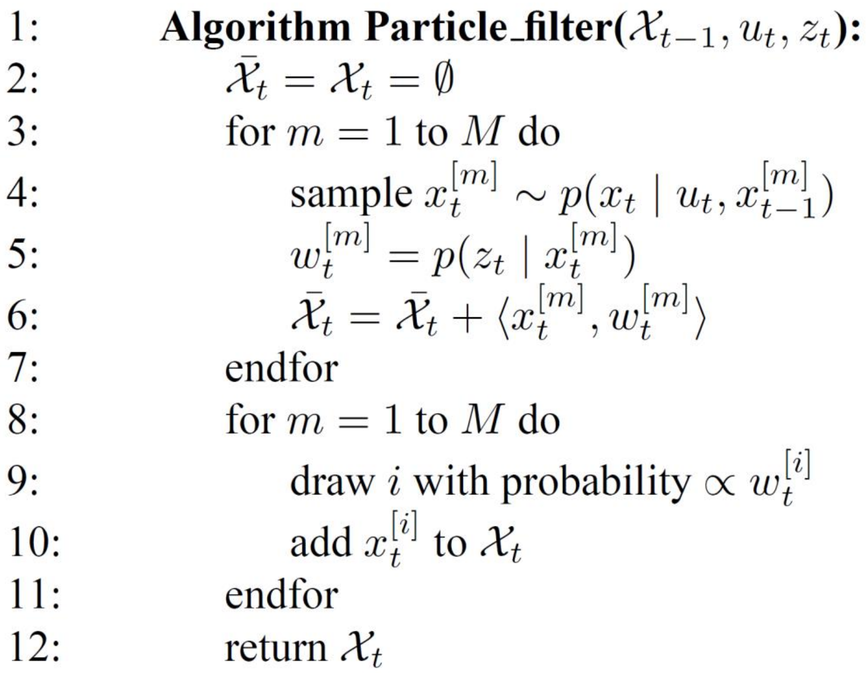
Stochastic universal sampling



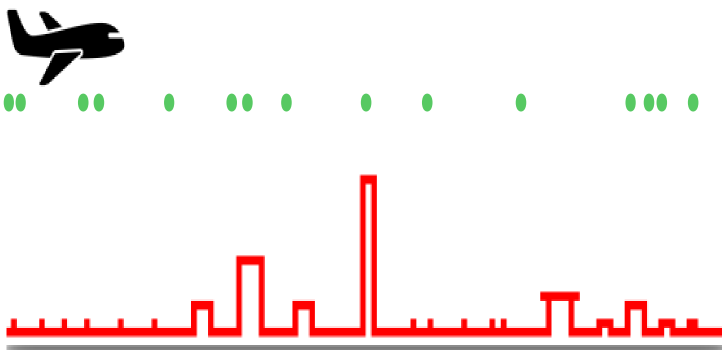
O (M)

Particles 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

Particles Filter

- ❑ Example

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

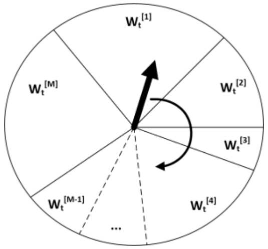
$$u_t \sim Rand\ Uniform\ [-1, 10] = \Delta x_t$$

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

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

$$w_t{}^{[m]} = \begin{cases} 0{,}9\ if\ z_t = esimate(\boldsymbol{z_t}^{[\boldsymbol{m}]}) \\ \quad\quad\quad 0{,}1 \end{cases}$$

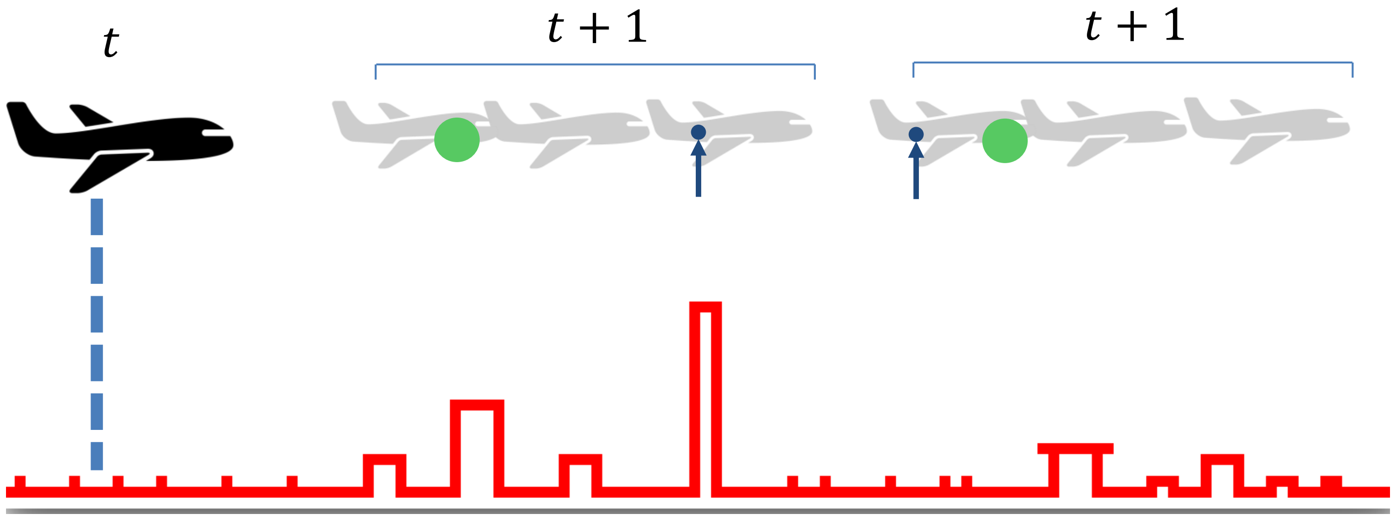
How to get new particles ?



