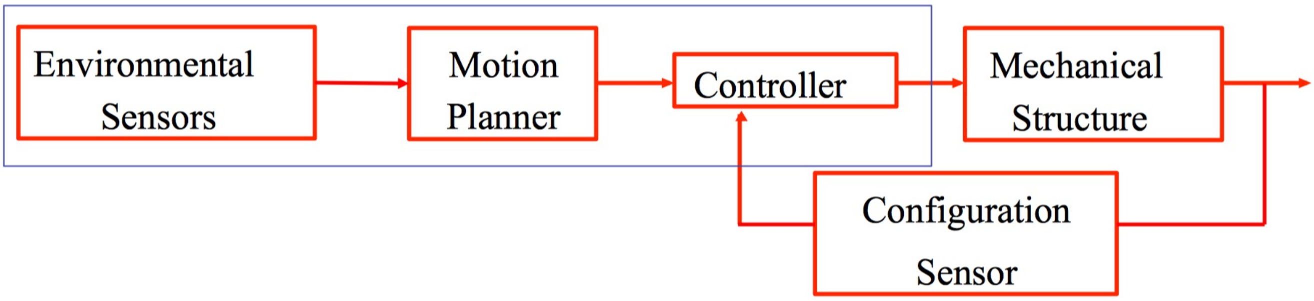
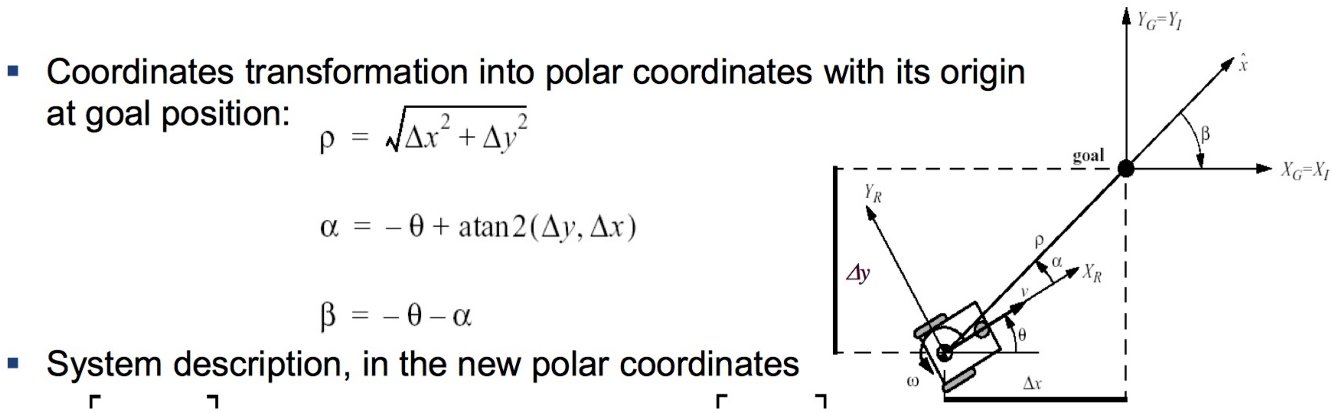
Chap0-3:

- Learning Methods：instruction; imitation; skill transfer; trial-and-error
- Optimization: Perception; Control; Decision; Planning; Scheduling



- Control & Decision Paradigms: Mathematical Model; System; Classical; Reactive; Hybrid; Potential Field Method.
- Behaviors: direct mapping; emergent; modularity.
- Hybrid Deliberative/Reactive Paradigm: plan/sense/act; avoid->wander->Follow Corridor.
- Wheel Types:Fixed; Centered orientable; Off-centered orientable; Swedish
- Drive: Differential; Ackermann; Synchonous; XR4000; Mecanum



- Contact sensors: Bumpers;;Internal sensors:Motor encoder; Accelerometers; Gyroscopes; Compasses; inclinometers;; Proximity sensors: Ultrasonic Range; Laser range-finders; Structured light;Infrared;;Visual sensors: Cameras;;Satellite-based sensors: GPS
- For the motor-shaft with 100 lines, the angle of each line is 360/100 = 3.6. For detection, there are 4 kinds of detection within 2 lines, so the smallest detection angle is 3.6/4 = 0.9

