▨ 평균 필터 (average filter)

=> 데이터를 다 저장하기 힘들어 (평균값, 추가데이터, 데이터 갯수만) 이용

$\bar{x}_k = (x_1 + x_2 + \cdots + x_k)/k$ => $\bar{x}_k = \frac{k-1}{k}\bar{x}_{k-1} + \frac{1}{k}x_k$

▨ 이동 평균 필터 (moving average filter)

=> 너무 오래된 데이터는 고려 안하고 싶어 최근 몇개의 측정값만 가지고 평균 냄

$\bar{x}_k - \bar{x}_{k-1} = \frac{x_k - x_{k-n}}{n}$ => $\bar{x}_k = \bar{x}_{k-1} + \frac{x_k - x_{k-n}}{n}$

↳ n개 data 고려

▨ 저주파 통과 필터 (low pass filter)

=> n개 데이터보다도 최근꺼에 가중치를 두고 싶음!

- 1차 저주파 : $\bar{x}_k = \alpha \bar{x}_{k-1} + (1-\alpha) x_k$ (이전평균 × 가중치 + 새데이터 × (1-가중))

$= \alpha(\alpha \bar{x}_{k-2} + (1-\alpha) x_{k-1}) + (1-\alpha) x_k$

↳ $0 < \alpha < 1$ 이므로 $\bar{x}_{k-\cdots}$ 점점 이전 영향은 α^n 으로 n이 늘어나며 영향이 작아진다.

(α가 작다 : 노이즈는 많지만 변화에 민감 (α^n이 너무작아 값들어오면 바로 반영)
(α가 크다 : 노이즈는 적지만 변화에 둔감 (α^n이 1에 가까워 가중치 비슷)

↳ 아마 새로운 이상한 값와도 (노이즈) 반영 비율 작아 노이즈 적음

▨ Localization and tracking

z_t = t지점 센서 값, u_t = t지점 control input 값

local 판단하고 판다 (관점 차이)

(Localization $bel(x_t) = p(x_t \mid z_{1:t}, u_{1:t})$ Control input 있음
(tracking $bel(x_t) = p(x_t \mid z_{1:t})$ Control input 없음 → 비전 예선

Control input 없어도

상태 방정식 (State equation)

Control input 제어입력 noise

$s_0 + vt + \frac{1}{2}at^2$

$$x(k) = x(k-1) + T\dot{x}(k-1) + \frac{T^2}{2}\ddot{x}(k-1) + \frac{T^2}{2}(u(k) + w(k))$$

$$\dot{x}(k) = \dot{x}(k-1) + T\ddot{x}(k-1) + T(u(k) + w(k))$$

$$\ddot{x}(k) = \ddot{x}(k-1) + (u(k) + w(k))$$

$$Z(k) = x(k) + v(k)$$

센서 data

노이즈

$\Downarrow$

state matrix

control matrix

- State eq $\Rightarrow$ $\begin{bmatrix} x(k) \\ \dot{x}(k) \\ \ddot{x}(k) \end{bmatrix} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k-1) \\ \dot{x}(k-1) \\ \ddot{x}(k-1) \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \\ 1 \end{bmatrix} (u(k) + w(k))$

- measure eq $\Rightarrow$ $Z(k) = [1 \ 0 \ 0] \begin{bmatrix} x(k) \\ \dot{x}(k) \\ \ddot{x}(k) \end{bmatrix} + v(k)$

transtion matrix

$\Downarrow$

$$x(k) = Ax(k-1) + B(u(k) + w(k))$$

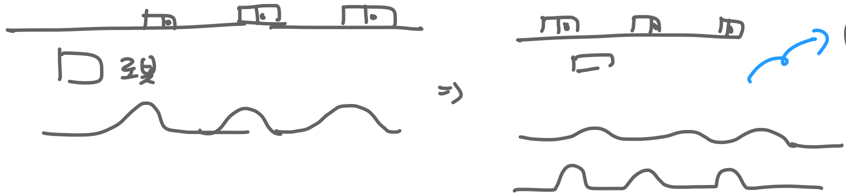
$$z(k) = Cx(k) + v(k)$$

Bayes Rule

Discrete $\quad p(x|b) = \dfrac{p(b|x)\,p(x)}{p(b)} = \dfrac{p(b|x)\,p(x)}{\sum p(b|x')\,p(x')}$

Continuous $\quad p(x|b) = \dfrac{p(b|x)\,p(x)}{p(b)} = \dfrac{p(b|x)\,p(x)}{\int p(b|x')\,p(x')\,dx}$

Base Filter



→ 물리적 이동 예측
센서감소 보정
(문으로)

→ 로봇이 문이다 아니다만 판단가능. 처음에 레이더 돌리면 문으로인해 Gaus 분포 생김

⇒ 움직이면 로봇을 따라가니 분포 높이가 낮아진다. (smooth)