Particles 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
...
```

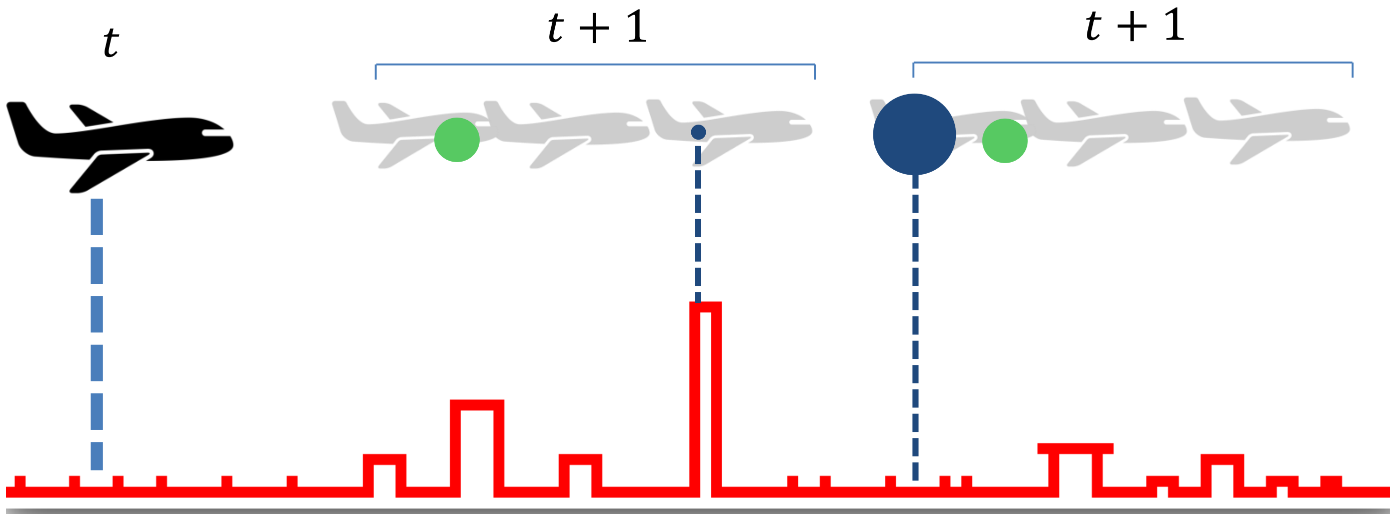


$$u_t \sim Rand\ Uniform\ [-1, 10] = \ \Delta x_t$$

Particles Filter

```
...
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
...
```

- ❑ Example

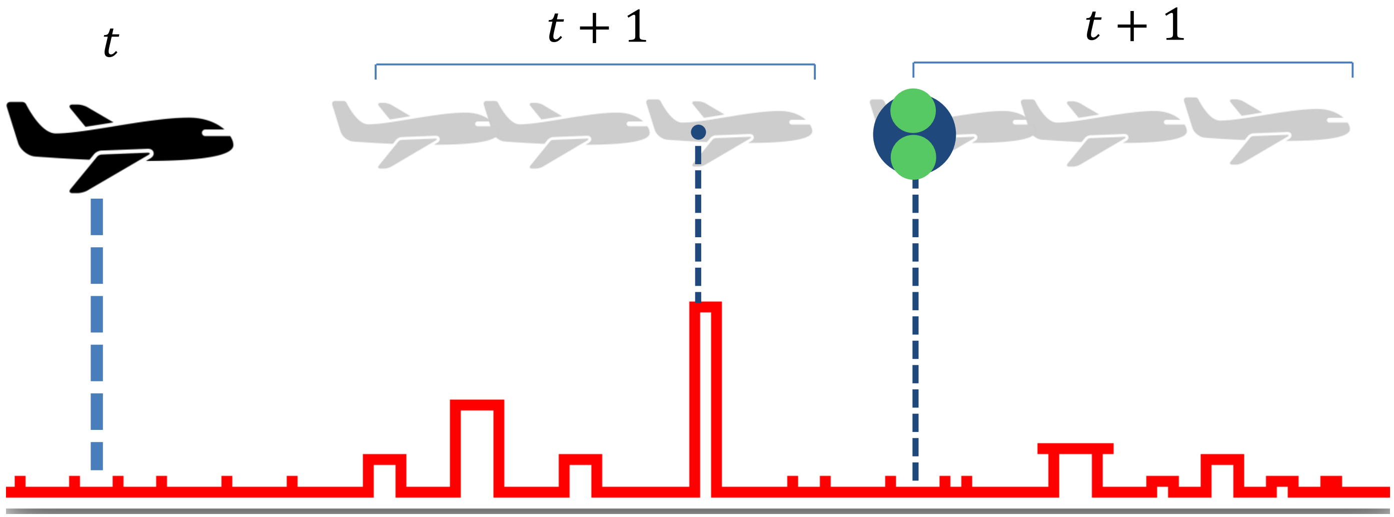


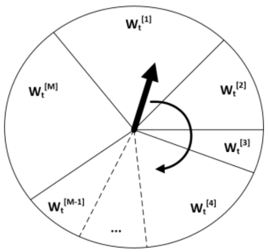
$$w_t{}^{[m]} = \begin{cases} 0{,}9 \; if \; z_t = esimate(\boldsymbol{z_t}{}^{[m]}) \\ \qquad\qquad 0{,}1 \end{cases}$$

Particles 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
…
```