CHAP 4: PROBABILISTIC ROBOTICS

- $$Pr(A \vee B) = PR(A) + Pr(B) - Pr(A \wedge B)$$
- $$P(x) = \sum_y P(x,y) \text{ 和 } p(x) = \int p(x,y)dy$$
- $$P(x,y) = P(x|y)P(y) = P(y|x)P(x)$$

$$P(x|y) = \frac{P(x,y)}{P(y)} = \frac{P(y|x)P(x)}{P(y)} = \frac{likehood \cdot prior}{evidence}$$

- Bayes Rule with Background Knowledge：

$$P(x|y,z) = \frac{P(y|x,z)P(x|z)}{P(y|z)}$$

Conditional Independence:

-
  - $P(x,y|z) = P(x|z)P(y|z)$ is equal to $P(x|z) = P(x|z,y)$ and $P(y|z) = P(y|z,x)$

  but not $P(x,y) = P(x)P(y)$ which is real independence.

  - $P(z) = P(z|open)p(open) + P(z|\urcorner open)p(\urcorner open)$
-

Markov assumption:

z n is independent of z 1 ,…,z n-1 if we know x（set z~ as background knowledge）
|后的多个条件可以被拆开的话，就是相互独立的。

$$P(open \mid z_2, z_1) = \frac{P(z_2 \mid open)\, P(open \mid z_1)}{P(z_2 \mid open)\, P(open \mid z_1) + P(z_2 \mid \neg open)\, P(\neg open \mid z_1)}$$

Action:

Actions are never carried out with absolute certainty. Actions generally increase the uncertainty.

$P(x|u,x')$:executing u changes the state from x′ to x. $P(x|u) = \sum P(x|u,x')P(x')$

eg:$P(closed|u) = P(closed|u,open)P(open) + P(closed|u,closed)P(closed)$

Bayes Filters:

probabilistic tool for estimating the state of dynamic systems.

- Sensor model $P(z|x)$ + Action model $P(x|u,x')$ + Prior probability $P(x)$
- ->$Bel(x_t) = P(x_t|u_1, z_1, \ldots, u_t, z_t)$

$$Bel(x_t) = \eta P(z_t|x_t) \int P(x_t|u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

```
Algorithm Bayes_filter( Bel(x),d ):
η=0
If d is a perceptual data item z then
    For all x do
        Bel'(x) = P(z | x) Bel(x)
        η = η + Bel'(x)
    For all x do
        Bel'(x) = η^{-1} Bel'(x)
Else if d is an action data item u then
    For all x do
        Bel'(x) = ∫ P(x | u, x') Bel(x') dx'
Return Bel'(x)
```

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

Chap5：Probabilistic Motion Models

$p(x|x', u)$ posterior probability： action u carries the robot from x′ to x.

- two types of motion models： Odometry-based/Velocity-based
- Calculate prob:
- normal: $\frac{1}{\sqrt{2\pi b^2}} exp(-\frac{a^2}{2b^2})$ triangular: $max(0, \frac{1}{\sqrt{6}b} - \frac{|a|}{6b^2})$
- Calculating the Posterior:

Algorithm motion_model_odometry(x,x′,u)

$$\delta_{trans} = \sqrt{(\bar{x}'-\bar{x})^2 + (\bar{y}'-\bar{y})^2}$$
$$\delta_{rot1} = \text{atan2}(\bar{y}'-\bar{y}, \bar{x}'-\bar{x}) - \bar{\theta}$$
$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$$

odometry values (u)

$$\hat{\delta}_{trans} = \sqrt{(x'-x)^2 + (y'-y)^2}$$
$$\hat{\delta}_{rot1} = \text{atan2}(y'-y, x'-x) - \bar{\theta}$$
$$\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$$

values of interest (x,x′)