⇒ 움직이고 다시 radar 쏘으면 이전 흐려진 분포와 새위치에서 갖는 정보통해 자신의 위치를 갱신한다 (과거 dist + 현재 dist ⇒ 추론)

∴ recurcive, iterative 하게 현재 위치 추론 ⇒ Bayes Filter

$bel(x_t) = p(x_t \mid z_{1:t}, u_{1:t})$ 지금까지 land mark, control vec 으로 위치추정
(물리적 위치, 이동정보)

베이즈 룰 적용 ⇒
(z 가 나옴)

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

$p(z_{1:t}, u_{1:t})$ → 상수. land, convec 이미 아는 값

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

상수

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

→ (z_t 는 z_{t-1}, u_t 와 관련 X. (Markov assumption))

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

(중요X) z, u는 그대로둠
⇒ x_{t-1} 이 추가되었는데 이는 recursive, 이전정보로 추정하겠다. (Law of total probability)

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

⇒ z는 x_t와 무관 모든 con vec 은 필요X, 딱 이전만 필요

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

⇒ u_t는 x_{t-1} 에 영향X, $u_{1:t} \to u_{1:t-1}$ 로

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

• Bayes filter 개념적 정리

$\eta\, p(z_t|x_t)\, \overline{bel}(x_t)$

$$bel(x_t) = \underbrace{\eta\, p(z_t|x_t)}_{③} \times \underbrace{\int \overbrace{p(x_t|x_{t-1}, u_t)}^{①}\, bel(x_{t-1})\, dx_{t-1}}_{②\ \overline{bel}(x_{t-1})}$$

⇒ 이전 state (확률분포) 정보에 특정 operation 으로 현재 시점예측

⇒ 즉, 과거 $bel(x_{t-1})$ 에서 u_t 현재 control 이용하면 x_t 예측한다.

$p(x) = \int p(x|y)\, p(y)\, dy$

① ⇒ Motion model : Control로 현재 state 추정

② ⇒ Prediction Step : 예측한것 = $\overline{bel}(x_t)$이다.

③ ⇒ Correction Step : predic 값 + observation 정보로 조정해준다.
(observation model) $\eta\, p(z_t|x_t)$

Fun ($bel(x_{t-1})$, u_t, z_t):

for x_t do

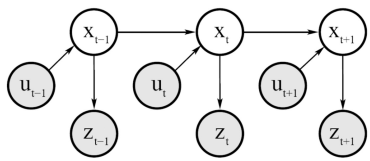
$$\underbrace{\overline{bel}(x_t)}_{③} = \int \underbrace{p(x_t | u_t, x_{t-1})}_{②}\, \underbrace{bel(x_{t-1})}_{①}\, dx$$

$$\underbrace{bel(x_t)}_{⑤} = \underbrace{\eta\, p(z_t|x_t)}_{④}\, \underbrace{\overline{bel}(x_t)}_{③}$$

endfor

return $bel(x_t)$

(상태 모델 : $p(x_t | x_{t-1}, u_t)$
센서 모델 : $p(z_t | x_t)$



① ⇒ 이전상태

② ⇒ $p(x_t| u_t, x_{t-1})$ 이전 상태와 제어값이 주어졌을때 현재상태 확률 분포

③ ⇒ $\overline{bel}(x_t)$ ⇒ Control update (prediction)

④ ⇒ $p(z_t| x_t)$: 현재 상태의 센서값의 확률분포

⑤ ⇒ $bel(x_t)$: measurement update (correction)

u_t = 로봇위치, 랜드마크 위치 (State)

Σ = 위에대한 분산? (확률도 업데이트한다.)

KF => 실제 parameter 들 계산 할수 있음을 보여줌.

그냥 추상적인 것을 가우시안으로 가정해서 parameter 계산가능.

(ρ, θ)

회전 θ 만큼 하고 ρ 만큼 이동 => 회전은 비선형이다

=> 즉 $(\rho\cos\theta, \rho\sin\theta)$ 가 되어 EKF 필요

G, H => 테일러 급수로 선형화 한것

결국 x (위치 pose)

A, B = 그냥 단위행렬. C = 나랑 landmark 거리

EKF

⇒ 칼만 필터는 로봇의 state 추정에 흔히 사용되는 방법이며, Bayes filter 임.

↳ control input 의한 prediction / 센서 observation 으로의 correction

↳ KF는 linear 안되어서 non linear 한 EKF 가 나왔음

SLAM은 vector로 state, 센서입력, 관찰값 등을 표현해
multi variable의 Gaussian을 많이 사용한다.

Gausian distri ⇒ $p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

multi variable Gau dist ⇒ $p(x) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$