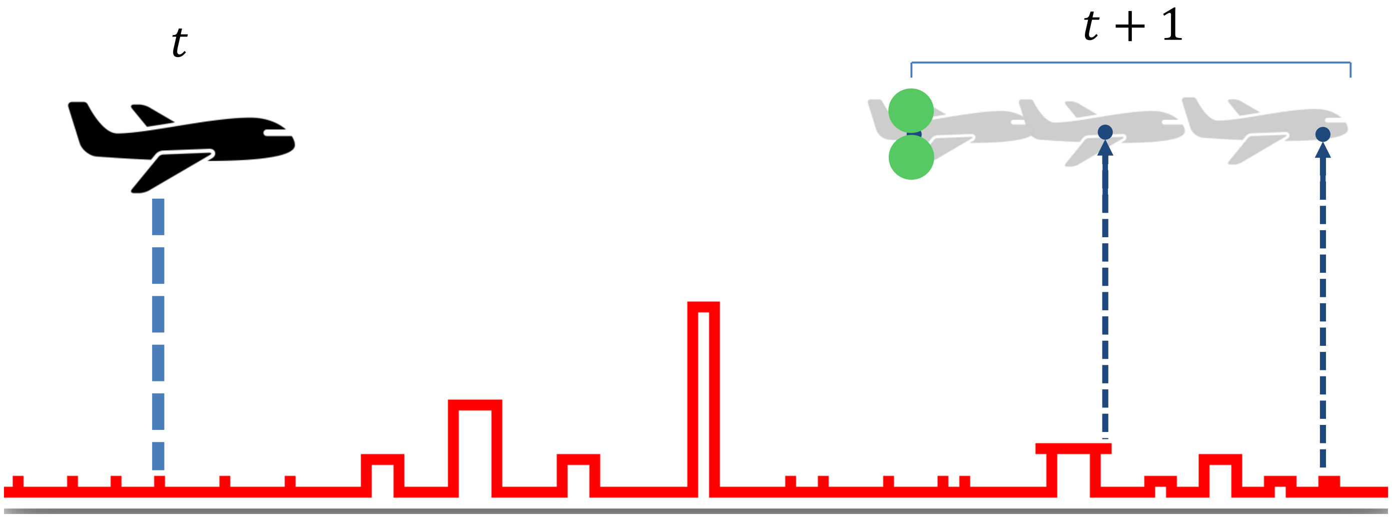


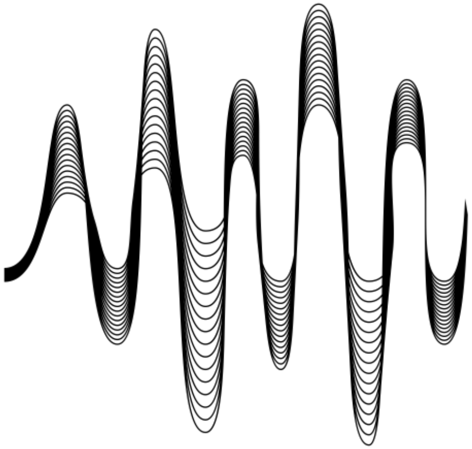


Particles Filter

❑ 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
…
```





Motion and Sensors data Modeling

Motion Model

- ❑ Odometry model (e.g vehicule) Thrun S. and Burgard W. and Fox D. Probabilistic Robotics (book), 2005
  - ❑ Hypothesis:
  - ❑ Robot in pose $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{\theta}})$ needs to go to $(\overline{\boldsymbol{x}}', \overline{\boldsymbol{y}}', \overline{\boldsymbol{\theta}}')$
  - ❑ Odometry information can be defined as a set of rotations and translations $\boldsymbol{u} = (\boldsymbol{\delta_{rot1}}, \boldsymbol{\delta_{trans}}, \boldsymbol{\delta_{rot2}})$

$$\boldsymbol{\delta_{trans}} = \sqrt{(\overline{\boldsymbol{x}}' - \overline{\boldsymbol{x}})^2 + (\overline{\boldsymbol{y}}' - \overline{\boldsymbol{y}})^2}$$