$$p_1 = \text{prob}(\delta_{\text{rot1}} - \hat{\delta}_{\text{rot1}}, \alpha_1 |\hat{\delta}_{\text{rot1}}| + \alpha_2 \hat{\delta}_{\text{trans}})$$
$$p_2 = \text{prob}(\delta_{\text{trans}} - \hat{\delta}_{\text{trans}}, \alpha_3 \hat{\delta}_{\text{trans}} + \alpha_4 (|\hat{\delta}_{\text{rot1}}| + |\hat{\delta}_{\text{rot2}}|))$$
$$p_3 = \text{prob}(\delta_{\text{rot2}} - \hat{\delta}_{\text{rot2}}, \alpha_1 |\hat{\delta}_{\text{rot2}}| + \alpha_2 \hat{\delta}_{\text{trans}})$$

return $p_1 \cdot p_2 \cdot p_3$

- Velocity-Based Model:

Center of circle:

λ=v/w

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\lambda \sin\theta \\ \lambda\cos\theta \end{pmatrix} = \begin{pmatrix} \frac{x+x'}{2} + \mu(y-y') \\ \frac{y+y'}{2} + \mu(x'-x) \end{pmatrix}$$

with

$$\mu = \frac{1}{2}\frac{(x-x')\cos\theta + (y-y')\sin\theta}{(y-y')\cos\theta - (x-x')\sin\theta}$$

1: **Algorithm sample_motion_model_velocity**(u_t, x_{t-1})**:**

2: $\hat{v} = v + \textbf{sample}(\alpha_1|v| + \alpha_2|\omega|)$

3: $\hat{\omega} = \omega + \textbf{sample}(\alpha_3|v| + \alpha_4|\omega|)$

4: $\hat{\gamma} = \textbf{sample}(\alpha_5|v| + \alpha_6|\omega|)$

5: $x' = x - \frac{\hat{v}}{\hat{\omega}}\sin\theta + \frac{\hat{v}}{\hat{\omega}}\, sin(\theta + \hat{\omega}\Delta t)$

6: $y' = y + \frac{\hat{v}}{\hat{\omega}}\cos\theta - \frac{\hat{v}}{\hat{\omega}}\, cos(\theta + \hat{\omega}\Delta t)$

7: $\theta' = \theta + \hat{\omega}\Delta t + \hat{\gamma}\Delta t$

8: *return* $x_t = (x', y', \theta')^T$

- Map-Consistent Motion Model:

Approximation: $p(x \mid u, x', m) = \eta \; p(x \mid m) \; p(x \mid u, x')$

CHAPTER 6: PROBABILISTIC SENSOR MODELS

- Beam-Based Sensors：Individual measurements are independent given the robot position. $P(z|x,m) = \prod_{k=1} P(z_k|x,m)$
- **4 errors**: a known obstacle; cross-talk; unexpected obstacle (people, furniture); missing all obstacles

Measurement noise

$$P_{hit}(z \mid x, m) = \eta \frac{1}{\sqrt{2\pi b}} e^{-\frac{1}{2}\frac{(z-z_{\exp})^2}{b}}$$

Unexpected obstacles

$$P_{unexp}(z \mid x, m) = \begin{cases} \eta \lambda e^{-\lambda z} & z < z_{\exp} \\ 0 & otherwise \end{cases}$$

Random measurement

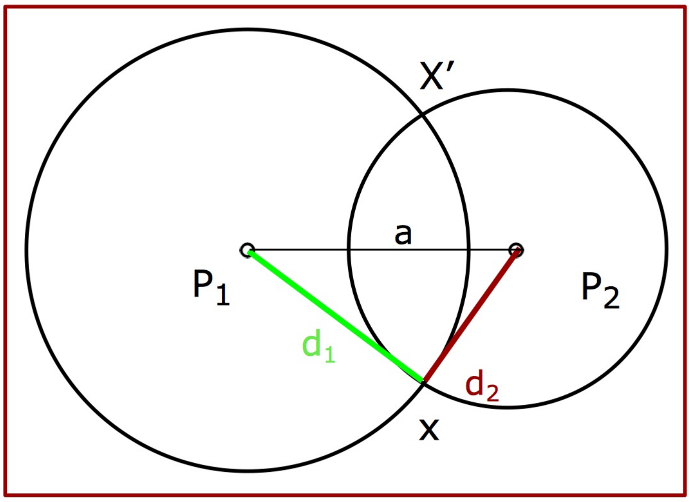
$$P_{rand}(z \mid x, m) = \eta \frac{1}{z_{\max}}$$

