Bayes Filter

* State Estimation

⇒ Estimate the state x of a system given observations z and controls u.

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

{ If x is a fixed thing then $P(x \mid z,u)$ is a number between 0 and 1.
elseif x is a variable then $P(x \mid z,u)$ is probability distribution in x space }

* Recursive Bayes Filter 1

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

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

{ belief that the state is x_t }

∴ { on applying bayes rule }

⇒ Given you know the state of the world, you can forget about what happened in the past.

{ Markov Assumption }

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

{ Applying Law of total Probability }

$$\int_{x_{t-1}} P(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) \, . \, P(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \, dx_{t-1}$$

$$P(x_t \mid x_{t-1}, u_t) \qquad P(x_{t-1} \mid z_{1:t-1}, u_{1:t-1})$$

$$bel(x_{t-1})$$

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

Recursive Equation.

* Prediction and Correction Step

⟹ Bayes filter can be written as a two step Process.

- **Prediction Step**

$$\overline{bel}(x_t) = \int_{x_{t-1}} P(x_t \mid u_t, x_{t-1}) \, bel(x_{t-1}) \, dx_{t-1}$$

- **Correction Step**

$$bel(x_t) = \eta \, P(z_t \mid x_t) \, \overline{bel}(x_t)$$

Observation model

motion model

⟹ The Bayes filter is a framework for recursive state estimation.

↳ There are different realizations.

→ **Kalman filter & friends**
- ↳ Gaussians distribution
- ↳ Linear or Linearized model

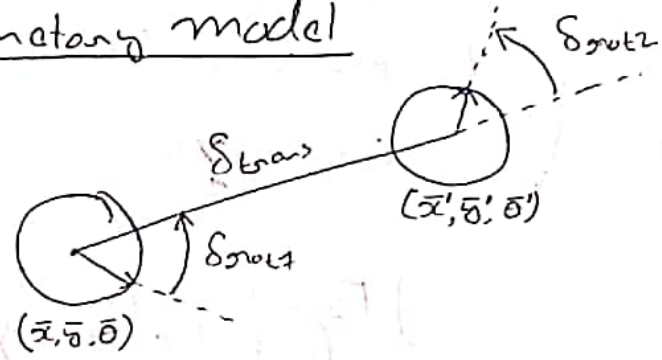
→ **Particle filter**
- ↳ Non-Parametric
- ↳ Arbitrary models (Sampling required)

★ Motion Model

Odometry Model — {best for wheeled robot}

Velocity based model — {flying robot, humanoid robot}

(a) Odometry model



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

$$\delta_{rot1} = atan2(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$$

$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$$

$\Rightarrow$ Odometry information $u = (\delta_{rot1}, \delta_{trans}, \delta_{rot2})$

(b) Velocity-based model

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

$\Rightarrow$ Let initial position of robot be (x, y, θ) and final position be (x', y', θ').

$\Rightarrow$ After executing the velocity command for time Δt.

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

Scanned by CamScanner

* Problem of the Velocity-Based Model

→ Robot moves on a circle

→ The circle constrains the final orientation

→ Ex: Introduce an additional noise term on the final orientation

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

{Term to account for final orientation}

★ Sensor model:

⇒ Strongly depends on sensor we are using.

Bearing Only Sensor ⟹	{Sensor that only provide angle data. eg ⇒ Camera}

* Model for Laser Scanners

⇒ Scan z consists of K measurements.

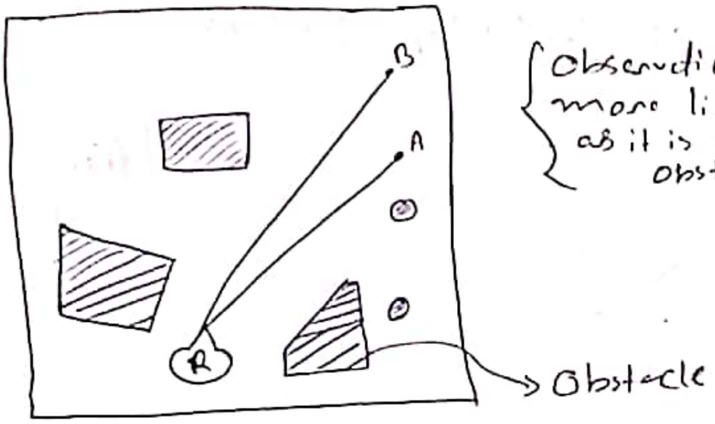
$$z_t = \{z_t^1, z_t^2 \dots z_t^K\}.$$

⇒ Individual measurements are independent given the robot position.

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

Beam-Endpoint Model

⇒ Closer the endpoint of the laser scan to an obstacle, higher chance is there that there was a obstacle.

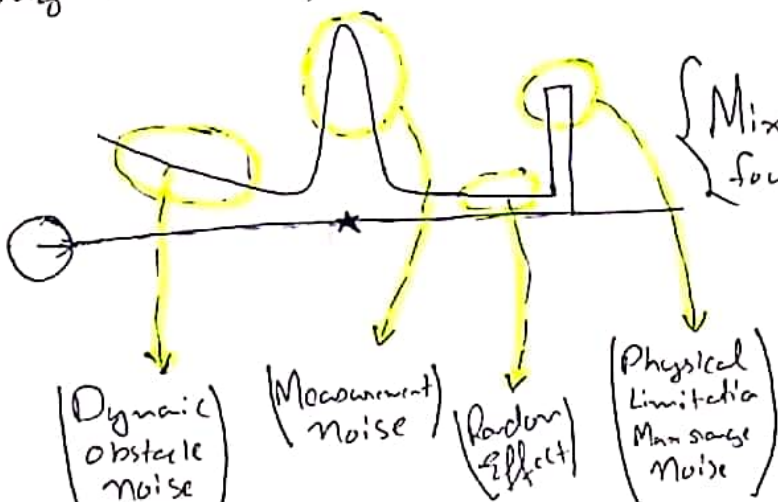


{Observation A is more likely than B as it is closer to an obstacle}

Ray-Cast Model

→ More expensive to compute than Beam-Endpoint Model.

→ Ray-cast model considers the first obstacle along the line of sight.



{Mixture of four model}

★ **Model for perceiving Landmarks with Range-Bearing Sensor**

⇒ Range-bearing $z_t^i = (r_t^i, \varphi_t^i)^T$

r_t^i → distance of landmark

φ_t^i → orientation of landmark wrt heading of robot

→ Robot Pose ⇒ $(x, y, \theta)^T$

⇒ Observation of feature j at location $(m_{j,x}, m_{j,y})^T$

$$\begin{pmatrix} r_t^i \\ \phi_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x}-x)^2 + (m_{j,y}-y)^2} \\ \operatorname{atan2}(m_{j,y}-y,\ m_{j,x}-x) - \theta \end{pmatrix} + \underbrace{Q_t}_{\text{Measurement noise}}$$