$$\boldsymbol{\delta_{rot1}} = \boldsymbol{atang2}(\overline{\boldsymbol{y}}' - \overline{\boldsymbol{y}}, \boldsymbol{x}' - \overline{\boldsymbol{x}}) - \overline{\boldsymbol{\theta}}$$

$$\boldsymbol{\delta_{rot2}} = \overline{\boldsymbol{\theta}}\text{'-}\ \overline{\boldsymbol{\theta}} - \boldsymbol{\delta_{rot1}}$$

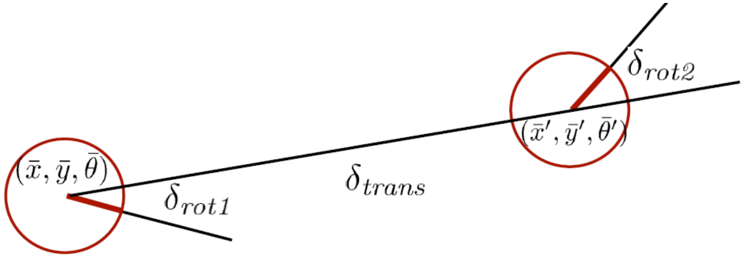

δ_{rot2}
$(\bar{x}', \bar{y}', \bar{\theta}')$
$(\bar{x}, \bar{y}, \bar{\theta})$
δ_{rot1}
δ_{trans}

Image: Cyrill Stachniss, Robot Mapping Courses, UNI Freiburg

Motion Model

- ❑ Odometry model (e.g vehicule)

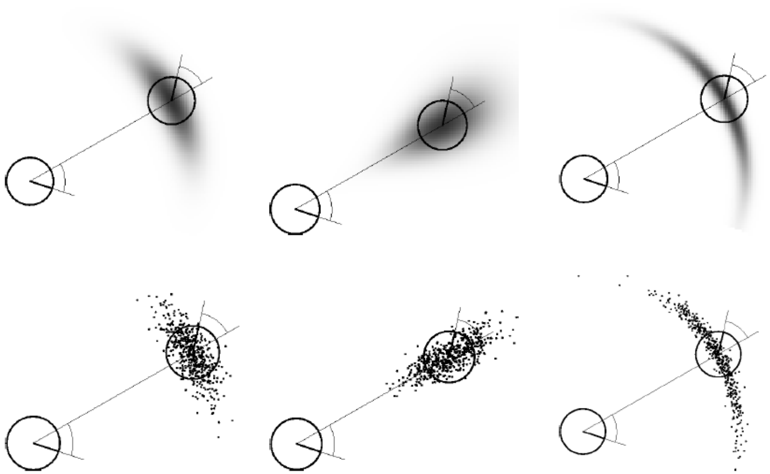
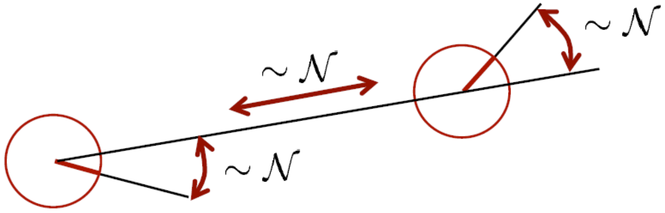


Image from : Thrun S. and Burgard W. and Fox D. Probabilistic Robotics (book), 2005.

- ❑ Noise in Odometry (e.g Gaussian Noise)



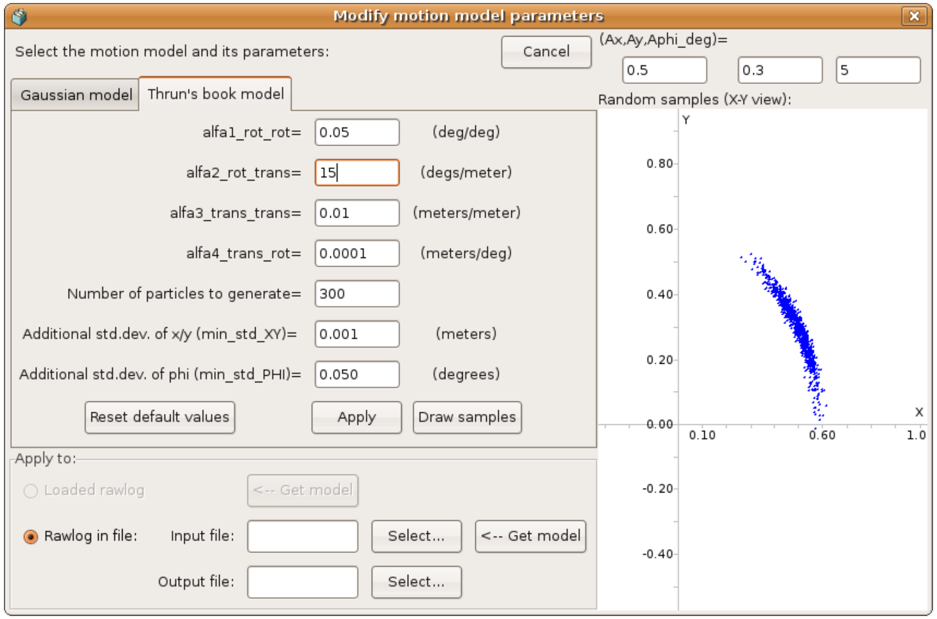