Max range

$$P_{\max}(z \mid x, m) = \eta \frac{1}{z_{small}}$$

- Scan-based Sensor Models：Instead of following along the beam, just check the end point.
- Landmark-based Models：

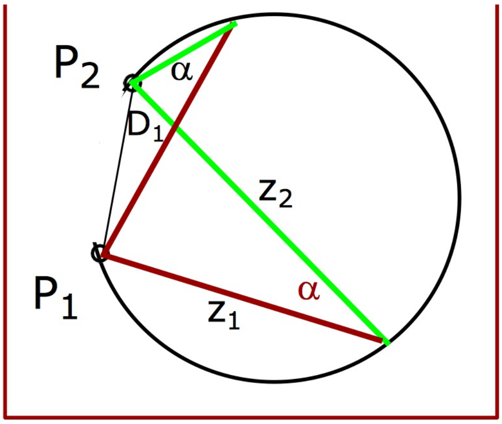
1. Algorithm landmark_detection_model(z,x,m):
   $z = \langle i, d, \alpha \rangle, x = \langle x, y, \theta \rangle$
2. $\hat{d} = \sqrt{(m_x(i) - x)^2 + (m_y(i) - y)^2}$
3. $\hat{a} = \operatorname{atan2}(m_y(i) - y, m_x(i) - x) - \theta$
4. $p_{\det} = \operatorname{prob}(\hat{d} - d, \varepsilon_d) \cdot \operatorname{prob}(\hat{\alpha} - \alpha, \varepsilon_\alpha)$
5. Return $z_{\det} p_{\det} + z_{\mathrm{fp}} P_{\mathrm{uniform}}(z \mid x, m)$



$P_1=(0,0)$

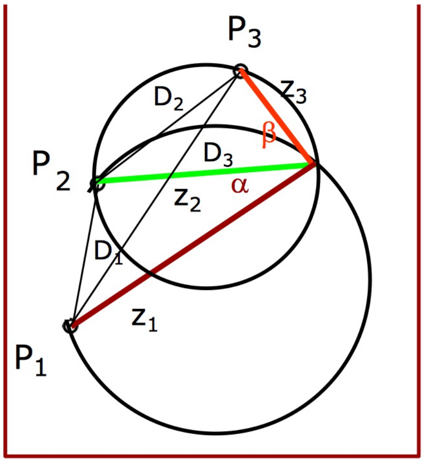
$P_2=(a,0)$

$$x = (a^2 + d_1^2 - d_2^2)/2a$$
$$y = \pm\sqrt{(d_1^2 - x^2)}$$



Law of cosine

$$D_1^2 = z_1^2 + z_2^2 - 2 z_1 z_2 \cos\alpha$$



$$D_1^2 = z_1^2 + z_2^2 - 2 z_1 z_2 \cos(\alpha)$$
$$D_2^2 = z_2^2 + z_3^2 - 2 z_1 z_2 \cos(\beta)$$
$$D_3^2 = z_1^2 + z_3^2 - 2 z_1 z_2 \cos(\alpha + \beta)$$

