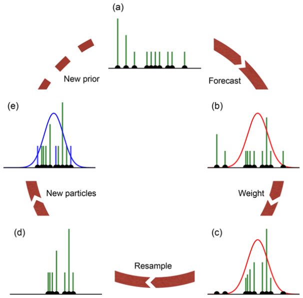
# Autonomous Mobile Robotics

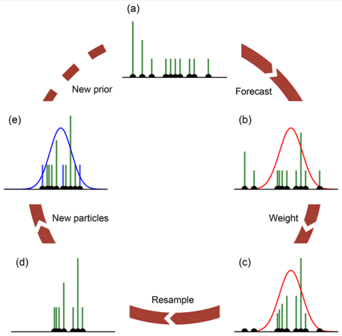
## Particle Filter

Geesara Kulathunga

November 2, 2022



# Particle Filter




https://hess.copernicus.org/articles/23/1163/2019/

# Particle Filter

Why do we need particle filter? Let's try to understand common problems we face in this contest

1. Do we need more than one landmarks for localization? Are we interested in tracking multiple objects simultaneously?
2. Are objects we are interested in visible all the time or have some occlusions?
3. How do we address the non-linear behaviour of the system and nonlinear measurements?
4. Have you ever considered about non-Gaussian noise? Can you explain a bit more on this?
5. How many states to be tracked? How do we define these states? Are they discrete or continuous?
6. When the process model is unknown, what can we do for that?

# Bayesian Filter

$$p(x_k|z_{1:k}, u_{0:k-1}) = \frac{p(z_k|x_k, z_1, ..., z_{k-1}, u_{0:k-1})p(x_k|z_1, ..., z_{k-1}, u_{0:k-1})}{p(z_k|z_{1:k-1}, u_{0:k-1})} = \frac{p(z_k|x_k)p(x_k|z_{1:k-1}, u_{0:k-1})}{p(z_k|z_{1:k-1}, u_{0:k-1})} \tag{1}$$

- $p(x_k|z_{1:k}, u_{0:k-1})$ state probability distribution at time step k, updated with measurement data and control inputs
- $p(z_k|x_k, z_{1:k-1}, u_{0:k-1})$ measurement probability distribution
- $p(x_k|z_{k-1}, u_{0:k-1})$ predicted state probability distribution
- $p(z_k|z_{1:k-1}, u_{0:k-1})$ measurement probability distribution

# Bayesian Filter

- Prediction step

$$p(x_k|z_{1:k-1}, u_{1:k-1}) = \Sigma_{x_{k-1} \in X}\ p(x_k|x_{k-1}, u_{k-1}) p(x_{k-1}|z_{1:k-1}, u_{0:k-1}) \tag{2}$$

- Correction step

$$p(x_k|z_{1:k}, u_{0:k-1}) = \frac{p(z_k|x_k) p(x_k|z_{1:k-1}, u_{0:k-1})}{p(z_k|z_{1:k-1}, u_{0:k-1})} \tag{3}$$

, where

$$p(z_k|z_{1:k-1}, u_{0:k-1}) = \Sigma_{x_k \in X}\ p(z_k|x_k) p(x_k|z_{1:k-1}, u_{0:k-1})$$

# Partical Filter

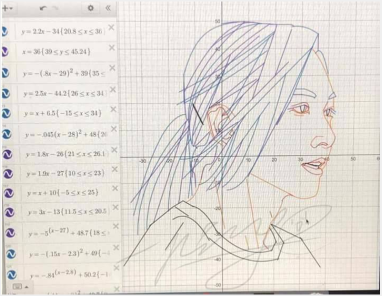
An explicit solution of Bayesian filter for the continuous state-space variables

$$p(x_k|z_{1:k}, u_{0:k-1}) \\ = \frac{p(z_k|x_k)}{p(z_k|z_{1:k-1}, u_{0:k-1})} \int p(x_k|x_{k-1}, u_{k-1}) p(x_{k-1}|z_{1:k-1}, u_{0:k-2}) dx_{k-1}$$

With the Markov assumption, the most probable state estimate

$$E\{\hat{x}_{k|k}\} = \int x_{k|k} \cdot p(x_k|z_{1:k}, u_{0:k-1}) dx_{k|k}$$

However, this form can be applied only where data distribution is Gaussian and system has to be linear, result is a Kalman filter. When the system is nonlinear result in EKF. Particle filter on the other had more general case where noise does not need to be Gaussian.


$y = 2.2x - 34\{20.8 \le x \le 36$
$x = 36\{39 \le y \le 45.24\}$
$y = -(.8x - 29)^2 + 39\{35 \le$
$y = 2.5x - 44.2\{26 \le x \le 34$
$y = x + 6.5\{-15 \le x \le 34\}$
$y = -.045(x - 28)^2 + 48\{2$
$y = 1.8x - 26\{21 \le x \le 26.1$
$y = 1.9x - 27\{10 \le x \le 23\}$
$y = x + 10\{-5 \le x \le 25\}$
$y = 3x - 13\{11.5 \le x \le 20.5$
$y = -5^{(x-27)} + 48.7\{18 \le$
$y = -(.15x - 2.3)^2 + 49\{-$
$y = -.84^{(x-2.8)} + 50.2\{-1$

**(a)**

**(b)**

# Monte Carlo Integration (MCI)

MCI is used to generate a finite number of random samples and pass through a system that we are interested in followed by computing the result on the transformed points. MCI is quite a powerful method for computing the value of complex integrals using probabilistic techniques

1. Generating set of points from a chosen distribution, $X_i = U[a, b]$, $i = 1, ..., N$
2. Calculate $Y_i = g(X_i)$, $i = 1, ..., N$ where g(X) is a function which is used to generate a point, in other words, a function to be integrated.
3. Calculate the estimation of g(x):

$$\bar{Y} = \frac{(b-a)}{N} \Sigma_{i=1}^{N} Y_i$$

4. Use $g(x) = 4\sqrt{1-x^2}$, try to estimate $\bar{Y}$? what will happen when N is increasing?

### Example 01

Can you estimate the value of $\pi$?

# References


Gregor Klancar, Andrej Zdesar, Saso Blazic, and Igor Skrjanc.
***Wheeled mobile robotics: from fundamentals towards autonomous systems.***
Butterworth-Heinemann, 2017.

Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza.

***Introduction to autonomous mobile robots.***
MIT press, 2011.

Sebastian Thrun.
**Probabilistic robotics.**
*Communications of the ACM*, 45(3):52–57, 2002.