$$\begin{aligned} \hat{\delta}_{trans} &= \delta_{trans} + \epsilon_{trans} & \epsilon_{trans} &\sim \mathcal{N}(0, \sigma^2_{trans}) \\ \hat{\delta}_{rot1} &= \delta_{rot1} + \epsilon_{rot1} & \epsilon_{rot1} &\sim \mathcal{N}(0, \sigma^2_{rot1}) \\ \hat{\delta}_{rot2} &= \delta_{rot2} + \epsilon_{rot2} & \epsilon_{rot2} &\sim \mathcal{N}(0, \sigma^2_{rot2}) \end{aligned}$$

Approximation for stand deviation computation:

$$\begin{aligned} \sigma_{rot1} &= \alpha_1 |\delta_{rot1}| + \alpha_2 \delta_{trans} \\ \sigma_{trans} &= \alpha_3 \delta_{trans} + \alpha_4 (|\delta_{rot1}| + |\delta_{rot2}|) \\ \sigma_{rot2} &= \alpha_1 |\delta_{rot2}| + \alpha_2 \delta_{trans} \end{aligned}$$

Motion Model

- ❑ Odometry model (e.g vehicule)



Select the motion model and its parameters:

(Ax,Ay,Aphi_deg)= 0.5 0.3 5

Random samples (X-Y view):

alfa1_rot_rot=	0.05	(deg/deg)
alfa2_rot_trans=	15	(degs/meter)
alfa3_trans_trans=	0.01	(meters/meter)
alfa4_trans_rot=	0.0001	(meters/deg)
Number of particles to generate=	300	