- Gaussians

$$p(x) \sim N(\mu, \sigma^2):$$

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

$$\left.\begin{array}{l} X \sim N(\mu, \sigma^2) \\ Y = aX + b \end{array}\right\} \Rightarrow \quad Y \sim N(a\mu + b, a^2\sigma^2)$$

$$\left.\begin{array}{l} X_1 \sim N(\mu_1, \sigma_1^{\ 2}) \\ X_2 \sim N(\mu_2, \sigma_2^{\ 2}) \end{array}\right\} \Rightarrow p(X_1) \cdot p(X_2) \sim N\left(\frac{\sigma_2^{\ 2}}{\sigma_1^{\ 2} + \sigma_2^{\ 2}} \mu_1 + \frac{\sigma_1^{\ 2}}{\sigma_1^{\ 2} + \sigma_2^{\ 2}} \mu_2, \quad \frac{1}{\sigma_1^{\ -2} + \sigma_2^{\ -2}} \right)$$

(1) Bayes Rule

$$p(x|z) = \frac{p(z|x)\, p(x)}{p(z)}$$

(2) Bayes Filter

$$p(x_{t+1}|z_{t+1}) = \frac{p(z_{t+1}|x_{t+1}) \int p(x_{t+1}|x_t)\, p(x_t)\, dx_t}{p(z_{t+1})}$$

