

Particle Filter

Particle Filter SLAM

Kinematic (실제로봇 위치 이동)
Prediction (가상로봇 위치 이동)
observe (실제로봇과 비콘들 측정)
Add Landmark (측정된 비콘 거리, 각도, ID)
Importance Weight (Particle 오차에 따른 Weight)
Resample (퍼져있는 Particle 모아줌)

■ Kinematic

Xr : Real Robot position

$$\mathbf{Xr} = \begin{pmatrix} \mathbf{Xr}_x \\ \mathbf{Xr}_y \\ \mathbf{Xr}_\theta \end{pmatrix} = \begin{pmatrix} \mathbf{Xr}_x \\ \mathbf{Xr}_y \\ \mathbf{Xr}_\theta \end{pmatrix} + \begin{pmatrix} \mathbf{u}_{\text{dist}} * \cos[\mathbf{Xr}_\theta] \\ \mathbf{u}_{\text{dist}} * \sin[\mathbf{Xr}_\theta] \\ \mathbf{u}_{\text{rad}} \end{pmatrix} + \begin{pmatrix} \mathbf{s}_{\text{dist}} * \cos[\mathbf{Xr}_\theta] \\ \mathbf{s}_{\text{dist}} * \sin[\mathbf{Xr}_\theta] \\ \mathbf{s}_{\text{rad}} \end{pmatrix}$$

■ Prediction

P(.) .Xk : Prediction Robot position

L = 1 : n_particle

$$\mathbf{P}[L] . \mathbf{Xk} = \begin{pmatrix} \mathbf{Xk}_x \\ \mathbf{Xk}_y \\ \mathbf{Xk}_\theta \\ \mathbf{Yk}_x^{z[1]} \\ \mathbf{Yk}_y^{z[1]} \\ \vdots \\ \mathbf{Yk}_x^{z[n]} \\ \mathbf{Yk}_y^{z[n]} \end{pmatrix}$$

$$\mathbf{P}[L] . \begin{pmatrix} \mathbf{Xk}_x \\ \mathbf{Xk}_y \\ \mathbf{Xk}_\theta \end{pmatrix} = \mathbf{P}[L] . \begin{pmatrix} \mathbf{Xk}_x \\ \mathbf{Xk}_y \\ \mathbf{Xk}_\theta \end{pmatrix} + \begin{pmatrix} \mathbf{u}_{\text{dist}} * \cos[\mathbf{Xk}_\theta] \\ \mathbf{u}_{\text{dist}} * \sin[\mathbf{Xk}_\theta] \\ \mathbf{u}_{\text{rad}} \end{pmatrix} + \begin{pmatrix} \mathbf{s}_{\text{dist}} * \cos[\mathbf{Xk}_\theta] \\ \mathbf{s}_{\text{dist}} * \sin[\mathbf{Xk}_\theta] \\ \mathbf{s}_{\text{rad}} \end{pmatrix}$$

Sdist Srad : noise

각 Particle 에서 noise는 다르기 때문에 Particle이 퍼짐

■ observe

Real Robot position Xr에서 측정 가능한 Beacon들 간 거리와 각도 z[n] 처리

$$\begin{pmatrix} z[n]_{\text{dist}} \\ z[n]_{\text{rad}} \end{pmatrix} = \begin{pmatrix} \sqrt{(\text{Beacon}[n]_x - Xr_x)^2 + (\text{Beacon}[n]_y - Xr_y)^2} \\ \text{atan2}(\text{Beacon}[n]_y - Xr_y, \text{Beacon}[n]_x - Xr_x) - Xr_\theta \end{pmatrix}$$

■ Add Landmark

측정된 Beacon중 저장된 landmark 중 $z[n]$ 가 없으면 새로 landmark에 추가

각 particle에 있는 prediction robot position에서 측정된 Beacon 중 $z[n]_{\text{dist}}$, $z[n]_{\text{rad}}$ 거리에 Landmark 추가

$$P[L] \cdot \begin{pmatrix} Yk_x^{z[n]} \\ Yk_y^{z[n]} \end{pmatrix} = P[L] \cdot \begin{pmatrix} Xk_x \\ Xk_y \end{pmatrix} + \begin{pmatrix} z[n]_{\text{dist}} * \cos[z[n]_{\text{rad}} + P[L] \cdot Xk_\theta] \\ z[n]_{\text{dist}} * \sin[z[n]_{\text{rad}} + P[L] \cdot Xk_\theta] \end{pmatrix}$$

■ Importance Weight

각 Particle에서 (측정된 $z[n]_{\text{dist}}$, $z[n]_{\text{rad}}$) 는

(prediction robot position 와 해당 Landmark 간의 거리, 각도) 차에 대한 오류를 구한다
하나의 Particle 에 SSE 오차를 (여러개의 측정된 값에서 구한 오차들을 더해줌) 구함

SSE 갯수는 Particle 갯수와 같음

$$\begin{pmatrix} z[n]_{\text{dist}}^{(L)} \\ z[n]_{\text{rad}}^{(L)} \end{pmatrix} = \begin{pmatrix} \sqrt{(P[L] \cdot Yk_x^{z[n]} - P[L] \cdot Xk_x)^2 + (P[L] \cdot Yk_y^{z[n]} - P[L] \cdot Xk_y)^2} \\ \text{atan2}(P[L] \cdot Yk_y^{z[n]} - P[L] \cdot Xk_y, P[L] \cdot Yk_x^{z[n]} - P[L] \cdot Xk_x) - P[L] \cdot Xk_\theta \end{pmatrix}$$

$$\begin{aligned} \text{SSE}[L] = & (z[1]_{\text{dist}}^{(L)} - z[1]_{\text{dist}})^2 + (z[1]_{\text{rad}}^{(L)} - z[1]_{\text{rad}})^2 + \\ & \dots + (z[n]_{\text{dist}}^{(L)} - z[n]_{\text{dist}})^2 + (z[n]_{\text{rad}}^{(L)} - z[n]_{\text{rad}})^2 \end{aligned}$$

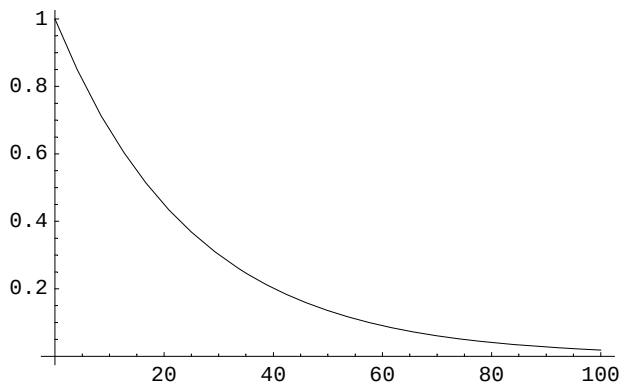
SSE로 Weight구함

SSE 는 오차이기에 때문에 작은 SSE 값을 큰 weight 값 으로 변환

$$Wk[L] = Wk[L] * e^{-\frac{4 \text{SSE}[L]}{\max[\text{SSE}]}}$$

$$Wk[L] = \frac{Wk[L]}{\sum[Wk]} : \text{weight normalize}$$

$$e^{-\frac{4 \text{SSE}[L]}{\max[\text{SSE}]}}$$



■ Resample

Wk cdf (Cumulative Distribution Function)

Importance Weight 누적 분포 함수 구한 뒤
오차 분포에서 특정 **threshold**를 넘으면
다시 오차가 **threshold** 넘지 않는 것들로 배치 한다.

퍼져있는 **Particle**을 모아준다.

Doc