Additional std.dev. of x/y (min_std_XY)=	0.001	(meters)
Additional std.dev. of phi (min_std_PHI)=	0.050	(degrees)

Apply to:

○ Loaded rawlog

● Rawlog in file: Input file:

Output file:

$$
\begin{aligned}
\hat{\delta}_{trans} &= \delta_{trans} + \epsilon_{trans} & \epsilon_{trans} &\sim \mathcal{N}(0, \sigma^2_{trans}) \\
\hat{\delta}_{rot1} &= \delta_{rot1} + \epsilon_{rot1} & \epsilon_{rot1} &\sim \mathcal{N}(0, \sigma^2_{rot1}) \\
\hat{\delta}_{rot2} &= \delta_{rot2} + \epsilon_{rot2} & \epsilon_{rot2} &\sim \mathcal{N}(0, \sigma^2_{rot2})
\end{aligned}
$$

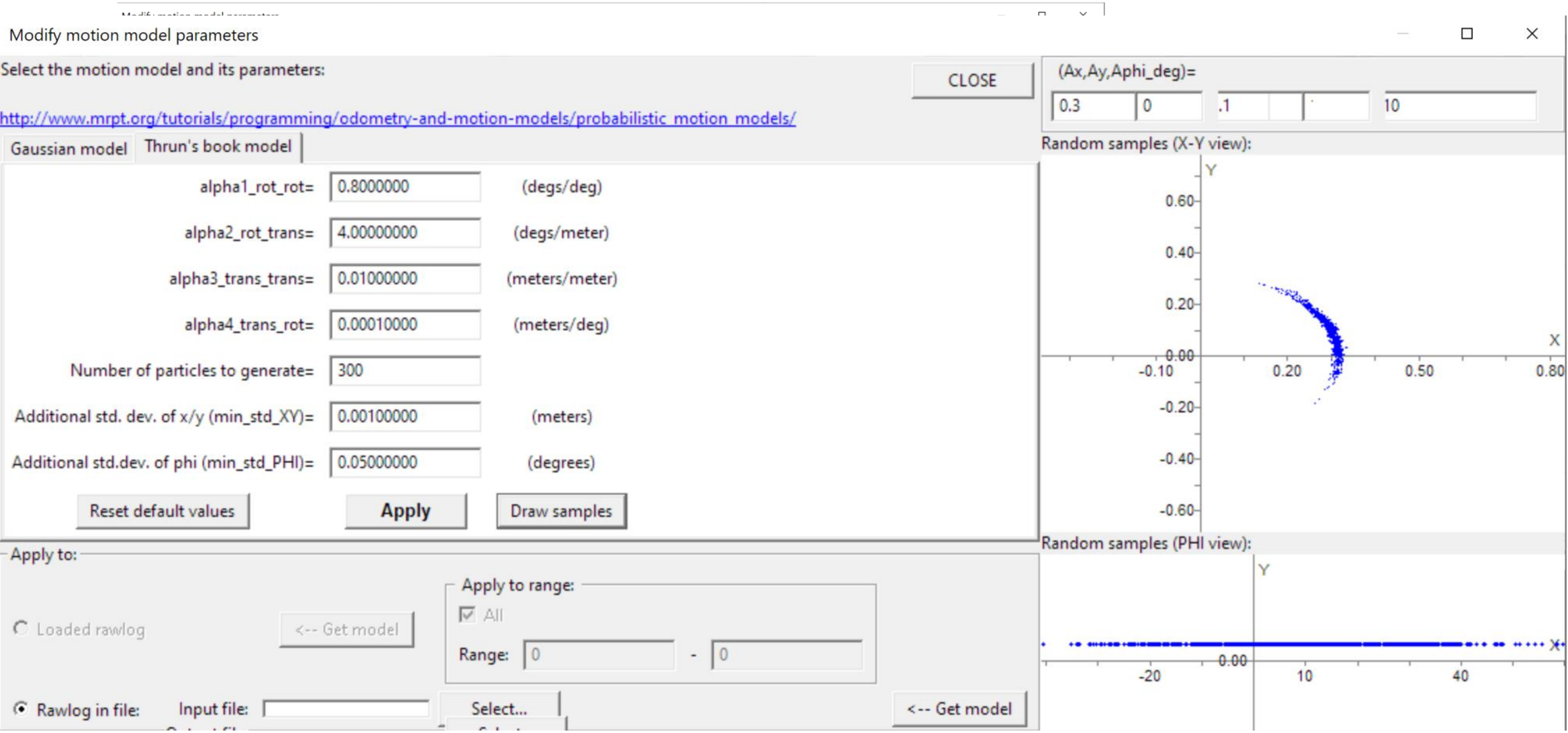
Approximation for stand deviation computation:

$$
\begin{aligned}
\sigma_{rot1} &= \alpha_1 |\delta_{rot1}| + \alpha_2 \delta_{trans} \\
\sigma_{trans} &= \alpha_3 \delta_{trans} + \alpha_4 (|\delta_{rot1}| + |\delta_{rot2}|) \\
\sigma_{rot2} &= \alpha_1 |\delta_{rot2}| + \alpha_2 \delta_{trans}