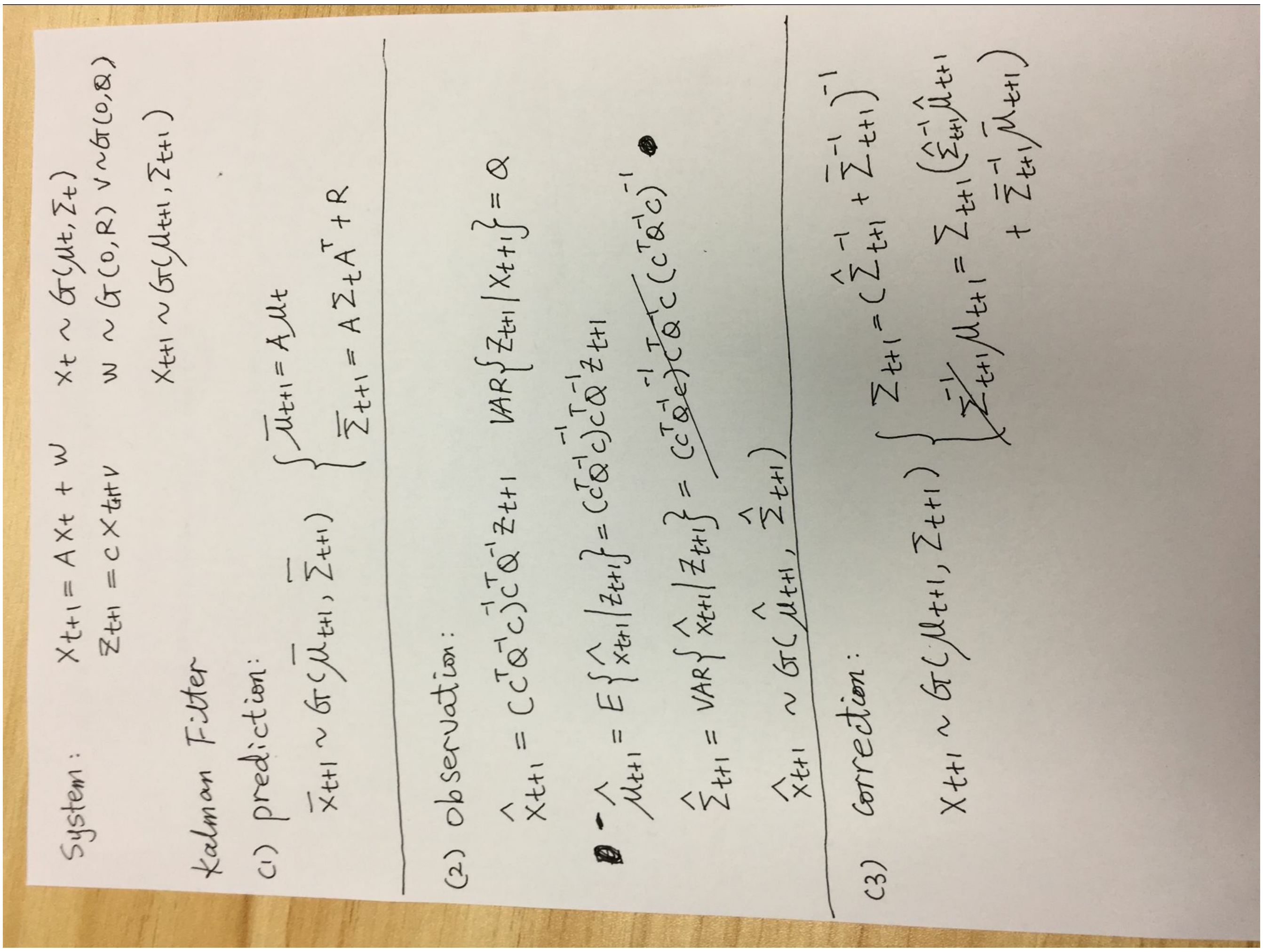
(3) Bayes Filter with Map & Odometry

$$p(x_{t+1}|z_{t+1}, m, u) = \frac{p(z_{t+1}|x_{t+1}, m) \int p(x_{t+1}|x_t, u)\, p(x_t)\, dx_t}{p(z_{t+1})}$$

(4) prediction $p(\bar{x}_{t+1}|u) = \int p(x_{t+1}|x_t, u)\, p(x_t)\, dx_t$

observation $p(\hat{x}_{t+1}|z_{t+1}, m) \longleftarrow p(z_{t+1}|x_{t+1}, m)$

correction $p(x_{t+1}|z_{t+1}, u, m) = p(\bar{x}_{t+1}|u)\, p(\hat{x}_{t+1}|z_{t+1}, m)$


System: $X_{t+1} = AX_t + W$ $\quad X_t \sim G(\mu_t, \Sigma_t)$

$Z_{t+1} = CX_{t+1} + V$ $\quad W \sim G(0, R)$ $\quad V \sim G(0, Q)$

$X_{t+1} \sim G(\mu_{t+1}, \Sigma_{t+1})$

Kalman Filter

(1) prediction:

$$\bar{X}_{t+1} \sim G(\bar{\mu}_{t+1}, \bar{\Sigma}_{t+1}) \quad \begin{cases} \bar{\mu}_{t+1} = A\mu_t \\ \bar{\Sigma}_{t+1} = A\Sigma_t A^T + R \end{cases}$$

(2) observation:

$$\hat{X}_{t+1} = (C^T Q^{-1} C)^{-1} C^T Q^{-1} Z_{t+1} \qquad VAR\{Z_{t+1} | X_{t+1}\} = Q$$

$$\hat{\mu}_{t+1} = E\{\hat{X}_{t+1} | Z_{t+1}\} = (C^T Q^{-1} C)^{-1} C^T Q^{-1} Z_{t+1}$$

$$\hat{\Sigma}_{t+1} = VAR\{\hat{X}_{t+1} | Z_{t+1}\} = \cancel{(C^T Q^{-1} C)^{-1} C^T Q^{-1} C} (C^T Q^{-1} C)^{-1}$$

$$\hat{X}_{t+1} \sim G(\hat{\mu}_{t+1}, \hat{\Sigma}_{t+1})$$

(3) correction:

$$X_{t+1} \sim G(\mu_{t+1}, \Sigma_{t+1}) \quad \begin{cases} \Sigma_{t+1} = (\hat{\Sigma}_{t+1}^{-1} + \bar{\Sigma}_{t+1}^{-1})^{-1} \\ \cancel{\Sigma_{t+1}^{-1}} \mu_{t+1} = \Sigma_{t+1}(\hat{\Sigma}_{t+1}^{-1} \hat{\mu}_{t+1} + \bar{\Sigma}_{t+1}^{-1} \bar{\mu}_{t+1}) \end{cases}$$

CHAPTER 8: PARTICLE FILTER AND MONTE CARLO LOCALIZATION

- Importance Sampling：

  Particle filters are a way to efficiently represent non-Gaussian distribution

  The more particles fall into an interval, the higher the probability of that interval