\end{aligned}
$$

https://www.mrpt.org/tutorials/programming/odometry-and-motion-models/probabilistic_motion_models/

Motion Model

- ❑ Odometry model (e.g vehicule)



Modify motion model parameters

Select the motion model and its parameters:

http://www.mrpt.org/tutorials/programming/odometry-and-motion-models/probabilistic_motion_models/

Thrun's book model

alpha1_rot_rot= 0.8000000 (degs/deg)

alpha2_rot_trans= 4.00000000 (degs/meter)

alpha3_trans_trans= 0.01000000 (meters/meter)

alpha4_trans_rot= 0.00010000 (meters/deg)

Number of particles to generate= 300

Additional std. dev. of x/y (min_std_XY)= 0.00100000 (meters)

Additional std.dev. of phi (min_std_PHI)= 0.05000000 (degrees)

(Ax,Ay,Aphi_deg)= 0.3 0 .1 10

Apply to:

○ Loaded rawlog

Apply to range: ☑ All

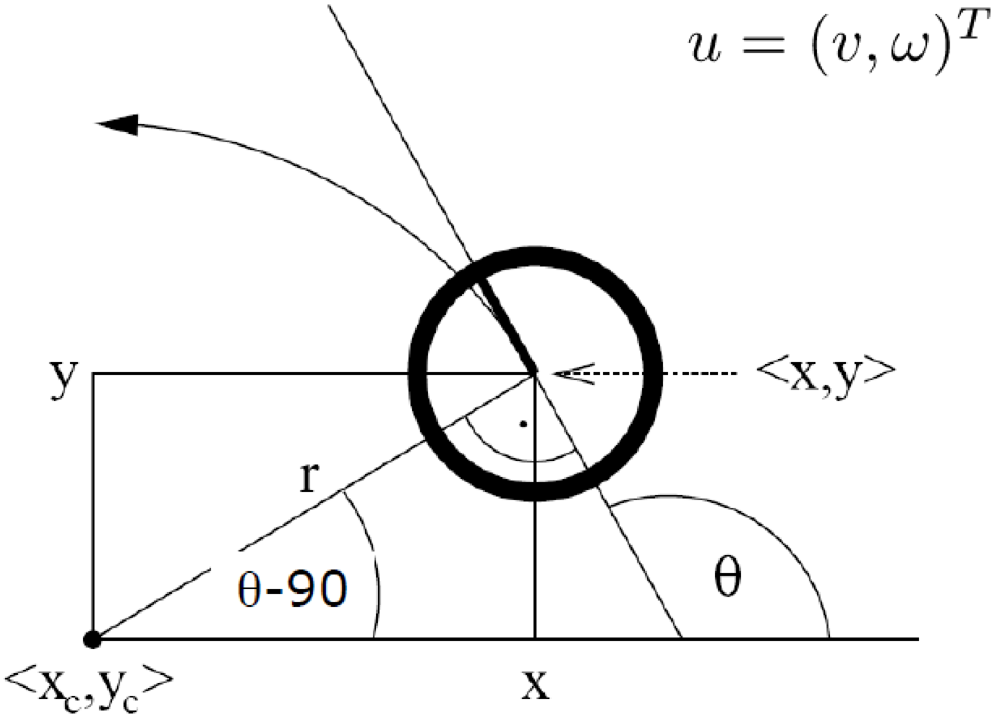
Range: 0 - 0

● Rawlog in file: Input file:

Motion Model

- ❑ Velocity model (e.g vehicule)

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



$$u = (v, \omega)^T$$

Motion Model

- ❑ Velocity model (e.g vehicule)

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

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

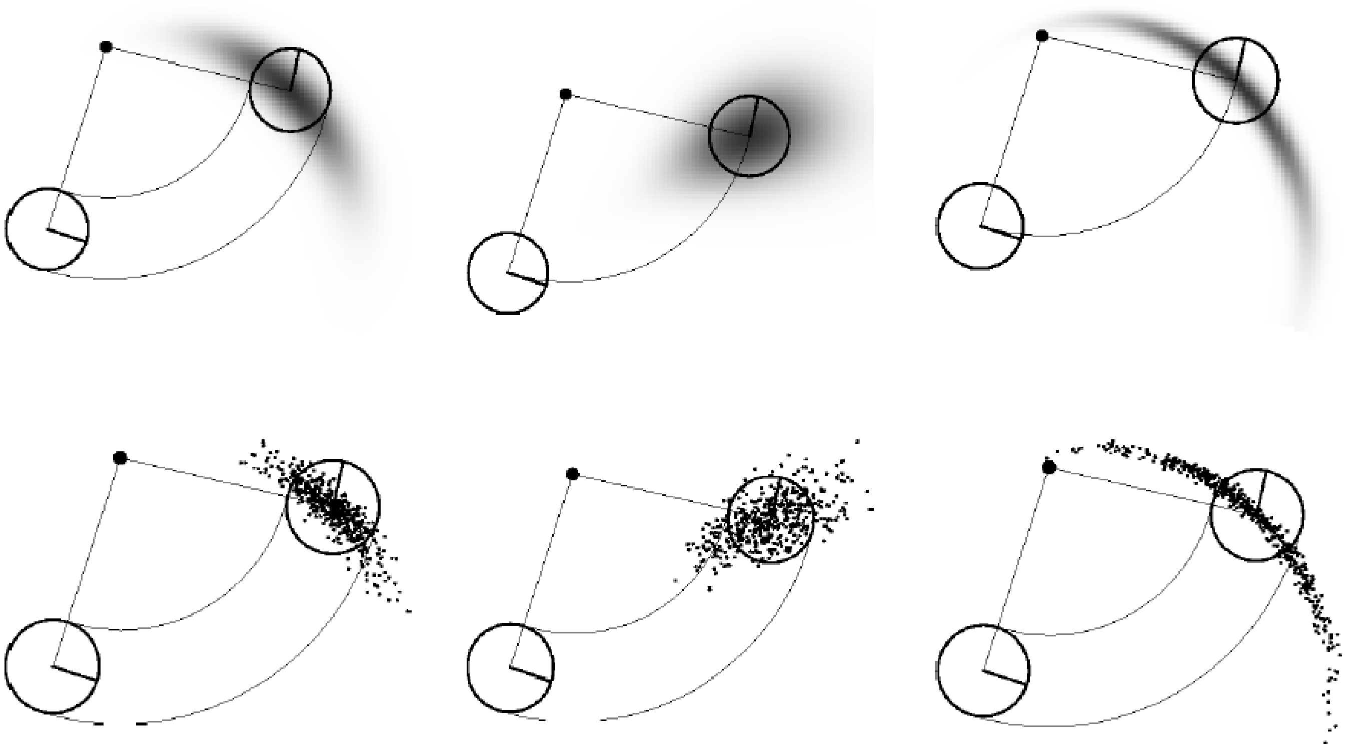
$$\uparrow$$

Term to account for the final rotation

Motion Model

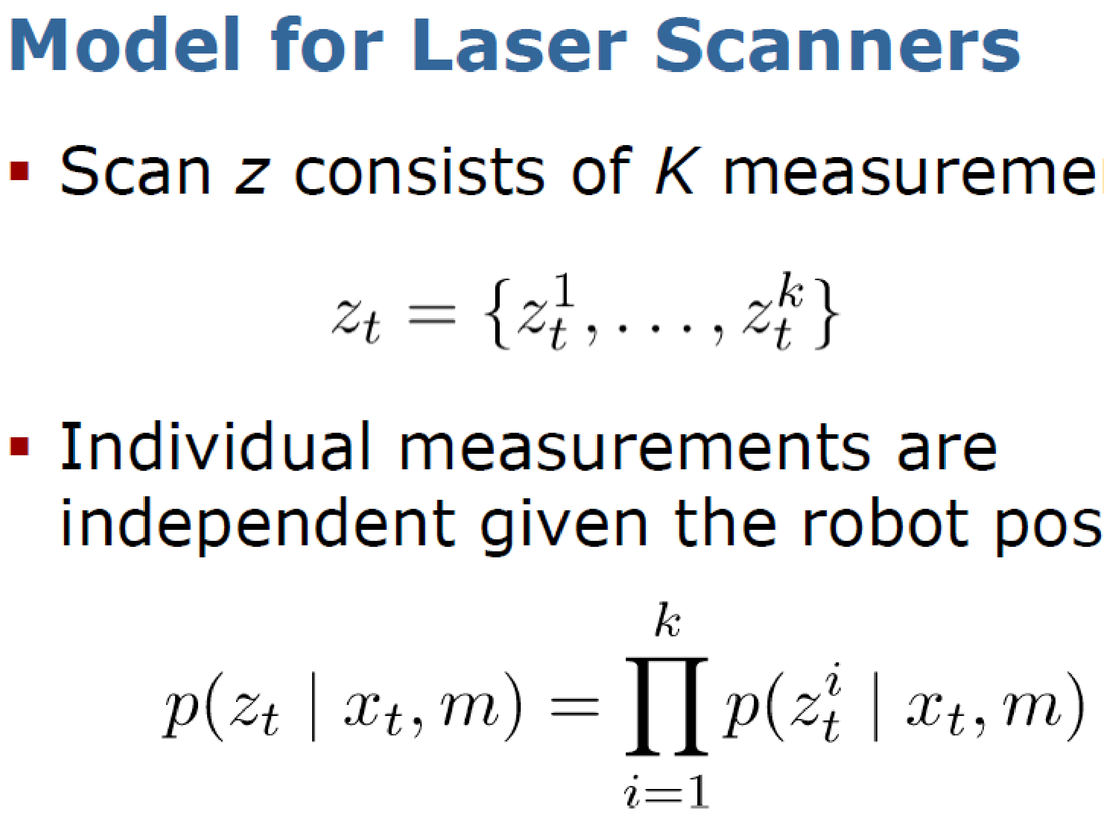
- ❑ Velocity model (e.g vehicule)

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.

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

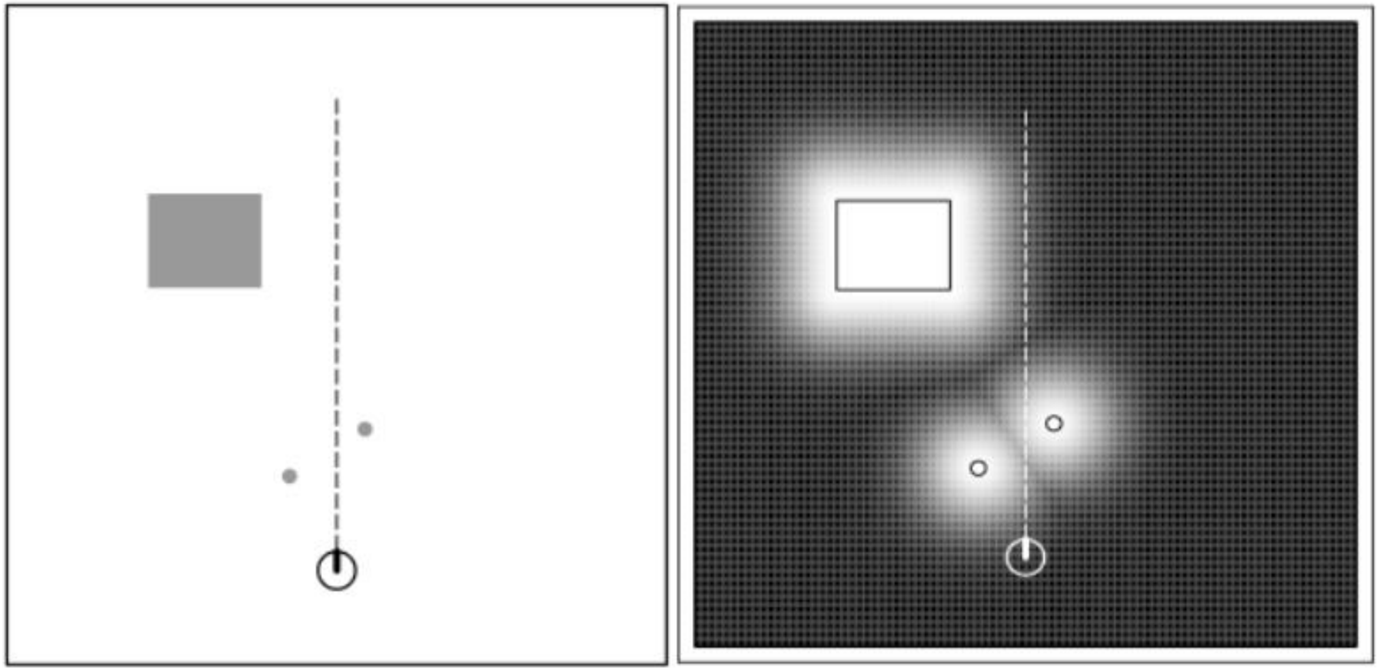
- Individual measurements are independent given the robot position

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

Sensor Model

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

Beam-Endpoint Model

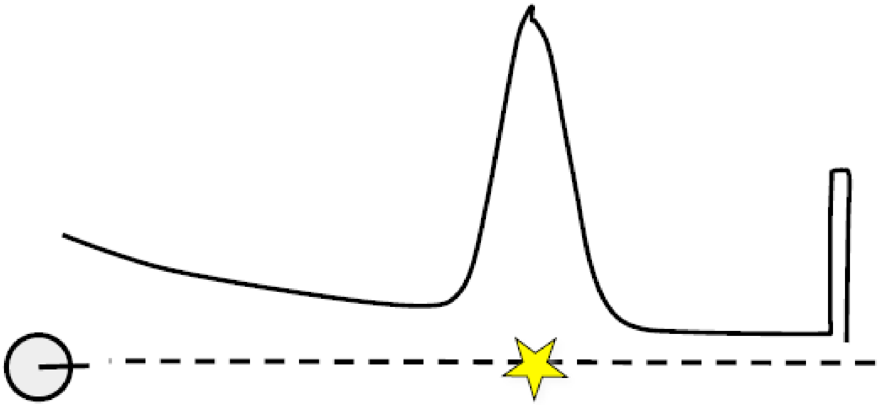


Sensor Model

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